



Journal of Statistical Software

April 2018, Volume 84, Issue 5.

doi: [10.18637/jss.v084.i05](https://doi.org/10.18637/jss.v084.i05)

evmix: An R package for Extreme Value Mixture Modeling, Threshold Estimation and Boundary Corrected Kernel Density Estimation

Yang Hu

University of Canterbury

Carl Scarrott

University of Canterbury

Abstract

evmix is an R package (R Core Team 2017) with two interlinked toolsets: i) for extreme value modeling and ii) kernel density estimation. A key issue in univariate extreme value modeling is the choice of threshold beyond which the asymptotically motivated extreme value models provide a suitable tail approximation. The package implements almost all existing extreme value mixture models, which permit objective threshold estimation and uncertainty quantification. Some traditional diagnostic plots for threshold choice are provided.

Kernel density estimation with a range of kernels is provided, including cross-validation maximum likelihood inference for the bandwidth. A key contribution over existing kernel smoothing packages in R is that a wide range of boundary corrected kernel density estimators are implemented, which are designed for populations with bounded support. These non-parametric density estimators are also incorporated into the extreme value mixture model framework to describe the density below the threshold.

The quartet of density, distribution, quantile and random number generation functions is provided along with parameter estimation by likelihood inference and standard model fit diagnostics, for both the mixture models and kernel density estimators. The key features of the mixture models and (boundary corrected) kernel density estimators are described and their implementation using the package demonstrated.

Keywords: extreme value mixture model, threshold estimation, boundary corrected kernel density estimation.

1. Introduction

The **evmix** package (Hu and Scarrott 2018) has the most comprehensive suite of extreme value mixture models of any R (R Core Team 2017) package, the current alternatives being **condmixt**

(Carreau 2012) and **CompLognormal** (Nadarajah 2013). Some of these mixture models utilize various kernel density estimators for the non-extreme component of the distribution. The kernel density estimation (KDE) functionality developed for these mixture models has been provided in a standalone form, as these should be of interest to a wide community. In particular, the **evmix** package implements a plethora of boundary corrected kernel density estimation methods, designed to reduce the bias of standard KDE when the range of support is bounded. Applications with bounded support are commonplace, so these boundary correction methods provide beneficial new functionality to R.

A brief introduction to extreme value modeling is provided in Section 1.1. Traditionally, the threshold beyond which the extreme value tail models are applied was pre-chosen using graphical diagnostics or various model fit criteria, giving the so called “fixed threshold approach”. Section 1.2 lists the graphical diagnostics implemented in the **evmix** package. Section 2 describes the recently developed concept of extreme value mixture models, which generally treat the threshold as a parameter to be estimated. A list of the implemented mixture models is provided in Section 2.2. The mixture model fitting functions include options for numerous threshold inference approaches which are detailed in Section 2.3, including the aforementioned fixed threshold approach. The extreme value mixture model functionality is demonstrated on real data in Section 4.

The KDE and boundary corrected KDE approaches implemented in the package are detailed in Section 3. This section is deliberately self-contained, for readers that are only interested in this functionality. In particular, examples of their usage are contained in the Sections 3.2 and 3.4. The usage of KDE within extreme value mixture models is demonstrated with the real data example in Section 4. Finally, Section 5 provides some general advice for users and developers of extreme value mixture models. All the examples use **evmix** version 2.7, which is available from the Comprehensive R Archive Network (CRAN) at <https://CRAN.R-project.org/package=evmix>.

1.1. Extreme value modeling

Extreme value theory is used to derive asymptotically justified models for the tails of distributions, see Coles (2001) for an introduction. A classic asymptotically motivated model for the exceedances of a suitably high threshold u is the generalized Pareto distribution (GPD). Suppose the random variable for an exceedance X of u follows a GPD, parameterized by the scale $\sigma_u > 0$ and shape ξ , with cumulative distribution function (CDF) given by:

$$G(x|u, \sigma_u, \xi) = P(X \leq x | X > u) = \begin{cases} 1 - \left[1 + \xi \left(\frac{x - u}{\sigma_u} \right)_+ \right]^{-1/\xi}, & \xi \neq 0, \\ 1 - \exp \left[- \left(\frac{x - u}{\sigma_u} \right)_+ \right], & \xi = 0, \end{cases} \quad (1)$$

where $x_+ = \max(x, 0)$. For $\xi < 0$ the support is $u < x < u - \sigma_u/\xi$, otherwise it is unbounded from above. The threshold is sometimes described as a location parameter.

In practice, the GPD is applied as an approximation to the upper tail of the population distribution above a sufficiently high threshold. Relatively mild conditions are required for the GPD to be a suitable limiting tail excess model, see Coles (2001) for details.

Implicitly underlying the GPD is a third parameter $\phi_u = P(X > u)$, the “threshold exceedance probability” or “tail fraction”, which is required in calculating quantities like the unconditional

survival probability:

$$P(X > x) = \phi_u [1 - P(X \leq x | X > u)]. \quad (2)$$

Traditionally, the first step in GPD modeling was to choose the threshold u , which is essentially a bias against variance trade-off. A sufficiently high threshold is needed for the asymptotically motivated GPD to give a reliable tail approximation, thus reducing bias, but which also increases the estimation variance due to the reduced sample. By contrast, too low a threshold may mean the GPD is not a good tail approximation, leading to bias, but providing a larger tail sample and so reducing the estimation variance.

Scarrott and MacDonald (2012) provide a reasonably comprehensive review of threshold estimation approaches. Direct threshold inference is not straightforward due to the varying sample size. Traditionally, graphical diagnostics evaluating aspects of the model fit were used to choose a threshold, which was subsequently fixed and so the associated uncertainty was ignored. Some commonly used graphical diagnostics for the so called “fixed threshold approach” are implemented in the package and described in the following Section 1.2.

Over the last decade there has been increasing interest in extreme value mixture models which combine a tail model above the threshold with a suitable model below the threshold. Most of these mixture models treat the threshold as a parameter which can be estimated using standard inference schemes and thus the associated uncertainty on tail inferences can (potentially) be accounted for. The majority of the mixture models in the literature have been implemented in the **evmix** package (Hu and Scarrott 2018). These models are detailed in Section 2.

Particularly flexible extreme value mixture models use non-parametric density estimators below the threshold, following Tancredi, Anderson, and O’Hagan (2006), MacDonald, Scarrott, Lee, Darlow, Reale, and Russell (2011) and MacDonald, Scarrott, and Lee (2013). The latter examples, co-developed by one of the authors of the **evmix** package, use a wide range of KDEs which have also been implemented in the package. These can be used as standalone functionality or as part of an extreme value mixture model. As mentioned in the introduction, the standalone KDE functionality is described in the self-contained Section 3. But an example of implementing the extreme value mixture models combining the KDE with a GPD tail model is provided with the other mixture model examples in Section 4.

1.2. Graphical diagnostics for the “fixed threshold approach”

The GPD exhibits certain threshold stability properties. Suppose the exceedances X above u , i.e., $X|X > u$, follow a $\text{GPD}(\sigma_u, \xi)$ then the exceedances $X|X > v$ of all higher thresholds $v > u$ follow a $\text{GPD}(\sigma_v, \xi)$. The shape parameter is threshold invariant, but the scale parameter is a linear function of the threshold differences $\sigma_v = \sigma_u + \xi(v - u)$. This stability is one of many GPD properties that have traditionally been used to select a suitable threshold in the “fixed threshold approach”, prior to GPD modeling of the threshold exceedances.

For further discussion of such threshold diagnostics see the review paper by Scarrott and MacDonald (2012). The threshold diagnostic plots implemented in the **evmix** package are:

- `mrlplot`: Mean residual life plot (Coles 2001).
- `tcplot`: Threshold stability plots for GPD shape and a modified scale parameter (Coles 2001).

- `hillplot`: Hill plot and some of its many variants; `AltHill`, `SmooHill` and `AltSmooHill` plots (Finkenstädt and Rootzén 2003).
- `pickandsplot`: Pickand’s plot (Finkenstädt and Rootzén 2003).

The latter two plots use the Hill and Pickand’s estimators of the shape parameter, rather than the maximum likelihood estimates. The Hill estimator is suitable for Pareto (or more generally regularly varying) tails, which have a power law form ($\xi > 0$). For brevity, these estimators are not detailed, the interested reader is referred to the help files and references.

The `tcplot`, `hillplot` and `pickandsplot` functions plot the estimated shape parameters against a range of possible thresholds. The guiding principle for threshold choice is to:

Find the lowest value of the threshold such that the parameter estimates stabilize to a constant value which is sustained at all higher thresholds, once the sample uncertainty has been accounted for (typically assessed by pointwise uncertainty intervals).

Hence, for lower thresholds the parameter estimates should be inconsistent with this constant value. The mean residual life (MRL), or equivalently “mean excess”, is given by:

$$E(X - v | X > v) = \frac{\sigma_u + \xi(v - u)}{1 - \xi}, \quad \text{for } \xi < 1 \quad (3)$$

and is infinite for $\xi \geq 1$. The MRL is linear in the excess of the higher threshold $v - u$ above the suitable threshold u with gradient $\xi/(1 - \xi)$ and intercept $\sigma_u/(1 - \xi)$, once the sample uncertainty is accounted for. The sample mean excess is plotted against potential thresholds (including pointwise uncertainty intervals) to give the sample mean residual life plot.

Examples of these diagnostic plots are given in Figures 1 and 2 for the Standard & Poor’s daily log returns, available from the `evir` package (Pfaff and McNeil 2012). The estimates and pointwise 95% confidence intervals are shown by solid and dashed lines respectively. In a more substantive analysis of this dataset we should explore both the upper and lower tails, but for demonstration purposes we will consider only the upper tail.

