Extreme value modelling of dependent series using R

Janet E. Heffernan and Harry Southworth

September 28, 2016

1 Introduction

This document illustrates the use of the `texmex` package, [6] for performing declustering of dependent data, and subsequent extreme value analysis in `R`, [5]. We use the automatic declustering algorithm of Ferro and Segers (2003), [2]. This involves the estimation of the extremal index of a sequence of excesses above a threshold, prior to the identification of independent clusters of excesses above that threshold. Following the identification of such clusters, the Generalised Pareto Distribution (GPD) tail model may be fitted to independent cluster maxima.

To cite this vignette, refer to Vignette name: `declustering` and use the package citation:

```
##
## To cite package 'texmex' in publications use:
##
##   Harry Southworth, Janet E. Heffernan and Paul D. Metcalfe
##   (2016). texmex: Statistical modelling of extreme values. R
##   package version 2.3.
##
## A BibTeX entry for LaTeX users is
##
##   @Manual{,
##     title = {texmex: Statistical modelling of extreme values},
##     author = {Harry Southworth and Janet E. Heffernan and Paul D. Metcalfe},
##     year = {2016},
##     note = {R package version 2.3},
##   }
```

1.1 Preliminaries

First install the `texmex` package. Depending on your installation of R, this can be done using the `install.packages` command in R, or by downloading the

package from CRAN and installing it from a local archive.

Once `texmex` is installed, use the `library` command to make the package available to the current session.

```
library(texmex)
palette(c("black","purple","cyan","orange"))
set.seed(20120118)
```

The final command sets the random seed so that the results in this vignette may be reproduced exactly.

1.2 texmex

The `texmex` package for R was written by Harry Southworth and Janet E. Heffernan. The work was funded by AstraZeneca. This vignette focusses on the functions in `texmex` which carry out cluster analysis of serially dependent data. More detailed information on the remainder of the `texmex` package is given in the texmex1d package vignette.

2 Statistical inference for clusters of extreme values

We summarise here the approach put forward by Ferro and Segers (2003) [2]. The full details are given in the cited paper, but we highlight the main points to be considered in taking this approach. For clarity, we adopt the same notation as that used in the original exposition.

We denote by $\{\xi_n : n = 1, 2, \ldots\}$, a strictly stationary sequence of random variables with marginal distribution function F, and tail function $\overline{F}$. F may have a finite upper end point or it may be a distribution with an infinite upper tail. We define M_n as the maximum of the first $n \geq 1$ observations of the process.

We are concerned with the behaviour of excesses of $\{\xi_n : n = 1, 2, \ldots\}$ above thresholds. As with all Extreme Value statistical modelling, the motivation behind the model comes from the limiting behaviour of extreme values of the process as we look further and further into the tails. Here, the limiting operation involves the threshold above which we observe excesses. This threshold is increased to the upper limit of the support of F and the limiting form of dependence between excesses gives the form of model that will be adopted for subsequent statistical analysis.

Dependence in the original sequence of random variables may or may not persist into the tails of their joint distribution. If the dependence does persist then the excesses of high thresholds can tend to cluster together in the limit. If such clustering occurs then the variables within a given cluster cannot be considered independent. This complicates inference about threshold exceedances

constructed from a sequence which exhibits such extremal dependence. However, clusters of exceedances can be considered to be independent and a common approach involves the identification of such clusters, and inference on the characteristics of these clusters, or on independent cluster maxima, see Coles [1] Chapter 5 for more details.

2.1 The extremal index

The extremal index is a measure of the extent to which threshold exceedances from the sequence $\{\xi_n : n = 1, 2, \ldots\}$ cluster in the limit as the threshold tends to the upper endpoint of the support of F. Its formal definition is as follows (see Leadbetter, 1983 [3]). The sequence $\{\xi_n : n = 1, 2, \ldots\}$ has an *extremal index* $\theta \in [0, 1]$ if for each $\tau > 0$ there is a sequence $\{u_n : n = 1, 2, \ldots\}$ for which both:

$$\begin{aligned} n\overline{F}(u_n) &\rightarrow \tau \\ \text{and } P(M_n \leq u_n) &\rightarrow \exp(-\theta\tau) \end{aligned}$$

as $n \rightarrow \infty$. The first condition here ensures that u_n increases at an appropriate rate to find the dependence structure in the process; the second shows that the growth of process maxima is mitigated by any dependence within the sequence. Values of $\theta < 1$ indicate a tendancy of threshold exceedances to cluster together as the threshold approaches its limit, whereas if $\theta = 1$ then threshold excesses occur in isolation in the limit. The extremal index, when it exists, characterises the size of clusters of threshold exceedances, having the interpretation as one over the mean cluster size (see [3]).

2.2 Estimation of the Extremal Index

In the development of their *intervals estimator* of the extremal index, Ferro and Segers focus on the times between consecutive threshold exceedances, the *interexceedance times*. The estimator is motivated by the observation that interexceedance times can arise in either of two ways:

inter-cluster times consecutive exceedances occur in different but adjacent clusters;

intra-cluster times consecutive exceedances are adjacent observations occuring in the same cluster.

Inter-cluster times are necessarily larger in distribution than intra-cluster times. The *intervals estimator* of Ferro and Segers models the distribution of all exceedance times as a mixture of inter-cluster and intra-cluster times, and identifies the two components of this mixture distribution as follows.

Let $T(u)$ be a random variable having the same distribution as the interexceedance times associated with threshold u:

$$\min\{n \geq 1 : \xi_{n+1} > u\} \text{ given that } \xi_1 > u.$$

Ferro and Segers show that as the threshold u tends to the upper limit of the support of F,

$$\overline{F}(u)T(u) \xrightarrow{\mathrm{D}} T_\theta, \tag{1}$$

where $\xrightarrow{\mathrm{D}}$ denotes convergence in distribution and the random variable T_θ follows the mixture distribution:

$$(1-\theta)\epsilon_0 + \theta\mu_\theta.$$

Here ϵ_0 is the degenerate distribution with point mass at zero, and μ_θ is the Exponential distribution with mean $1/\theta$. The extremal index has a dual purpose here: θ is the limiting proportion of non-zero interexceedance times (the proportion of interexceedance times which are also inter-cluster times); it is also one over the mean of the non-zero interexceedance times. These properties are exploited to obtain a moment estimator of the extremal index θ, the *intervals estimator*, based on equation (1). In practice, the estimation is carried out by using a fixed threshold, which is chosen at a suitably high level. We discuss the choice of this threshold in Section 2.5.

2.3 Cluster identification

The identification of clusters makes use of the characterisation of interexceedance times as either *inter-cluster* or *intra-cluster*. Under the Ferro and Segers model, the extremal index θ arises as the proportion of interexceedance times which are also inter-cluster times. This means that interexceedance times can be easily categorised in these two groups as a natural consquence of this model. Assuming that we have observed N threshold exceedances, then the $\lfloor \theta N \rfloor$ largest interexceedance times are assumed to be approximately independent inter-cluster times, which separate the remaining exceedance times into intra-cluster times. In practice, the estimated value of θ is used to identify the critical cut-off interexceedance time that distinguishes inter- from intra-cluster times.

Once clusters have been identified, the cluster characteristics can be examined, or the original threshold exceedances can be thinned to give approximately independent cluster maxima. Inference on these cluster maxima can then be carried out – such inference is much more straightforward than inference on the original dependent sequence, although arguably this approach is wasteful of information as it discards all but the largest observation in each cluster. We show an example of such an analysis in Section 2.6.

2.4 Estimation uncertainty

The uncertainty in the estimation of θ and in the subsequent cluster identification is estimated using a bootstrap algorithm. This algorithm maintains the within-cluster dependence structure, and exploits the independence between clusters by resampling clusters and inter-cluster times rather than individual observations from the original dependent sequence. More specifically:

1. resample with replacement from the set of $C - 1$ observed inter-cluster times;

2. resample with replacement from the set of C observed clusters (obtaining both intra-cluster exceedance times and the sizes of associated threshold excesses);

3. interpose the interexceedance times and clusters to obtain a bootstrap sample of the process of threshold excesses and their times of occurrance;

4. estimate θ and obtain N for the bootstrap process and decluster using these values;

5. estimate cluster characteristics or find cluster maxima and carry out inference on these.