Three potential thresholds are easily identified from each diagnostic plot, as shown in Table 1. For each threshold choice the GPD is fit using maximum likelihood estimation via the `fgpd` function, e.g., for $u = 0.4$ use `fgpd(spto87, u = 0.4)`. There is clearly large uncertainty in the threshold choice and the resultant shape parameter estimates, which is ignored in the “fixed threshold approach”. Many authors (e.g., Coles 2001) have commented on the subjectivity and substantial experience needed in interpreting these diagnostics. The idea underlying the development of the following extreme value mixture models is to try and provide a more objective threshold estimation approach and to account for the associated uncertainty.

```
R> library("evmix")
R> data("spto87", package = "evir")
R> mrlplot(spto87, try.thresh = c(2.6, 1.5, 0.4))
R> tshapeplot(spto87, try.thresh = c(2.7, 1.8, 0.4), ylim = c(-1, 1),
+   legend = "bottomleft")
R> pickandsplot(spto87, try.thresh = c(1.8, 1.3, 1.05), ylim = c(-1, 1))
```

| Graphical diagnostic | u_1 | $\hat{\xi}_1$ | u_2 | $\hat{\xi}_2$ | \hat{u}_3 | $\hat{\xi}_3$ |
|--------------------------|-------|---------------|-------|---------------|-------------|---------------|
| Mean residual life plot | 0.4 | 0.063 (0.024) | 1.5 | 0.077 (0.070) | 2.6 | -0.27 (0.21) |
| Threshold stability plot | 0.4 | 0.063 (0.024) | 1.8 | 0.098 (0.102) | 2.7 | -0.30 (0.23) |
| Pickand's plot | 1 | 0.041 (0.043) | 1.3 | 0.029 (0.053) | 1.8 | 0.098 (0.102) |

Table 1: Estimated parameters for the GPD model where the threshold is fixed at three potential values determined from the graphical diagnostic plots.

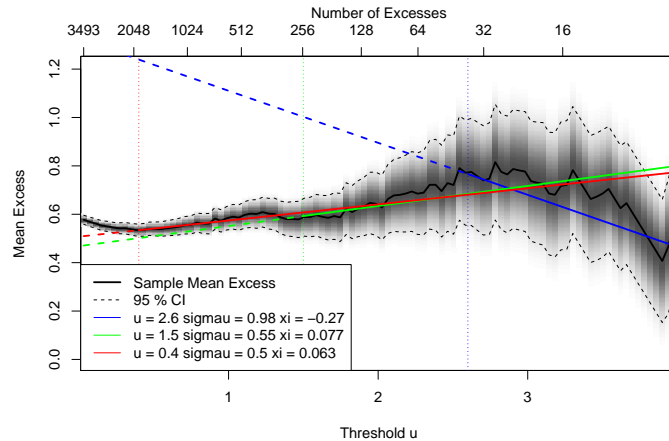
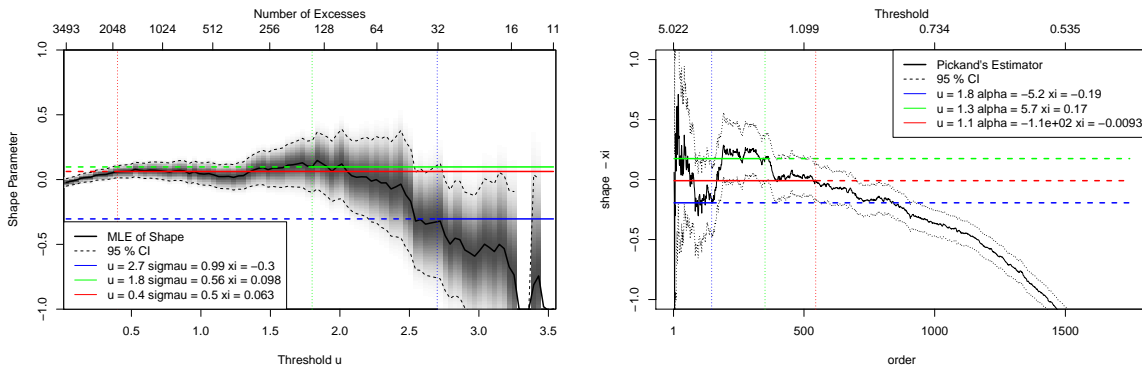


Figure 1: Mean residual life plot for Standard & Poor's returns data.



(a) Shape parameter threshold stability plot.

(b) Pickand's plot.

Figure 2: Shape parameter threshold stability plot and Pickand's plot for Standard & Poor's returns data.

2. Extreme value mixture models

A simple extreme value mixture model form was proposed by Behrens, Lopes, and Gamerman (2004) as depicted in Figure 3, which splices together a parametric model up to the threshold with the usual GPD for the upper tail. We refer to the former as the “bulk model” and the latter as the “tail model”. The CDF takes the form:

$$F(x|\boldsymbol{\theta}, u, \sigma_u, \xi) = \begin{cases} H(x|\boldsymbol{\theta}) & \text{for } x \leq u, \\ H(u|\boldsymbol{\theta}) + [1 - H(u|\boldsymbol{\theta})]G(x|u, \sigma_u, \xi) & \text{for } x > u, \end{cases} \quad (4)$$

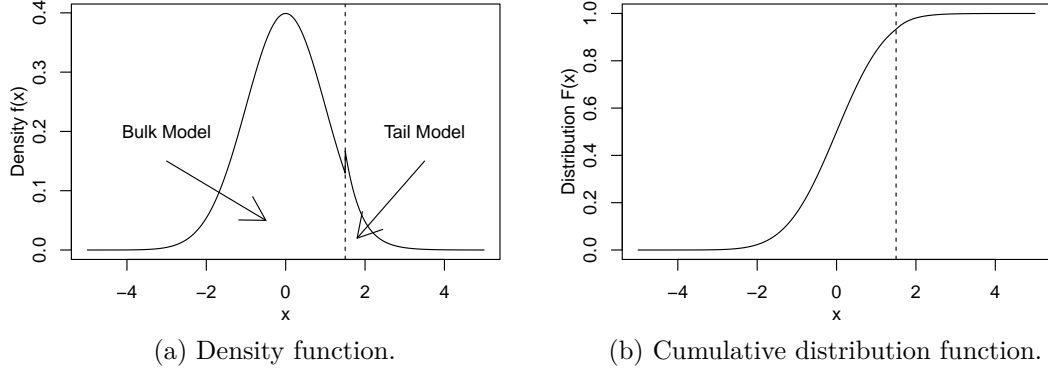


Figure 3: Density function and cumulative distribution function of the parametric extreme value mixture model, with a normal distribution for the bulk model with $\boldsymbol{\theta} = (\mu = 0, \sigma = 1)$, threshold $u = 1.5$ and GPD for the upper tail with $\sigma_u = 0.4$ and $\xi = 0$. The vertical dashed line indicates the threshold.

where $H(\cdot)$ is the bulk model CDF with parameter vector $\boldsymbol{\theta}$.

In this definition, the tail fraction ϕ_u from the unconditional tail model in Equation 2 is replaced by the survival probability of the bulk model assuming it continues above the threshold, i.e., $\phi_u = 1 - H(u|\boldsymbol{\theta})$. Hence, we will refer to it as the “bulk model based tail fraction” approach. It benefits from borrowing information from the usually ample bulk data, to estimate the tail fraction. However, it also exposes tail estimation to misspecification of the bulk model, which is demonstrated in Section 4 and discussed in Section 5.

MacDonald *et al.* (2011) consider a slightly more general form:

$$F(x|\boldsymbol{\theta}, u, \sigma_u, \xi, \phi_u) = \begin{cases} \frac{(1-\phi_u)}{H(u|\boldsymbol{\theta})} H(x|\boldsymbol{\theta}) & \text{for } x \leq u, \\ (1-\phi_u) + \phi_u G(x|u, \sigma_u, \xi) & \text{for } x > u, \end{cases} \quad (5)$$

which is closer to the classical tail modeling approach in Equation 2, which has an explicit extra parameter for the tail fraction ϕ_u . We refer to this as the “parameterized tail fraction” approach. Section 4 shows that this approach can reduce the impact of bulk model misspecification on tail estimates. It should be clear that either specification gives a proper density and that the parameterized tail fraction approach includes the bulk model based tail fraction as a special case, where $\phi_u = 1 - H(u|\boldsymbol{\theta})$.

Figure 3 is obtained using:

```
R> x <- seq(-5, 5, 0.001)
R> y <- dnormgpd(x, u = 1.5, sigmau = 0.4)
R> plot(x, y, xlab = "x", ylab = "Density f(x)", type = "l")
R> abline(v = 1.5, lty = 2)
R> y <- pnormgpd(x, u = 1.5, sigmau = 0.4)
R> plot(x, y, xlab = "x", ylab = "Distribution F(x)", type = "l")
R> abline(v = 1.5, lty = 2)
```

Figure 3 is an example density with a (truncated) normal for the bulk model, using the bulk model based tail fraction as in Equation 4. Notice that the density may be discontinuous at the threshold, but the CDF will be continuous provided the bulk model CDF is continuous.

2.1. Continuity constraint and other extensions

A straightforward adaptation can be made to include a constraint of continuity at the threshold. The density of the GPD at the threshold is simply $g(u|u, \sigma_u, \xi) = 1/\sigma_u$, so in the bulk model based tail fraction approach continuity can be achieved by constraining σ_u :

$$\sigma_u = \frac{1 - H(u|\boldsymbol{\theta})}{h(u|\boldsymbol{\theta})} \quad (6)$$

and for the parameterized tail fraction approach:

$$\sigma_u = \frac{H(u|\boldsymbol{\theta})}{1 - \phi_u} \frac{\phi_u}{h(u|\boldsymbol{\theta})}. \quad (7)$$

All of the extreme value mixture models of the general form presented in Equations 4 and 5 have been implemented in the **evmix** package with two distinct versions: with and without a constraint of continuity in the probability density function (PDF) at the threshold. See Section 2.2 for details.

Further extensions to these extreme value mixture models are possible. For example, continuity constraints up to the second derivative (Carreau and Bengio 2009), a point process representation of the tail model (MacDonald *et al.* 2011), alternative tail models like a Pareto (Solari and Losada 2004) or extended GPD (Papastathopoulos and Tawn 2013). As none of these have been explicitly implemented in the package, they will not be discussed further.

2.2. Implemented mixture models

In the following three subsections the extreme value mixture models of the commonly used form defined by Equations 4 and 5 are listed, the function name conventions are detailed and the mixture models which do not fit this standard framework are also listed.

Standard extreme value mixture models

The bulk models considered in the literature cover the full spectrum of parametric, semi-parametric and non-parametric forms. All the parametric bulk models we are aware of having been used in the literature have been implemented, namely:

- Normal with GPD for upper tail (Cabras and Castellanos 2011), see `help("normgpd")`.
- Gamma with GPD for upper tail (Behrens *et al.* 2004), see `help("gammagpd")`.
- Weibull with GPD for upper tail (Behrens *et al.* 2004), see `help("weibullgpd")`.
- Log-normal with GPD for upper tail (Solari and Losada 2004), see `help("lognormgpd")`.
- Beta with GPD for upper tail (MacDonald 2012), see `help("betagpd")`.
- Normal with GPD for both upper and lower tails (Zhao, Scarrott, Reale, and Oxley 2010; Mendes and Lopes 2004), see `help("gng")`.