The resulting estimates of θ and any other parameters associated with cluster characteristics represent a boostrap estimate of the sampling distribution of these estimators.

The bootstrap procedure described above is that given in Ferro and Segers' original paper. This is sufficient for estimation of uncertainty of $\hat{\theta}$ and many other cluster characteristics such as mean cluster size. However, in the case where we go on to decluster the original series and fit the GPD model to cluster maxima, the bootstrap procedure as it stands has a particular shortcoming. Estimates of the GPD parameters (σ, ξ) – the scale and shape parameter respectively – are very sensitive to the largest values in the data. Specifically, if there are ties among the largest observed data points, then this is taken as strong evidence of a finite upper end point close to these largest values and resulting estimates of the shape parameter can be severely biased downwards. For this reason, the simple non-parametric bootstrap of the threshold excesses within each cluster is not appropriate for estimating uncertainty of GPD model parameter estimates. Ties arise frequently in this resampling-with-replacement approach.

The implementation of the bootstrap in `texmex` provides a simple alteration to the non-parametric bootstrap described above in the case where the GPD parameters are to be estimated from cluster maxima. Prior to carrying out the bootstrap procedure, the original series is declustered and the GPD model estimated from the original cluster maxima. Then the bootstrap is carried out as above but with an addition to step 5 which becomes:

5. find cluster maxima; replace sampled cluster maxima with an independent sample of the same size from the GPD with estimated model parameters obtained from the original cluster maxima; carry out inference on these (i.e. estimate GPD model for these simulated cluster maxima).

Thus the bootstrap procedure in this case is semi-parametric: we sample from the empirical distribution of interexceedance times and from the fitted parametric distribution describing the cluster maxima.

2.5 Threshold selection

As mentioned in Section 2.2, the model adopted for inference about the extremal index and subsequent declustering arises in the limit as the threshold attains the upper end point of the distribution F. Whilst this is the case, to carry out this inference, the threshold must be set at a finite level, balancing opposing pulls on the value which it takes. The threshold should be high enough that the underlying distribution is well approximated by its limiting form. It should preferrably also be sufficiently low that we can use observed threshold excesses to estimate the model parameters with some precision. There are two diagnostic tools which we put forward to aid the selection of a suitable threshold, both of which exploit the properties of the Ferro and Segers model in Equation 1.

2.5.1 Quantile-quantile plot of normalised interexceedance times

The distribution of normalised interexceedance times in Equation 1 is a mixture distribution. Having extimated $\hat{\theta}$ by using the intervals estimator, we can check the goodness of fit of this underlying model: $\hat{\theta}$ estimates the proportion of interexceedance times which correspond to inter-cluster times and whose normalised values should be well approximated by the Exponential distribution with mean $1/\hat{\theta}$. The remaining normalised interexceedance times are intra-cluster times and should be small relative to the inter-cluster times.

Ferro and Segers propose a quantile-quantile (Q-Q) plot of observed normalised interexceedance times against standard Exponential quantiles to diagnose model fit. Its appearance should be slightly different from a standard Q-Q plot (which shows the data hugging a straight line in the case of good model fit). In our case, since the underlying limit model is a mixture of degeneracy at zero and the Exponential distribution, we look instead for a broken stick shape. The $(1-\hat{\theta})$ quantile is indicated with a vertical line: above this line we look for the observed and theoretical quantiles hugging a straight line with gradient $1/\hat{\theta}$ (also marked); below this line we look for a sudden attenuation of the observed times close to zero. This is demonstrated in Figure 1, which mimics Figure 1(a) in the original paper.

2.5.2 Threshold stability plot

We exploit another useful property of the extremal index estimator to derive a further tool for threshold selection. This property is that of *threshold stability*, which is used in threshold selection for many other extreme value models. More details of its use in other models are are given in the texmex1d package vignette.

The threshold stability of the extremal index estimator refers to its invariance to change in threshold above a suitably high threshold. Simply put, once a threshold is *high enough*, raising the threshold further should not dramatically change the estimated value of θ. As the threshold increases, the sample size used for estimation will fall so sampling uncertainty will increase, but the estimated values of of $\hat{\theta}$ should not alter above this anticipated sampling variation.