The following semi-parametric bulk model has been implemented:

- Finite mixture of gammas with GPD for upper tail (do Nascimento, Gamerman, and Lopes 2012), see `help("mgammagpd")`.

The semi-parametric bulk model using a mixture of exponentials of Lee, Li, and Wong (2012) is not implemented directly, as it is a special case of the mixture of gammas.

The following non-parametric bulk models have been implemented:

- Standard KDE using a constant bandwidth with GPD for upper tail (MacDonald *et al.* 2011), see `help("kdengpd")`.
- Standard KDE using a constant bandwidth with GPD for both upper and lower tails (MacDonald *et al.* 2013), see `help("gkg")`.
- Boundary corrected KDE with GPD for upper tail (MacDonald *et al.* 2013; MacDonald 2012), see `help("bckdengpd")`.

The implemented kernel density estimators are described in the self-contained Section 3, as these are also available in a standalone form outside of the extreme value mixture models.

Naming conventions

A common naming convention is provided for all the distribution functions for the extreme value mixture models, consistent with the prefixes of `d`, `p`, `q` and `r` and inputs in the `base` package (R Core Team 2017); e.g., for the normal with GPD for upper tail.

- `dnormgpd`: Density function.
- `pnormgpd`: Cumulative distribution function.
- `qnormgpd`: Quantile function.
- `rnormgpd`: Random number generation.

These conventions are also extended to the likelihood inference related functions:

- `fnormgpd`: Maximum likelihood estimation of parameters.
- `lnormgpd`: Log-likelihood function.
- `nlnormgpd`: Negative log-likelihood function.
- `proflunormgpd`, `nlnunormgpd`: Profile and negative log-likelihood for given threshold u .

As noted above, all such mixture models are implemented in their simplest form, with no continuity constraints, and with continuity of the PDF at the threshold. The naming convention for the continuity constraint version is to append “con” to the function name. For example:

- `dnormgpd`: Density for normal bulk and GPD tail with no continuity constraint.
- `dnormgpdcon`: With additional constraint of continuity of the PDF at threshold.

Alternative mixture models

The non-standard extreme value mixture models which are not compliant with Equations 4 and 5 that have been implemented in the **evmix** package are:

- Dynamically weighted mixture model (Frigessi, Haug, and Rue 2003), see `help("dwm")`.
- Interval transition mixture model (Holden and Haug 2009), variants listed below.
- Hybrid Pareto (Carreau and Bengio 2009), see `help("hpd")`.

The hybrid Pareto has continuity constraints on the zeroth and first derivative of the PDF at the threshold. A variant of the hybrid Pareto with a single continuity constraint of the PDF at the threshold is provided, see `help("hpdcon")`. Further variants of the hybrid Pareto model are provided in the **condmixt** package (Carreau 2012).

Various bulk models within the interval transition mixture model of Holden and Haug (2009) have been implemented. The model names have been prefixed with `itm`:

- Weibull and GPD for upper tail, see `help("itmweibullgpd")`.
- Normal and GPD for upper tail, see `help("itmnormgpd")`.
- Normal and GPD for both the upper and lower tails, see `help("itmgnng")`.

2.3. Threshold inference within extreme value mixture models

Optimization of the likelihood function for these mixture models is challenging as they often exhibit local modes, which is an inherent model feature as there are often many potential thresholds. Furthermore, once a sufficiently high threshold is reached, the GPD will be an appropriate fit for all higher values (up to sample variation). The extreme value mixture model fitting functions have options for different maximum likelihood approaches for the threshold.

1. `useq = NULL`: Maximum likelihood estimates (MLEs) of all parameters based on the complete likelihood including the threshold (currently the default option, but not the preferred option see Sections 4 and 5 below).
2. `useq` as a scalar and `fixedu = TRUE`: “Fixed threshold approach” with threshold set to `useq`.
3. `useq` as a scalar and `fixedu = FALSE`: Use `useq` as the initial value for the threshold in the optimization of the complete likelihood.
4. `useq` as a vector and `fixedu = TRUE`: Carry out a grid search over thresholds in `useq` to find the value which maximizes the profile likelihood, which is then fixed at that value, and the MLEs of the non-threshold parameters obtained using the conditional likelihood.
5. `useq` as a vector and `fixedu = FALSE`: Carry out a grid search over thresholds in `useq` to find the value, which maximizes the profile likelihood, which is then used as the initial value for the threshold in the maximum likelihood estimation with the complete likelihood.

Further details are given in the help files and in particular within `help("fnormgpd")`. These options provide considerable flexibility for users as to how they wish to infer the threshold.

Traditional graphical diagnostics like those in Section 1.2 can be used to a-priori fix the threshold using option (2), as in the classical “fixed threshold approach”. The review paper by [Scarrott and MacDonald \(2012\)](#) and also one of the reviewers highlight numerous alternative threshold estimators/algorithms, e.g., the robust estimator of [Dupuis \(2000\)](#), bootstrap methods ([Danielsson, De Haan, Peng, and De Vries 2001](#)), exponential regression models ([Beirlant, Dierckx, Guillou, and Staarica 2002](#)) or sequential methods ([Drees and Kaufmann 1998](#)).

However, one of the key motivations for developing extreme value mixtures is to provide a more objective approach to threshold estimation (and uncertainty quantification) compared to the traditional graphical diagnostics. Approaches 1, 3, 4 and 5 provide estimation options for the threshold, but only options 1, 3 and 5 currently provide uncertainty estimates for the threshold. Provision of uncertainty estimates in all cases is a topic of current research. The reliability of the observed information matrix based uncertainty estimates for the threshold has seemingly not yet been explored in the literature for these models.

The default option (1) obtains the MLE of the complete parameter vector using the "BFGS" method within the `optim` function from the `stats` package ([R Core Team 2017](#)). As mentioned above, direct optimization of the likelihood is challenging due to the commonplace multimodality of the likelihood. This implies that the optimization is very sensitive to the initial value for the threshold. The recommended alternative is to apply a grid search over a set of potential thresholds to find the value that maximizes the profile likelihood (thus avoiding the problem of the optimization routine getting stuck in a local mode) which should then be fixed, as in option (4) above.

The grid search is less than ideal, as it adds to the computation time as will be seen in Section 4. More sophisticated likelihood inference approaches (and optimization schemes for them) are a topic of ongoing research for extreme value mixture models. The underlying likelihood functions are documented and visible to users who wish to employ their own likelihood inference/optimization schemes.

3. Kernel density estimation

Numerous of the extreme value mixture models listed in Section 2.2 use a non-parametric KDE for the non-extreme component of the population distribution. Existing functions in R packages, e.g., `stats`, `KernSmooth` ([Wand 2015](#)) and `ks` ([Duong 2015](#)), do not lend themselves to this purpose. Furthermore, currently there is limited functionality for boundary corrected KDE in R, typically the `logspline` ([Kooperberg 2013](#)) or the log-transformation approach in the `ks` package are used. As the KDE functionality is of interest to a wide group of users, these functions have been made available separately. The standalone functionality is described in this section and their utilization within an extreme value mixture model is demonstrated in Section 4.

Let x_1, x_2, \dots, x_n be a sample of a random variable X from a population density $f(x)$. [Wand and Jones \(1995\)](#) show that the density of X can be estimated by a KDE of the form:

$$\hat{f}(x) = \frac{1}{n\lambda} \sum_{i=1}^n K\left(\frac{x - x_i}{\lambda}\right), \quad (8)$$

where the kernel function $K(\cdot)$ has the following properties:

$$(a) \int K(z) dz = 1, \quad (b) \int z K(z) dz = 0 \quad \text{and} \quad (c) \int z^2 K(z) dz < \infty.$$

These imply that the kernel function integrates to one, is symmetric (mean zero) and has finite variance. Although not required, the kernel function is typically non-negative $K(z) \geq 0$ so that it could be a proper density function. The bandwidth $\lambda > 0$ is key in controlling the level of smoothing, which is a trade-off between the bias and variance. Too small a bandwidth leads to under-smoothing and an increase in the variance, whereas too large a bandwidth will reduce the variance but increase the bias. Most kernel functions bounded support on $|z| \leq 1$, making the bandwidth λ the half-width of the kernel function. All the commonly used kernel functions are implemented in the `evmix` package, see `help("kernels")` for details.

3.1. Bandwidth estimation

To complement the inbuilt bandwidth estimation procedures we have implemented cross-validation likelihood inference, treating it as a parameter in a mixture model which was first proposed by [Habbema, Hermans, and Van den Broek \(1974\)](#) and [Duin \(1976\)](#). To avoid the degeneracy in direct maximum likelihood estimation they suggest using leave-one-out cross-validation likelihood:

$$L(\lambda|\mathbf{X}) = \prod_{i=1}^n \frac{1}{(n-1)} \sum_{\substack{j=1 \\ j \neq i}}^n K\left(\frac{x_i - x_j}{\lambda}\right).$$

This cross-validation likelihood is included within the likelihood for the extreme value mixture models, for the bulk observations when a KDE is used for the bulk model.

3.2. Example of kernel density estimation

The following code demonstrates using maximum cross-validation likelihood estimation for the bandwidth with the default Gaussian kernel for a simulated standard normal dataset.

```
R> set.seed(0)
R> xdata <- rnorm(50)
R> fit <- fkden(xdata)
R> xp <- seq(-4, 4, 0.01)
R> hist(xdata, 10, freq = FALSE, xlim = c(-4, 4), ylim = c(0, 0.7))
R> rug(xdata)
R> lines(xp, dnorm(xp), col = "black")
R> with(fit, lines(xp, dkden(xp, x, lambda), col = "red"))
```

which can be compared to the KDE obtained using the inbuilt `density` function:

```
R> lines(density(xdata), lty = 2, col = "green")
R> legend("topright", c("True Density", "KDE fitted by MLE in evmix",
+ "KDE using density, default bw"), lty = c(1, 1, 2),
+ col = c("black", "red", "green"))
```

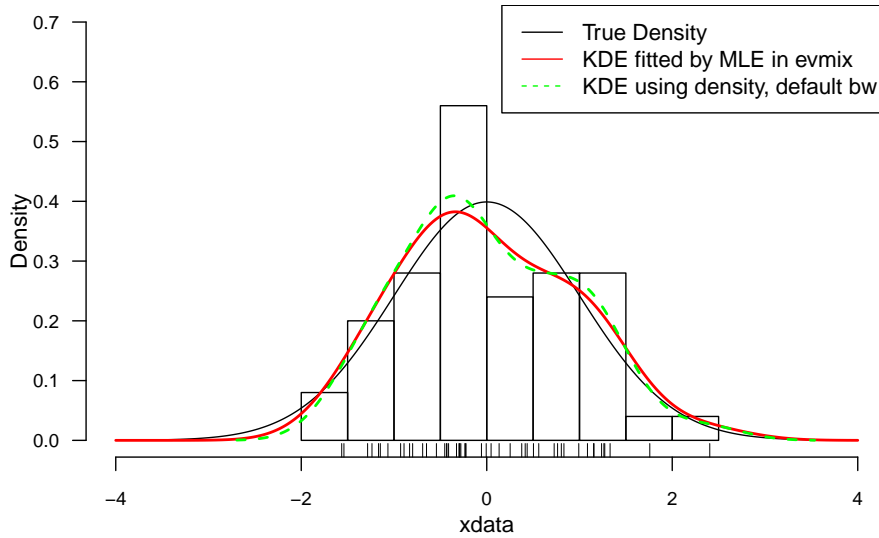


Figure 4: Example of the fitted KDE for 50 simulated data points from a standard normal using the **evmix** package in red. Comparison to the **density** function using the default bandwidth estimator in green.

Notice in the resulting Figure 4 that the bandwidth estimated using the cross-validation likelihood is rather similar to that of Silverman’s rule of thumb used by default in the **density** function.

The principle drawback of maximum cross-validation likelihood for the bandwidth is that it is upwardly biased (leading to over-smoothing) for heavy tailed distributions, as the distance between the upper and lower order statistics does not decay to zero as $n \rightarrow \infty$, see [MacDonald *et al.* \(2011\)](#) for details. An example of over-smoothing is given in the following code and Figure 5, where the blue line is clearly drastically over-smoothed. A robust estimator of bandwidth can be achieved by ignoring the heavy tails in the cross-validation likelihood, but note that the observations in the tails are still required in the KDE otherwise a boundary bias would occur. Denote the observations not in the tails by $A = \{i : -3 \leq x_i \leq 3\}$ then the robust cross-validation likelihood is:

$$L_A(\lambda|\mathbf{X}) = \prod_A \frac{1}{(n-1)} \sum_{\substack{j=1 \\ j \neq i}}^n K\left(\frac{x_i - x_j}{\lambda}\right), \quad (9)$$

where the product includes only observations in A , but the KDE sum includes the extra tail observations in $A^c = \{i : (x_i < -3) \cap (x_i > 3)\}$. The reduced set of observations A which contributes to the cross-validation likelihood are specified by the usual **kerncentres** input and the extra tail observations which contribute to the KDE sum are given by the **extracentres** input. Notice that the bandwidth estimate obtained by ignoring the tails in the maximum likelihood estimation is robust to the heavy tails (red line) in Figure 5. An alternative approach to overcoming the bandwidth bias for heavy tails, is to use an extreme value mixture model with the KDE for the bulk distribution and GPD for the tails, as shown in Section 4.

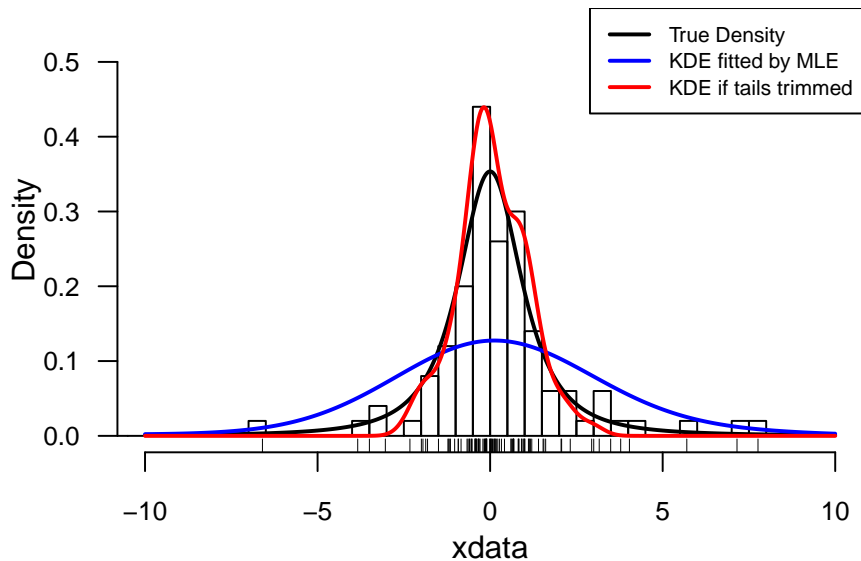


Figure 5: Example of fitted KDE for simulation from a Student- $t(\nu = 2)$ distribution using **evmix** in blue. Robust bandwidth estimator with heavy tails ignored in the cross-validation likelihood in red.

The following code simulates 100 variates from a heavy tailed Student- $t(\nu = 2)$ distribution and uses the entire sample in the cross-validation likelihood estimation:

```
R> set.seed(0)
R> xdata <- rt(100, df = 2)
R> fit <- fkden(xdata)
```

and re-fits using a trimmed estimate where data larger in magnitude than 3 are ignored:

```
R> intails <- ((xdata <= -3) | (xdata >= 3))
R> fit.notails <- fkden(xdata[!intails], extracentres = xdata[intails])
```

The trimmed bandwidth estimate is then applied across the entire sample to get the KDE:

```
R> xp <- seq(-10, 10, 0.01)
R> hist(xdata, seq(floor(min(xdata)), ceiling(max(xdata)), 0.5),
+     freq = FALSE, xlim = c(-10, 10))
R> rug(xdata)
R> lines(xp, dt(xp, df = 2), col = "black")
R> with(fit, lines(xp, dkden(xp, x, lambda), col = "blue"))
R> with(fit.notails, lines(xp, dkden(xp, x, lambda), col = "red"))
R> legend("topright", c("True Density", "KDE fitted by MLE",
+ "KDE if tails trimmed"), col = c("black", "blue", "red"))
```

The resulting plot is given in Figure 5.

3.3. Boundary corrected kernel density estimators

Many population distributions have bounded or semi-bounded support, for example the exponential distribution. In such situations it is clearly desirable that $\hat{f}(x)$ based on KDE has the same support. Direct application of the standard KDE as in Equation 8 will not necessarily satisfy this constraint, potentially leading to a negative bias near the boundary as probability mass is allowed to leak beyond the range of support. A wide range of boundary corrected kernel density estimators (BCKDEs) have been proposed in the literature, to overcome this bias and reduce the variance of near-boundary estimates. The approaches implemented in the **evmix** package are listed below. Seemingly, only the log-transformation and truncation methods below are currently available in R packages, so these functions extend available functionality in R.

Each BCKDE is listed by their name in the `bcmethod` input, see `help("bckden")` for details. For brevity, the formal definitions are not all given but the key references are provided. For notational simplicity, the kernels are assumed to have bounded support $|z| < 1$. The following definitions assume a semi-bounded support on $x \in [0, \infty)$. The data can be negated for an upper bound and/or shifted to achieve an alternative bound.

- `bcmethod = "simple"` (default) applies the simple boundary correction method in Equation 3.4 of Jones (1993) and is equivalent to kernel weighted local linear fitting near the boundary. These boundary corrected kernels can be negative, so a negative density estimate can result. The non-negative correction method of Jones and Foster (1996) has been implemented and is applied by default (`nn = "jf96"`). Further, the density estimate may not be proper and so re-normalization to unity is applied by default (`proper = TRUE`).
- `bcmethod = "cutnorm"` applies the cut and normalization method of Gasser and Müller (1979). Kernels $K(\cdot)$ which are centered at a data point x_i sufficiently close to the boundary to cause leakage $x_i < \lambda$ are simply truncated and re-normalized to unity:

$$K_{\text{cut}}\left(\frac{x - x_i}{\lambda}\right) = \frac{1}{a_0(x_i/\lambda)} K\left(\frac{x - x_i}{\lambda}\right) I(x \geq 0). \quad (10)$$

The re-normalization constant $a_0(p) = \int_{-1}^p K(v) dv$ is the area under the kernel above the boundary. The kernels centered away from the boundary $x_i > \lambda$ are unaffected.

- `bcmethod = "renorm"` applies the first order bias correction of Diggle (1985), where the kernel density estimate $\hat{f}(x)$ at $x = p\lambda$ is adjusted near the boundary where $p < 1$:

$$\tilde{f}(x) = \frac{1}{a_0(p)} \hat{f}(x). \quad (11)$$

As with the "simple" method the resultant density may need re-normalization to unity. Further from the boundary $p \geq 1$ the adjustment $a_0(p) = 1$ has no effect.

- `bcmethod = "reflect"` applies the reflection method of Schuster (1985) which is equivalent to the dataset being supplemented by the same dataset negated. This method implicitly assumes $f'(0) = 0$, which can cause artifacts at the boundary.

- `bcmethod = "logtrans"` applies KDE on the log-scale and then back-transforms (with explicit normalization) following [Marron and Ruppert \(1994\)](#), which is the approach implemented in the `ks` package ([Duong 2015](#)). As the KDE is applied on the log scale, the effective bandwidth on the original scale is non-constant.

The above methods can use any of the kernels used for standard KDE, see `help("kernels")`. The following boundary correction methods do not use kernels in their usual sense, but are commonly used and have similar features. The first two of these pseudo-kernel density estimators allow for an upper and lower boundary on the support, which has been implemented using a lower boundary of zero and upper boundary specified by the `xmax` input.