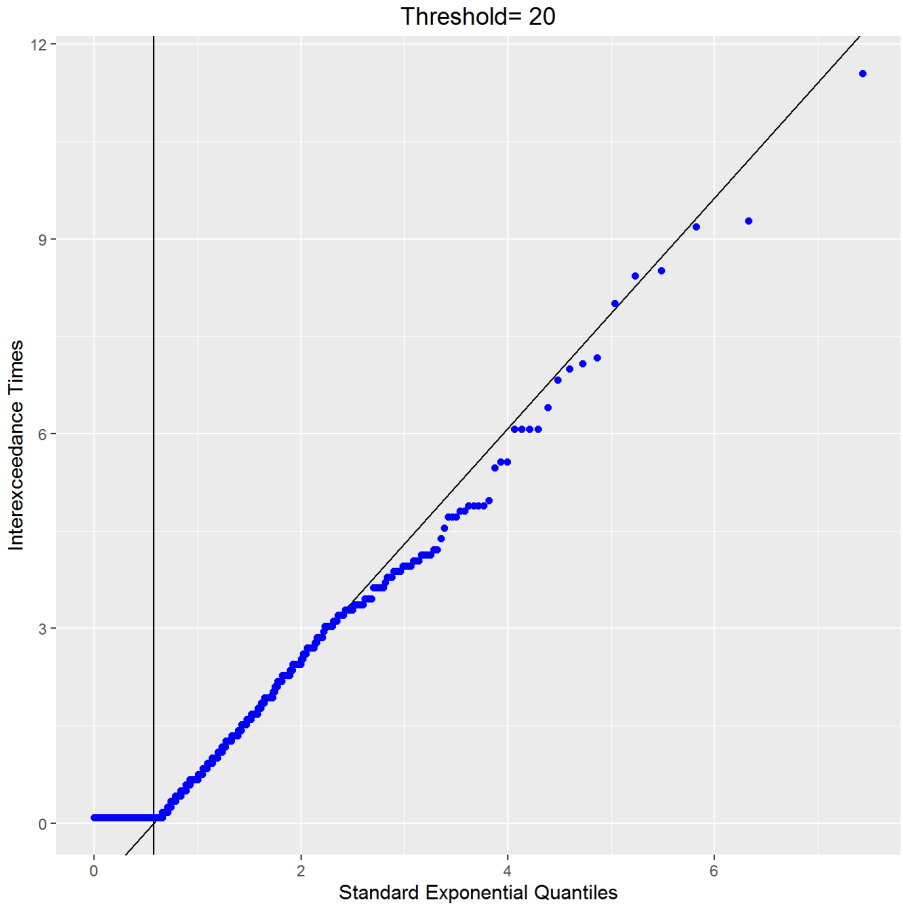


Figure 1: Quantile-quantile plot of normalised interexceedance times against standard exponential quantiles. Vertical line shows the $(1-\hat{\theta})$ quantile; sloping line has gradient $1/\hat{\theta}$. Data are simulated from a max-autoregressive process with extremal index $\theta = 0.2$.

Any apparent trend in the estimated θ with threshold is an indication that the threshold is too low.

The *threshold stability plot* examines a range of thresholds for invariance of $\hat{\theta}$ to change in threshold. At each threshold spanning a range of thresholds, the extremal index is calculated and a confidence interval estimated by using the bootstrap described in Section 2.4. These are plotted against threshold. The plot is used to choose the lowest threshold above which estimates of θ are approximately constant.

If the data are to be declustered and GPD models fitted to resulting cluster maxima, then it can also be useful to plot the estimated GPD parameters against threshold in an analogous manner. Confidence intervals for the GPD parameters calculated by using the bootstrap scheme described in Section 2.4 will reflect uncertainty due to declustering. A suitable threshold for declustering and GPD estimation will exhibit the threshold stability property for both the extremal index θ and the parameters of the GPD.

2.6 Data analysis in R using `texmex`

We now demonstrate how to carry out the above analysis in R using the implementation in the package `texmex`. Further details of function usage are given in the package documentation, accessed using `help(texmex)`.

2.7 Daily rainfall series

We work here with the `rain` data available in the `texmex` package. This is a series of daily rainfall observations collected at a location in the the south-west of England over 1914 – 1962.