- `bcmethod = "beta1"` and `"beta2"` implement the beta and modified beta pseudo-kernels which [Chen \(1999\)](#) denotes by $\hat{f}_1(x)$ and $\hat{f}_2(x)$ respectively on page 133 of their paper. Re-normalization is required to obtain a proper density.
- `bcmethod = "copula"` uses the bivariate normal copula based kernels of [Jones and Henderson \(2007\)](#). The bandwidth is defined as $\lambda = 1 - \rho^2$ where ρ is the usual correlation coefficients and so is limited to $(0, 1)$.
- `bcmethod = "gamma1"` and `"gamma2"` use the gamma and modified gamma kernels of [Chen \(2000\)](#) specified by their $\hat{f}_1(x)$ on page 472 and $\hat{f}_2(x)$ on page 473 respectively. Re-normalization is required to obtain a proper density.

Maximum cross-validation likelihood estimation for the bandwidth is implemented for all these BCKDE methods. The only difference between this estimation method for the BCKDE, versus the KDE above, is that in the latter an initial value for the bandwidth is provided (using Silverman's rule of thumb in the `bw.nrd0` function) but no default is provided for BCKDE. Not all the implemented BCKDE approaches have a "rule of thumb" type bandwidth estimator, so the user must provide an initial guess. If the initial guess is inappropriate then the `fbckden` maximum likelihood estimation fitting function will automatically try values up to 32 times bigger (or smaller) than that provided, which in most cases will permit the optimization routine to get past the initial step.

3.4. Example of boundary corrected kernel density estimation

The following code and result in [Figure 6](#) give an example of estimating the bandwidth using maximum cross-validation likelihood estimation implementing the reflection method. The reflection and cut & normalization approaches are the most computationally efficient of the implemented boundary correction methods, as re-normalization and non-negativity corrections are not required. The data are simulated from a $\text{gamma}(\alpha = 1, \beta = 2)$ distribution, which is chosen to show the challenging situation of a pole at the boundary.

```
R> set.seed(0)
R> xdata <- rgamma(500, shape = 1, scale = 2)
R> fit <- fbckden(xdata, linit = 1, bcmethod = "reflect")
```

In general, we find that bandwidth estimates from the computationally efficient reflection or cut & normalization methods provide reliable estimates when supplemented with the simple or re-normalization boundary correction KDE. This sub-optimal approach avoids the computational burden associated with the re-normalization (and/or non-negative correction).

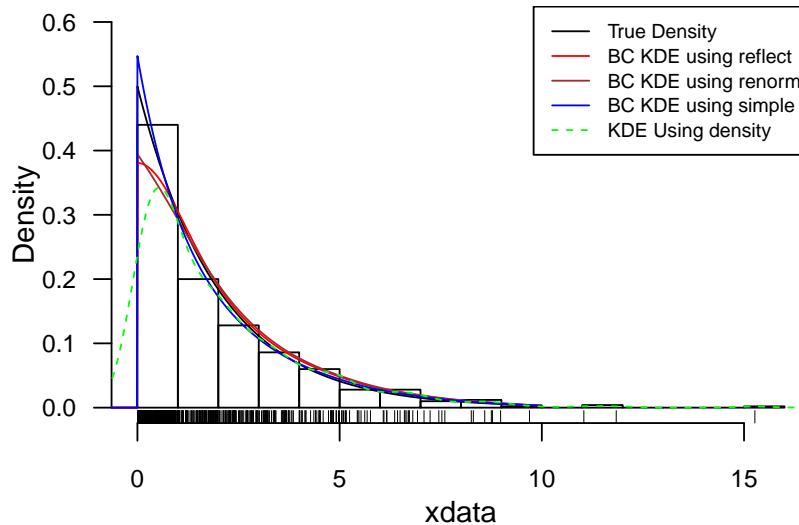


Figure 6: Example of fitted KDE for simulated data from $\text{gamma}(1, 2)$ using the reflection method in red. The simple and re-normalization boundary correction methods are shown in blue and brown, with the bandwidth from the reflection method. Comparison to the poor performance of the standard KDE with the `density` function using the default bandwidth estimator in green.

```
R> xp <- seq(-1, 10, 0.01)
R> hist(xdata, 20, freq = FALSE, ylim = c(0, 0.6))
R> rug(xdata)
R> lines(xp, dgamma(xp, shape = 1, scale = 2), col = "black")
R> with(fit, lines(xp, dbckden(xp, xdata, lambda, bcmethod = bcmethod),
+   col = "red"))
R> with(fit, lines(xp, dbckden(xp, xdata, lambda, bcmethod = "renorm"),
+   col = "brown"))
R> with(fit, lines(xp, dbckden(xp, xdata, lambda, bcmethod = "simple"),
+   col = "blue"))
R> lines(density(xdata), lty = 2, col = "green")
R> legend("topright", c("True Density", "BC KDE using reflect",
+   "BC KDE using renorm", "BC KDE using simple", "KDE Using density"),
+   lty = c(rep(1, 4), 2), col = c("black", "red", "brown", "blue", "green"))
```

The standard kernel density estimator shown in green is severely biased near the boundary. The reflection approach has much better performance, but still exhibits some bias. The simple boundary correction method performs much better, even when using the bandwidth estimate obtained using the reflection method.

4. Extreme value mixture modeling example

The Standard & Poor's daily log returns `spto87` from the `evir` package is used here to demonstrate the package `evmix` and the features of extreme value mixture models. The time series is

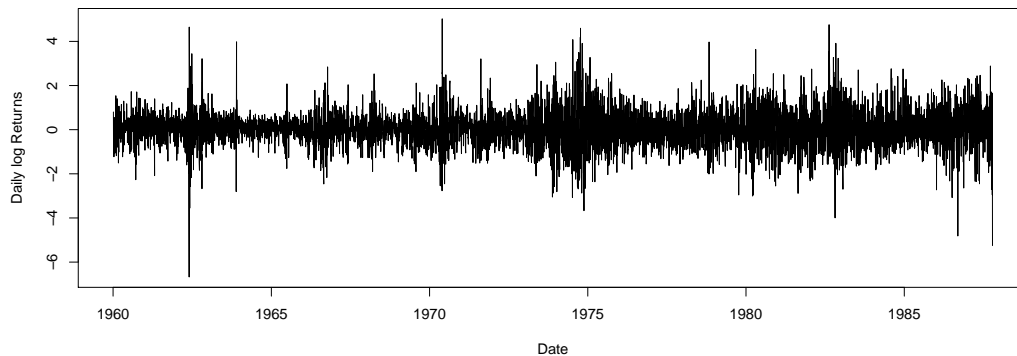


Figure 7: Time series plot of the Standard & Poor's daily log returns dataset `spto87` from the `evir` package.

shown in Figure 7, which exhibits the stylized facts expected of financial returns, in particular the volatility clustering, but these complexities are ignored for brevity.

```
R> data("spto87", package = "evir")
R> plot(attr(spto87, "times"), spto87, type = "l", xlab = "Date",
+       ylab = "Daily log Returns")
```

The following code compares the fitted density of a normal distribution and an extreme value mixture model with a normal for the bulk and GPD for the tail (denoted as normal+GPD). Variants of the normal+GPD with both specifications of the tail fraction have been fitted. The usual unbiased estimates of the normal parameters are `mean(spto87) = 0.025` and `sd(spto87) = 0.805`. The poor fit of the normal model is clear as the red line is so far from the sample histogram density estimator in Figure 8. The poor fit is due to both empirical tails being heavier than the asymptotically exponential form of the normal distribution.

```
R> fit.bulk <- fnormgpd(spto87, useq = seq(0.2, 2, 0.1), fixedu = TRUE)
R> fit.para <- fnormgpd(spto87, useq = seq(0.2, 2, 0.1), fixedu = TRUE,
+   phiu = FALSE)
R> hist(spto87, freq = FALSE, breaks = 50, xlab = "Daily log Returns",
+   xlim = c(-4, 4), ylim = c(0, 0.8))
R> xp <- seq(-7, 7, 0.01)
R> lines(xp, dnorm(xp, mean(spto87), sd(spto87)), col = "red")
R> fx.bulk <- with(fit.bulk, dnormgpd(xp, nmean, nsd, u, sigmau, xi))
R> lines(xp, fx.bulk, col = "blue", lwd = 2)
R> abline(v = fit.bulk$u, col = "blue", lwd = 2)
R> fx.para <- with(fit.para, dnormgpd(xp, nmean, nsd, u, sigmau, xi, phiu))
R> lines(xp, fx.para, col = "green", lty = 2, lwd = 2)
R> abline(v = fit.para$u, col = "green", lty = 2, lwd = 2)
```

The poor performance of the normal model persists when used within the normal+GPD mixture. Notice that the bulk model based (blue solid line) and parameterized tail fraction (green dashed line) provide very different density fits, particularly below the threshold. The bulk model based tail fraction does not modify the density below the threshold in Equation 4,

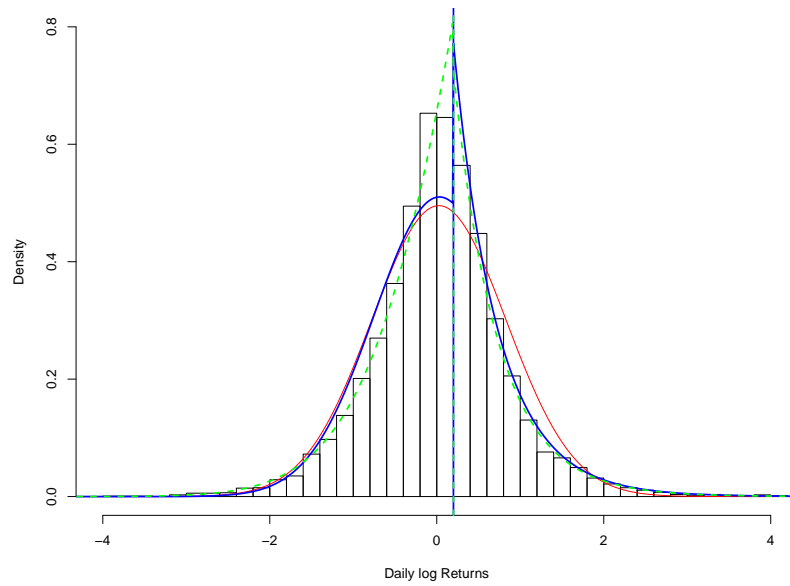


Figure 8: Density histogram of the Standard & Poor's daily log returns dataset `spt087` from the `evmix` package. A fitted normal distribution is shown by the red solid line. The fitted density function of the normal+GPD with bulk model based (blue solid line) and parameterized tail fraction (green dashed line).

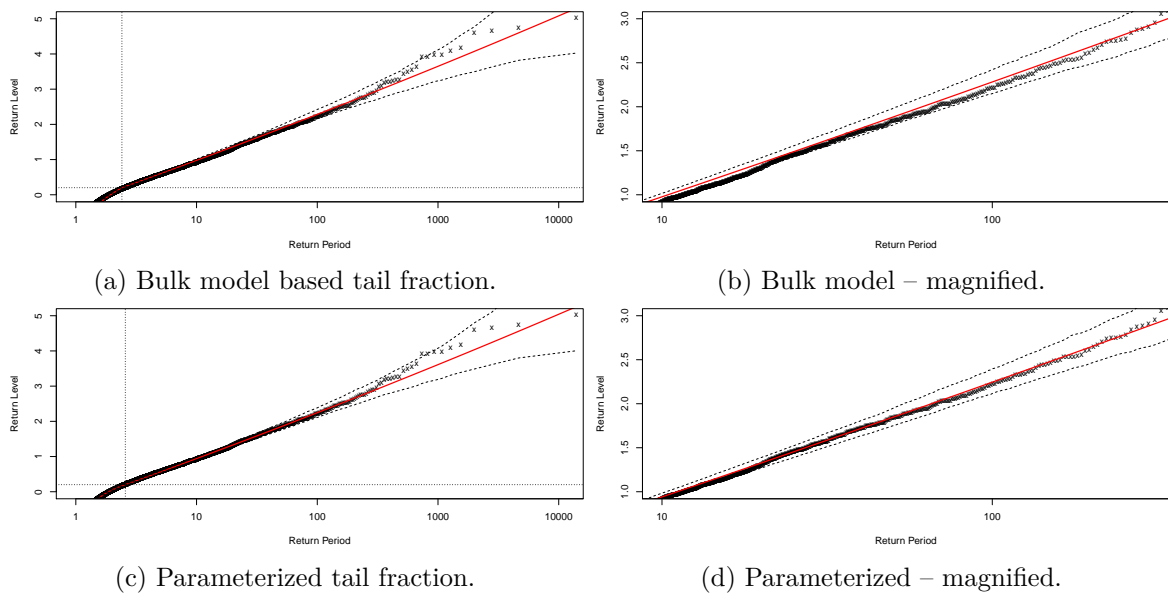


Figure 9: Return level plots for normal+GPD model fits with bulk model based tail fraction (upper plots) and parameterized tail fraction (lower plots).

so below the threshold the density is very similar to that of the standalone normal (blue line close to red). The estimated normal bulk parameters are $\hat{\mu} = 0.032$ (0.011) and $\hat{\sigma} = 0.782$ (0.009) which are similar to the unbiased normal parameter estimates given above.

The parameterized tail fraction approach provides an extra degree of freedom used to re-scale

both the tail and bulk components in Equation 5. The extra parameter is justified by the deviance drop in Table 2. The tail fit has improved, as the tail fraction is estimated from the sample fraction of exceedances so is not biased by the poor bulk model which undermines the bulk model based tail fraction estimate. The re-scaling of the bulk density allows the mixture to provide a much better fit below the estimated threshold.

The following code produces return level plots for evaluation of the tail fit.

```
R> rlplot(fit.bulk)
R> rlplot(fit.para)
```

Both the bulk model based and parameterized tail fraction approaches produce similar fits for the highest return periods above around 1,000 in Figure 9. However, nearer the threshold (horizontal line) the fit is better for the parameterized tail fraction approach, shown by the distance between the data points and red line in the magnified graphs on the right of Figure 9.

```
R> rlplot(fit.bulk, rplim = c(10, 300), rllim = c(1, 3))
R> rlplot(fit.para, rplim = c(10, 300), rllim = c(1, 3))
```

The poor bulk model not only affects the tail fraction, but also the threshold which is biased downwards. It is lower than indicated by the traditional graphical diagnostics in Table 1 from Section 1.2. The estimated thresholds are the lowest values considered in the profile likelihood grid search. The low thresholds allow the GPD to overcome as much of the poor tail fit as it is capable. Such a result is a common indicator of a poor choice of (bulk) model. The default optimization of the complete likelihood (Option 1 in Section 2.3) can get stuck in local modes, which are often close to the initial values for the parameters (in particular the threshold). This problem is demonstrated by the following code. The default initial value for the threshold is the 90% sample quantile estimated using the default settings of the `quantile` function, which is 0.915 for this dataset. The estimated threshold is $\hat{u} = 0.896$, close to the initial value. The current advice is to use the grid search of the profile likelihood over a range of sensible user chosen thresholds (Option 4 in Section 2.3), followed by fixing the threshold. Unfortunately, this optimization problem has not been well explored in the literature and it is a non-trivial exercise to make this option the default for all the mixture models in the package, so this is an area of ongoing research and package development.

```
R> fit.default <- fnormgpd(spto87)
R> c(fit.default$u, quantile(spto87, 0.9))
```

```
          90%
0.8963452 0.9147473
```

A superior fit is obtained by splicing a GPD for both tails, to capture the empirical heavy tail. The following code fits the GPD+normal+GPD mixture model and creates the plot in Figure 10. Overall, the density is a better fit. The GPD+normal+GPD model with bulk model based tail fraction has 3 extra parameters (lower tail threshold and GPD parameters) compared to the normal+GPD. The deviance increase of $2 * (\text{fit.bulk}\$nllh - \text{fit.gng}\$nllh) = 355.97$ provides validation as it is well above the $\chi_3^2(0.95) = 7.815$ critical

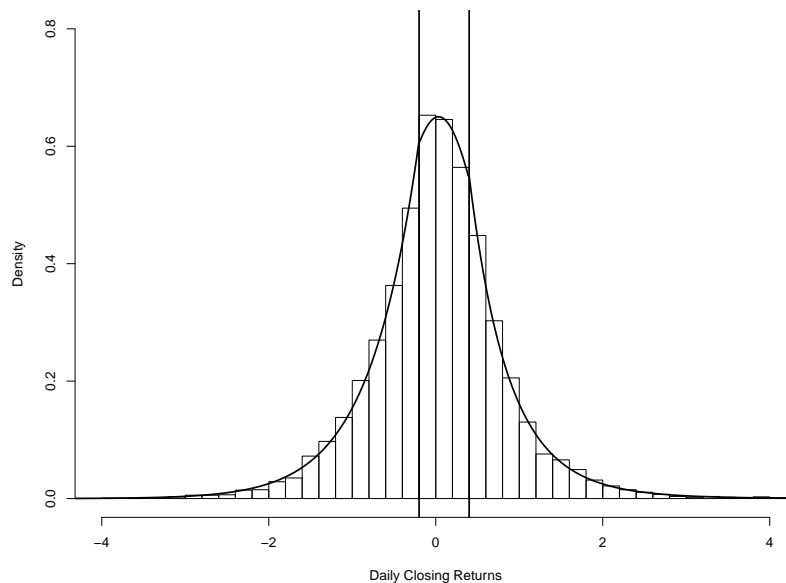


Figure 10: Density histogram of the Standard & Poor's daily log returns dataset `spto87` from the `evir` package. The fitted density function of the GPD+normal+GPD with bulk model based tail fraction.

value. The quartet of model fit diagnostics in Figure 11 confirms that the model performs well for upper tail estimation.

The estimated upper tail threshold for the GPD+normal+GPD is consistent with the lowest values selected from the traditional mean residual life plot and threshold stability plots in Table 1 in Section 1.2. But it is much smaller than all those suggested by the Pickand's plot.

```
R> fit.gng <- fgng(spto87, ulseq = seq(-2, -0.2, 0.1),
+   urseq = seq(0.2, 2, 0.1), fixedu = TRUE)
R> hist(spto87, freq = FALSE, breaks = 50, xlab = "Daily Closing Returns",
+   xlim = c(-4, 4), ylim = c(0, 0.8))
R> fx <- with(fit.gng, dgng(xp, nmean, nsd, ul, sigmaul, xil, phiul = TRUE,
+   ur, sigmaur, xir, phiur = TRUE))
R> lines(xp, fx)
R> abline(v = c(fit.gng$ul, fit.gng$ur))
```

Table 2 gives the computing time for fitting all the above models on a quad core Intel-i5 2.4Ghz processor with 16 GB RAM. If one compares the two variants of the normal+GPD with bulk model based of parameterized tail fraction, one will notice that the difference in computing time due to estimating the additional tail fraction parameter is inconsequential. The reason for the massive increase in computing time for the two-tailed GPD+normal+GPD model is that the profile likelihood grid search is carried out over the outer product of all combinations of the upper and lower threshold. For the one tailed model there is a sequence of 19 thresholds under consideration, which increases to $19^2 = 361$. The high computational cost for the grid search of the thresholds motivates investigation of alternative inference approaches.

The semi-parametric and non-parametric bulk models listed in Section 2.2 are more computational complex than the parametric alternatives. The semi-parametric mixture of gammas

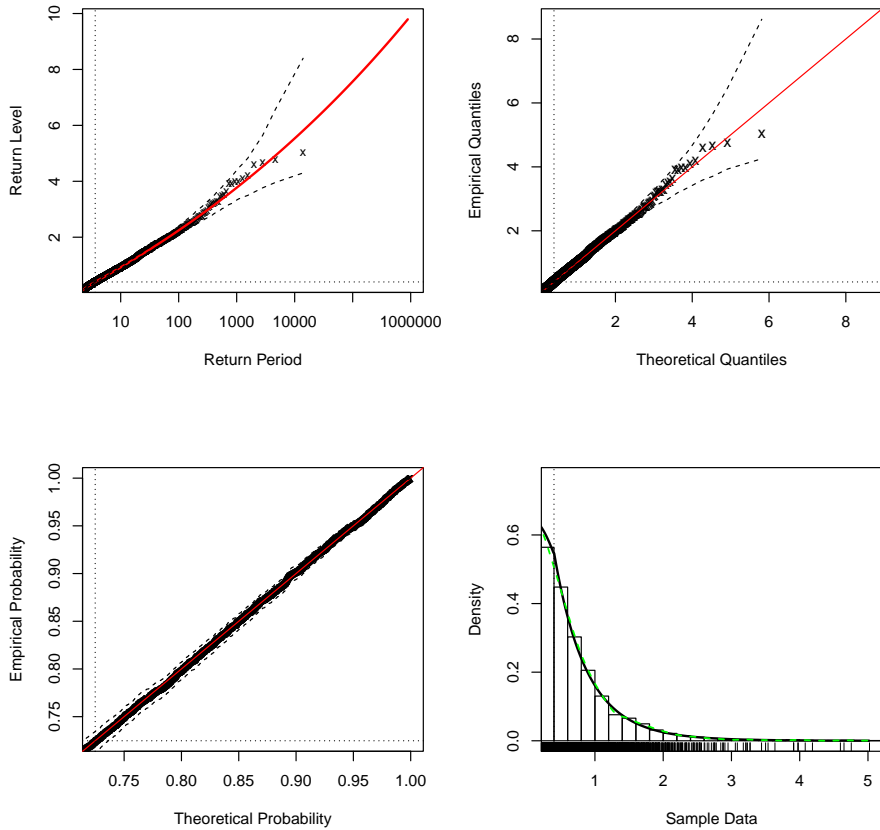


Figure 11: Standard model fit diagnostics for the upper tail of the bulk model based tail fraction GPD+normal+GPD model fitted to the Standard & Poor’s daily log returns dataset.