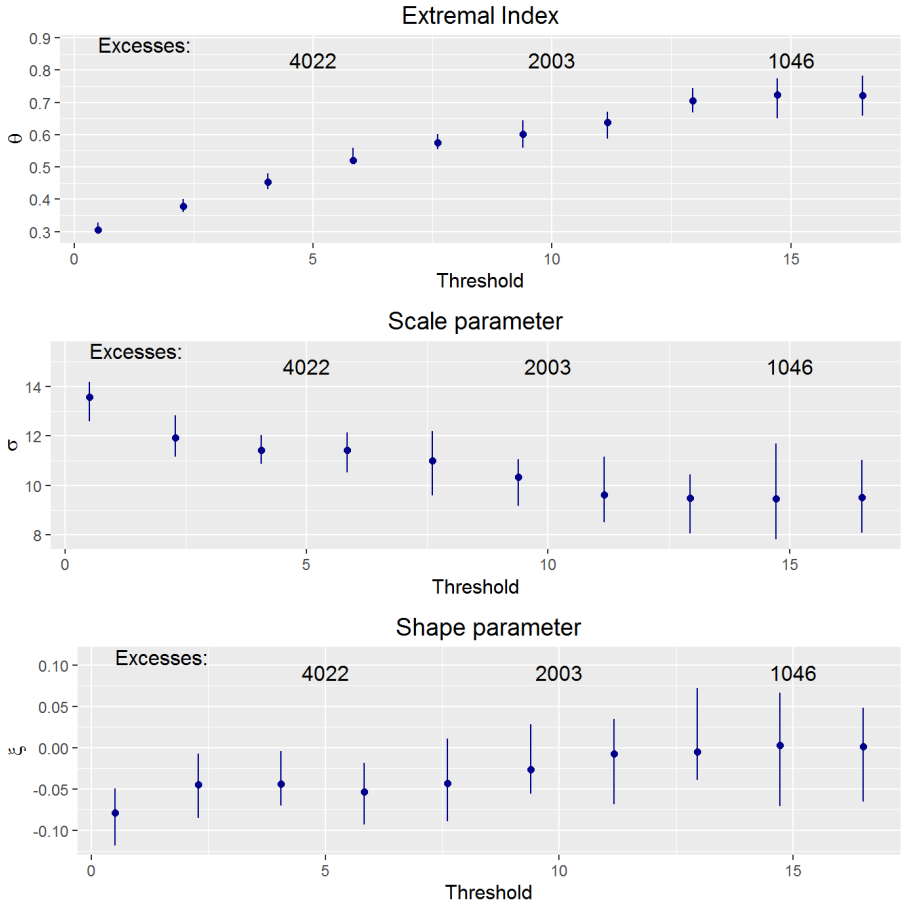
To plot the raw rainfall data (not shown):

```
ggplot(data=data.frame(rain=rain,index=1:length(rain)),
       aes(index,rain)) + geom_point(alpha=0.5,col=4)
```

Extremal index estimation and declustering

We begin by looking at the *threshold stability plot* for the extremal index and (by default) also the parameters of the GPD fitted to cluster maxima.

```
erf <- extremalIndexRangeFit(rain,nboot=20,verbose=FALSE)
p <- ggplot(erf)
gridExtra::grid.arrange(p[[1]],p[[2]],p[[3]],ncol=1)
```

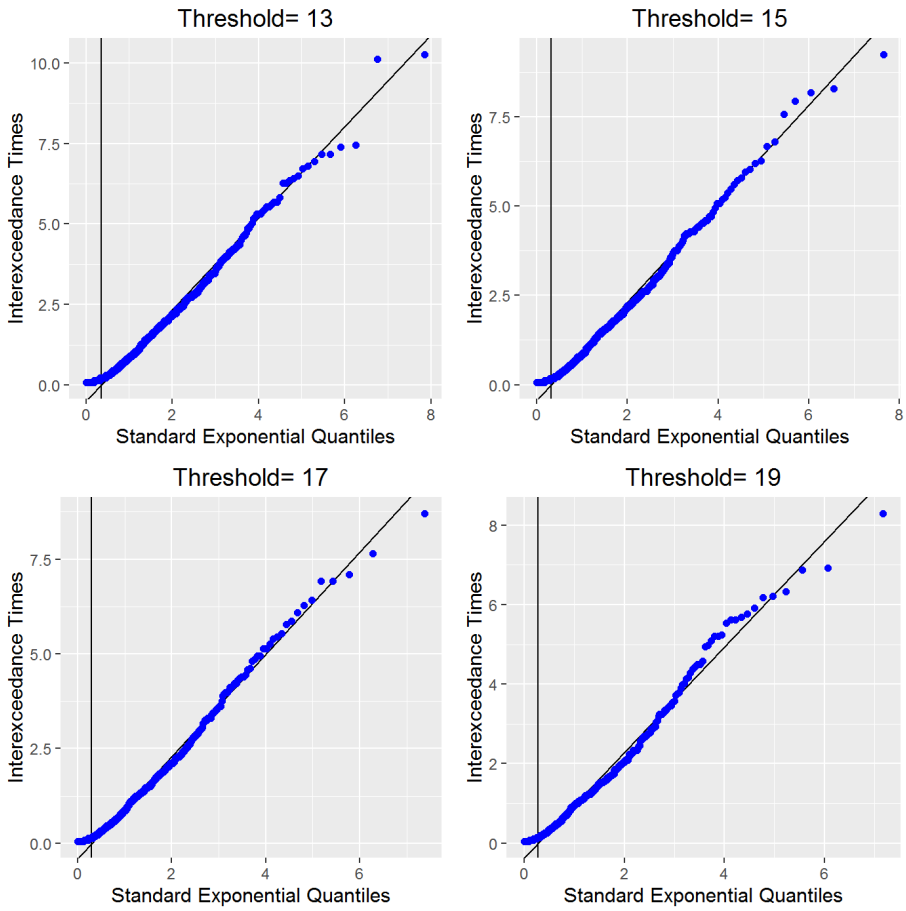


We can see from this plot that while the estimated parameters of the GPD are relatively stable to the choice of threshold used for declustering, the estimates of the extremal index $\hat{\theta}$ are not. Lower thresholds have smaller values of $\hat{\theta}$ so that clusters above lower thresholds are larger than those occurring above higher thresholds. Since we are interested in finding the clustering tendancies of the very highest values, we should choose the value of threshold for which the estimates of all three parameters are approximately constant. This leads to a threshold choice of 13mm or above .

We check the goodness of fit of our model for cluster occurrance by using the Q-Q plot descibed in Section 2.5.1.

```
ei <- extremalIndex(rain,threshold=13)
g1 <- ggplot(ei)
ei <- extremalIndex(rain,threshold=15)
g2 <- ggplot(ei)
ei <- extremalIndex(rain,threshold=17)
g3 <- ggplot(ei)
```

```
ei <- extremalIndex(rain,threshold=19)
g4 <- ggplot(ei)
gridExtra::grid.arrange(g1,g2,g3,g4,ncol=2)
```



We can be reassured by these plots that a threshold of 13mm gives a good fit of the model to the rainfall data. For this choice of threshold, the estimate of the extremal index is about 0.7, so that the average cluster size is $1/0.7 = 1.4$. This is telling us that rainfall tends to be heavy on consecutive days but very rainy spells tend not to last longer than 1 or 2 days.

We now proceed to decluster the sequence by using the automatic declustering method described in Section 2.3.

```
ei <- extremalIndex(rain, threshold=13)
ei

##
## Length of original series 17531
```

```
## Threshold 13
## Number of Threshold Exceedances 1294
## Intervals estimator of Extremal Index 0.7046188

dc <- declust(ei)
dc

## declust.extremalIndex(y = ei)
##
## Threshold  13
## Declustering using the intervals method, run length 3
## Identified 817 clusters.
```

There is a `ggplot` method for this output, although this is not shown here.

Alternatively the above steps can be carried out in a single call to the function `declust`, passing the original data.

```
dc <- declust(rain, threshold=13)
```

So of the original series which is of length 17531, there are 1294 exceedances of the threshold 13mm. However, we have identified serial dependence in the data, so the threshold excesses are not independent and in fact correspond to only 817 approximately independent clusters.