| Model | u_r | $\hat{\xi}_r$ | log-likelihood | # parameters | Computing time |
|-------------------------|-------|---------------|----------------|--------------|----------------|
| normal+GPD (bulk) | 0.2 | 0.020 (0.018) | -8208.56 | 5 | 12 |
| normal+GPD (parametric) | 0.2 | 0.020 (0.018) | -8053.55 | 6 | 13 |
| GPD+normal+GPD (bulk) | 0.4 | 0.063 (0.024) | -8030.58 | 8 | 634 |

Table 2: Estimated upper tail threshold and shape parameters (and standard errors) for the mixture models fitted to the Standard & Poor’s daily log returns. The log-likelihood, number of parameters and computing time in seconds for each model, where the threshold is estimated using a profile likelihood grid search.

bulk model utilizes the EM algorithm, which involves significant computing time. The non-parametric KDE bulk model also suffers from the computing time required to calculate the cross-validation likelihood in Equation 9. An open research question for all of the extreme value mixture models with a non-parametric bulk model, is how to measure the complexity of the fitted distribution. It is not appropriate, for example, to assume that a KDE only uses one degree of freedom for the bandwidth λ parameter. In the non-parametric density estimation literature (Wand and Jones 1995) they use the concept of “equivalent degrees of freedom” to measure complexity, but there appears to have been no attempt to adapt this approach to mixtures of spliced densities.

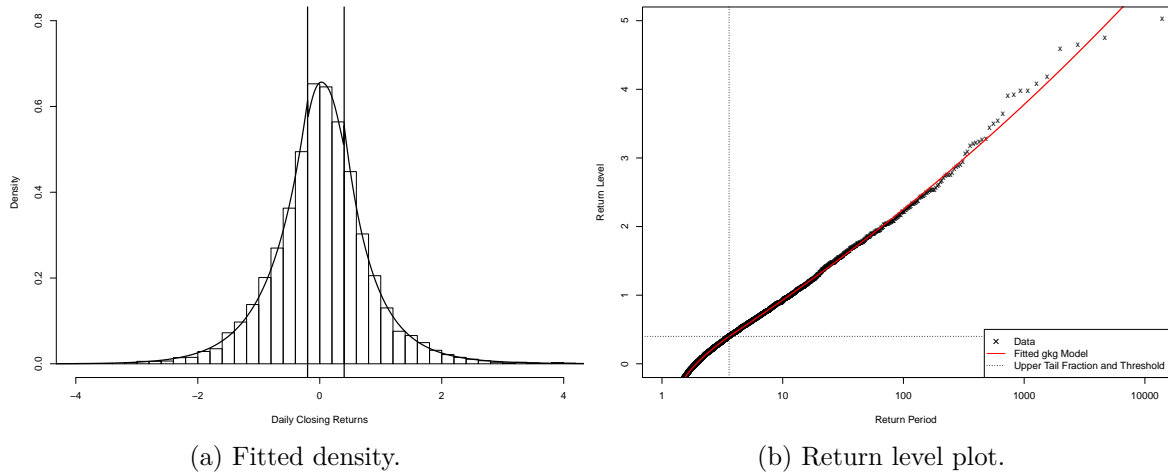


Figure 12: Density histogram of the Standard & Poor's daily log returns dataset `spto87` from the `evir` package overlaid with the fitted GPD+KDE+GPD with bulk model based tail fraction density (left plot) and the return level plot (right plot).

To reduce the computations for the GPD+KDE+GPD model (standard KDE between the two thresholds) and to demonstrate the “fixed threshold approach”, the following code fixes the thresholds to the values estimated from the GPD+normal+GPD model above. Computations are also saved using `alpha = NULL` in the return level plot function, which avoids the calculation of the Monte Carlo uncertainty intervals. The fitted density and return level plot are shown in Figure 12. The fitted density has a similarly good fit to that observed for the GPD+normal+GPD model in Figure 10. The shape parameter estimate for the upper tail is $\xi_r = 0.063$ (0.024). The upper tail fit depicted in the return level plot is also very good.

```
R> fit.gkg <- fgkg(spto87, ulseq = fit.gng$ul, urseq = fit.gng$ur,
+   fixedu = TRUE)
R> hist(spto87, freq = FALSE, breaks = 50, xlab = "Daily Closing Returns",
+   xlim = c(-4, 4), ylim = c(0, 0.8))
R> fx <- with(fit.gkg, dgkg(xp, spto87, lambda, ul, sigmaul, xil,
+   phiul = TRUE, ur, sigmaur, xir, phiur = TRUE))
R> lines(xp, fx)
R> abline(v = c(fit.gkg$ul, fit.gkg$ur))
R> rlplot(fit.gkg, alpha = NULL, rplim = c(1, 11000), rllim = c(0, 5))
```

5. Some parting advice on mixture models

Hu (2013) created the first version of the `evmix` package as part of his Master's thesis along with a detailed simulation study comparing the performance of many of the aforementioned extreme value mixture models for approximating the tails of a wide variety of population distributions. The performance in estimating various tail quantiles was evaluated using the root mean squared error (RMSE). The broad conclusions from this study provide some useful advice, in addition to that drawn out from the examples in this paper:

- The estimates of very high quantiles (e.g., 99.9% for samples of size 1000 and 5000) are often very similar across sensible extreme value mixture models, whereas the lower quantiles (e.g., 90–99%) can be very sensitive to the model specification.
- The bulk model based tail fraction $\phi_u = 1 - H(u|\boldsymbol{\theta})$ should be used if the bulk model is known to be the same as the population distribution, or the sample size is sufficient to adequately assess that the bulk model is a good fit. In this way, the tail fraction estimate benefits from the usually ample bulk data compared to the limited tail data.
- Conversely, in the more usual situation of an unknown population distribution, a poor bulk model or where the fit of bulk model cannot be adequately assessed, it is safest to use the parameterized tail fraction approach as the tail estimates are more robust to the bulk model fit (as seen in Section 4).
- The constraint of continuity at the threshold provides little, if any, benefit for estimation of the highest quantiles. There is some evidence of a benefit for lower quantiles, near the threshold, when the bulk and tail models are a good fit overall.
- However, this continuity constraint has a detrimental effect if the bulk model (or tail model) is poor, as the tail estimation is less robust to the bulk fit (and vice-versa). Hence, it is generally safest not to include a constraint of continuity at the threshold, especially in the usual situation when the population distribution is unknown and the sample size is insufficient for the bulk and tail fits to be properly evaluated.
- The usual advice for choosing between parametric versus semi-/non-parametric estimators holds for extreme value mixture models. For example, if the bulk model is the same as the population distribution or is a good approximation then a parametric bulk model will provide the best tail estimator. However, in the more usual situation of an unknown population distribution the non-parametric extreme value mixture models provide similar overall levels of performance in terms of tail estimation, but of course they add to the computational complexity and care has to be taken to avoid over-fitting.
- The lack of tail fraction scaling in the hybrid Pareto leads to poor performance in wide applications, due to the domination of the GPD tail over the bulk model (see `help("fhpd")` for an example). When developing extreme value mixture models, do not forget that the GPD is a conditional model so a tail fraction scaling is needed to make it unconditional.

There are some limitations in this simulation study which are under further investigation, but are not likely to strongly change these rather intuitive conclusions. Firstly, the model was fitted using maximum likelihood estimation based on the complete likelihood, with the initial value for the threshold set as the 90% quantile (or 10% quantile for lower tail). As mentioned in Section 2.3, the likelihood for these mixture models frequently has local modes so this is sub-optimal. Profile likelihood estimation of the threshold can be used to ameliorate this issue, see Section 4. Hu (2013) compared some of the likelihood based simulation results to those using Bayesian inference with posterior sampling via Markov chain Monte Carlo, which does not suffer from the same optimization challenges, but reached with the same broad conclusions.

The use of the RMSE as the measure of performance ignores the potential for over-fitting, as there is no penalty for the degrees of freedom used in the model fit. This provides a slight advantage to the parameterized tail fraction approach and variants of the mixture models with no continuity constraints on the PDF at the threshold. The non-parametric and semi-parametric extreme value mixture models also benefit from extra degrees of the freedom. However, this performance gain is likely to be small as the degrees of freedom are expended on the bulk component, not on the tail(s) where the performance is being evaluated.

5.1. Obtaining the package and guides

The latest package is released under the GPL-3 license and can be downloaded from the Comprehensive R Archive Network (CRAN) at <https://CRAN.R-project.org/package=evmix>. This manuscript is included as a vignette in the package and further resources are available on the package website <http://www.math.canterbury.ac.nz/~c.scarrott/evmix/>.

Acknowledgments

Some of the code for the package were originally written for MATLAB ([The MathWorks Inc. 2017](#)) by Anna MacDonald and Xin Zhao. Clement Lee and Emma Eastoe receive thanks for many helpful discussions about these models and feedback on the package. In particular, they recommended the implementation of profile likelihood estimation of the threshold. The authors (Alec Stephenson, Janet Heffernan and Eric Gilleland) of the *evd* ([Stephenson 2002](#)) and *ismev* ([Heffernan and Stephenson 2016](#)) packages are also thanked as some aspects of the GPD and graphical diagnostic code are based around their functions. We would like to thank the reviewers, whose insights substantially improved this paper.

References

- Behrens CN, Lopes HF, Gamerman D (2004). “Bayesian Analysis of Extreme Events with Threshold Estimation.” *Statistical Modelling*, **4**(3), 227–244. doi:10.1191/1471082x04st075oa.
- Beirlant J, Dierckx G, Guillou A, Staarica C (2002). “On Exponential Representations of Log-Spacings of Extreme Order Statistics.” *Extremes*, **5**(2), 157–180. doi:10.1023/a:1022171205129.
- Cabras S, Castellanos ME (2011). “A Bayesian Approach for Estimating Extreme Quantiles Under a Semiparametric Mixture Model.” *ASTIN Bulletin*, **41**(1), 87–106. doi:10.2143/AST.41.1.2084387.
- Carreau J (2012). *condmixt: Conditional Density Estimation with Neural Network Conditional Mixtures*. R package version 1.0, URL <https://CRAN.R-project.org/package=condmixt>.
- Carreau J, Bengio Y (2009). “A Hybrid Pareto Model for Asymmetric Fat-Tailed Data: The Univariate Case.” *Extremes*, **12**(1), 53–76. doi:10.1007/s10687-008-0068-0.

- Chen SX (1999). “Beta Kernel Estimators for Density Functions.” *Computational Statistics & Data Analysis*, **31**(2), 131–145. doi:[10.1016/s0167-9473\(99\)00010-9](https://doi.org/10.1016/s0167-9473(99)00010-9).
- Chen SX (2000). “Probability Density Function Estimation Using Gamma Kernels.” *The Annals of the Institute of Statistical Mathematics*, **52**(3), 471–480. doi:[10.1023/a:1004165218295](https://doi.org/10.1023/a:1004165218295).
- Coles SG (2001). *An Introduction to Statistical Modelling of Extreme Values*. Springer Series in Statistics. Springer-Verlag. doi:[10.1007/978-1-4471-3675-0](https://doi.org/10.1007/978-1-4471-3675-0).
- Danielsson J, De Haan L, Peng L, De Vries CG (2001). “Using a Bootstrap Method to Choose the Sample Fraction in Tail Index Estimation.” *Journal of Multivariate Analysis*, **76**(2), 226–248. doi:[10.1006/jmva.2000.1903](https://doi.org/10.1006/jmva.2000.1903).
- Diggle PJ (1985). “A Kernel Method for Smoothing Point Process Data.” *Journal of the Royal Statistical Society C*, **34**(2), 138–147. doi:[10.2307/2347366](https://doi.org/10.2307/2347366).
- do Nascimento FF, Gamerman D, Lopes HF (2012). “A Semiparametric Bayesian Approach to Extreme Value Estimation.” *Statistics and Computing*, **22**(2), 661–675. doi:[10.1007/s11222-011-9270-z](https://doi.org/10.1007/s11222-011-9270-z).
- Drees H, Kaufmann E (1998). “Selecting the Optimal Sample Fraction in Univariate Extreme Value Estimation.” *Stochastics Processes and their Application*, **75**(2), 149–172. doi:[10.1016/s0304-4149\(98\)00017-9](https://doi.org/10.1016/s0304-4149(98)00017-9).
- Duin RPW (1976). “On the Choice of Smoothing Parameters for Parzen Estimators of Probability Density Functions.” *IEEE Transactions on Computing*, **C-25**(11), 1175–1179. doi:[10.1109/tc.1976.1674577](https://doi.org/10.1109/tc.1976.1674577).
- Duong T (2015). *ks: Kernel Smoothing*. R package version 1.9.4, URL <https://CRAN.R-project.org/package=ks>.
- Dupuis DJ (2000). “Exceedances over High Thresholds: A Guide to Threshold Selection.” *Extremes*, **1**(3), 251–261. doi:[10.1023/a:1009914915709](https://doi.org/10.1023/a:1009914915709).
- Finkenstädt B, Rootzén H (eds.) (2003). *Extreme Values in Finance, Telecommunications, and the Environment*. Chapman & Hall/CRC.
- Frigessi A, Haug O, Rue H (2003). “A Dynamic Mixture Model for Unsupervised Tail Estimation Without Threshold Selection.” *Extremes*, **5**(3), 219–235. doi:[10.1023/a:1024072610684](https://doi.org/10.1023/a:1024072610684).
- Gasser T, Müller HG (1979). “Kernel Estimation of Regression Functions.” In T Gasser, M Rosenblatt (eds.), *Smoothing Techniques for Curve Estimation*, pp. 23–68. Springer-Verlag.
- Habbema JDF, Hermans J, Van den Broek K (1974). “A Stepwise Discriminant Analysis Program Using Density Estimation.” In G Bruckmann (ed.), *Proceedings of COMPSTAT 1974*, pp. 101–110. Physica-Verlag, Vienna.
- Heffernan JE, Stephenson AG (2016). *ismev: An Introduction to Statistical Modeling of Extreme Values*. R package version 1.41, URL <https://CRAN.R-project.org/package=ismev>.

- Holden L, Haug O (2009). “A Mixture Model for Unsupervised Tail Estimation.” arXiv:0902.4137 [stat.ME], URL <http://arxiv.org/abs/0902.4137>.
- Hu Y (2013). *Extreme Value Mixture Modeling with Simulation Study and Applications in Finance and Insurance*. MSc thesis, University of Canterbury, New Zealand. URL <http://hdl.handle.net/10092/8538>.
- Hu Y, Scarrott CJ (2018). *evmix: Extreme Value Mixture Modelling, Threshold Estimation and Boundary Corrected Kernel Density Estimation*. R package version 2.10, URL <https://CRAN.R-project.org/package=evmix>.
- Jones MC (1993). “Simple Boundary Correction for Kernel Density Estimation.” *Statistics and Computing*, **3**(3), 135–146. doi:10.1007/bf00147776.
- Jones MC, Foster PJ (1996). “A Simple Nonnegative Boundary Correction Method for Kernel Density Estimation.” *Statistica Sinica*, **6**(4), 1005–1013.
- Jones MC, Henderson DA (2007). “Miscellaneous Kernel-Type Density Estimation On the Unit Interval.” *Biometrika*, **94**(4), 977–984. doi:10.1093/biomet/asm068.
- Kooperberg C (2013). *logspline: Logspline Density Estimation Routines*. R package version 2.1.5, URL <https://CRAN.R-project.org/package=logspline>.
- Lee D, Li WK, Wong TST (2012). “Modeling Insurance Claims via a Mixture Exponential Model Combined with Peaks-over-Threshold Approach.” *Insurance: Mathematics and Economic*, **51**(3), 538–550. doi:10.1016/j.insmatheco.2012.07.008.
- MacDonald A (2012). *Extreme Value Mixture Modelling with Medical and Industrial Applications*. PhD thesis, University of Canterbury, New Zealand. URL <http://hdl.handle.net/10092/6679>.
- MacDonald A, Scarrott CJ, Lee D, Darlow B, Reale M, Russell G (2011). “A Flexible Extreme Value Mixture Model.” *Computational Statistics & Data Analysis*, **55**(6), 2137–2157. doi:10.1016/j.csda.2011.01.005.
- MacDonald A, Scarrott CJ, Lee DS (2013). “Boundary Correction, Consistency and Robustness of Kernel Densities Using Extreme Value Theory.” Unpublished manuscript.
- Marron JS, Ruppert D (1994). “Transformations to Reduce Boundary Bias in Kernel Density Estimation.” *Journal of the Royal Statistical Society B*, **56**(4), 653–671.
- Mendes B, Lopes HF (2004). “Data Driven Estimates for Mixtures.” *Computational Statistics & Data Analysis*, **47**(3), 583–598. doi:10.1016/j.csda.2003.12.006.
- Nadarajah S (2013). *CompLognormal: Functions for Actuarial Scientists*. R package version 3.0, URL <https://CRAN.R-project.org/package=CompLognormal>.
- Papastathopoulos I, Tawn JA (2013). “Extended Generalised Pareto Models for Tail Estimation.” *Journal of Statistical Planning and Inference*, **143**(1), 131–143. doi:10.1016/j.jspi.2012.07.001.
- Pfaff B, McNeil A (2012). *evir: Extreme Values in R*. R package version 1.7-3, URL <https://CRAN.R-project.org/package=evir>.

- R Core Team (2017). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. URL <https://www.R-project.org/>.
- Scarrott CJ, MacDonald A (2012). “A Review of Extreme Value Threshold Estimation and Uncertainty Quantification.” *REVSTAT – Statistical Journal*, **10**(1), 33–60.
- Schuster EF (1985). “Incorporating Support Constraints into Nonparametric Estimators of Densities.” *Communications in Statistics – Theory and Methods*, **14**(5), 1123–1136. doi: [10.1080/03610928508828965](https://doi.org/10.1080/03610928508828965).
- Solari S, Losada MA (2004). “A Unified Statistical Model for Hydrological Variables Including the Selection of Threshold for the Peak over Threshold Method.” *Water Resources Research*, **48**(10), W10541. doi: [10.1029/2011wr011475](https://doi.org/10.1029/2011wr011475).
- Stephenson AG (2002). “**evd**: Extreme Value Distributions.” *R News*, **2**(2), 31–32.
- Tancredi A, Anderson CW, O’Hagan A (2006). “Accounting for Threshold Uncertainty in Extreme Value Estimation.” *Extremes*, **9**(2), 87–106. doi: [10.1007/s10687-006-0009-8](https://doi.org/10.1007/s10687-006-0009-8).
- The MathWorks Inc (2017). *MATLAB – The Language of Technical Computing, Version R2017a*. Natick. URL <http://www.mathworks.com/products/matlab/>.
- Wand M (2015). **KernSmooth**: *Functions for Kernel Smoothing Supporting Wand & Jones (1995)*. R package version 2.23-14, URL <https://CRAN.R-project.org/package=KernSmooth>.
- Wand MP, Jones MC (1995). *Kernel Smoothing*. Chapman & Hall/CRC, London.
- Zhao X, Scarrott CJ, Reale M, Oxley L (2010). “Extreme Value Modelling for Forecasting the Market Crisis.” *Applied Financial Economics*, **20**(1), 63–72. doi: [10.1080/09603100903262947](https://doi.org/10.1080/09603100903262947).

Affiliation:

Carl Scarrott
School of Mathematics and Statistics
University of Canterbury
Christchurch 8140, New Zealand
E-mail: carl.scarrott@canterbury.ac.nz
URL: <http://www.math.canterbury.ac.nz/~c.scarrott/>