Fitting the Generalised Pareto distribution to cluster maxima

We can now go on to estimate the parameters of the Generalised Pareto Distribution used to describe the conditional distribution of a cluster maximum given that it exceeds the threshold used for declustering. In `texmex` we can use the declustered series directly in a call to the Extreme Value Model fitting routine `evm`, using the default family GPD as follows:

```
rain.gpd <- evm(dc)
rain.gpd

## Call: evm.declustered(y = dc)
## Family:       GPD
##
## Model fit by maximum likelihood.
##
## Convergence: TRUE
## Threshold: 13
## Rate of excess: 0.0466
##
##   Log. lik   AIC
##   -2692.660  5389.319
```

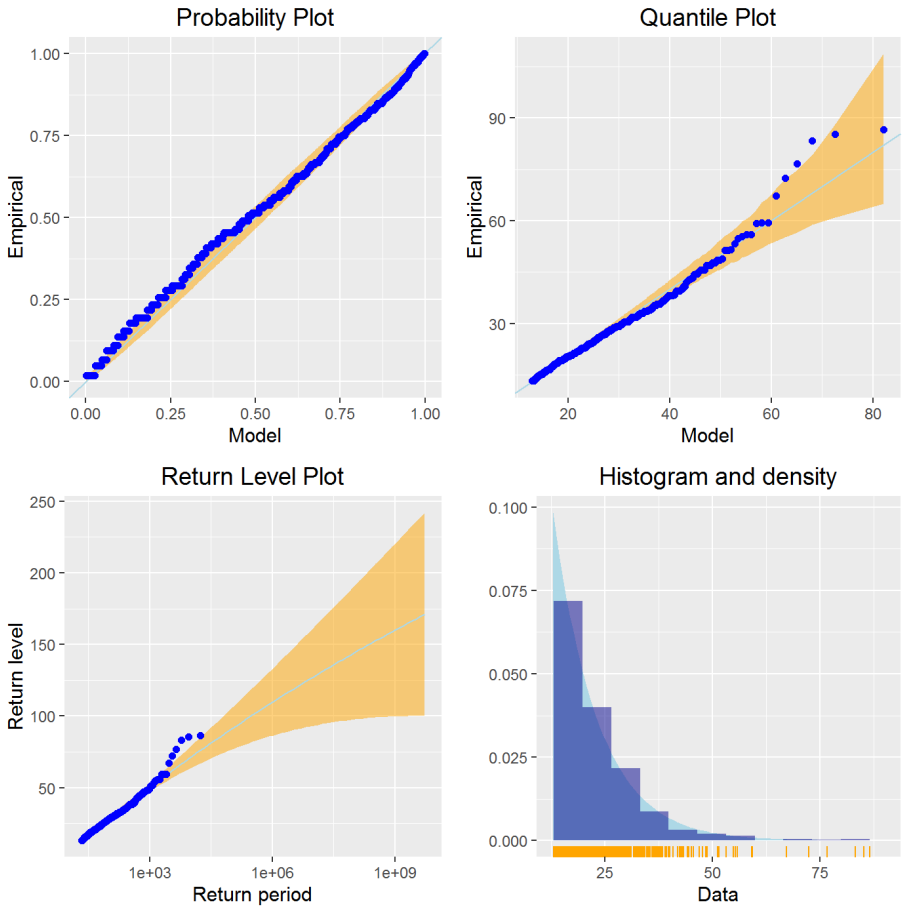
```
##
##
## Coefficients:
##        Value     SE
## phi:    2.31937   0.04524
## xi:    -0.02353   0.02871
```

Here, the `rate of excess` refers to the rate at which the cluster maxima occur in the original series. Using this threshold of 13mm there are 817 clusters so the rate of occurrance is given by:

```
dc$nCluster

## [1] 817

length(rain)

## [1] 17531

dc$nCluster / length(rain)

## [1] 0.04660316
```

We now look a the diagnostic plots to check the fit of the GPD:

```
ggplot(rain.gpd)
```



These plots give minor cause for concern: a couple of data points at the top of the Q-Q plot lie sightly outside the tolerance intervals for this plot. This slight deviation from model fit is not improved particularly by increasing the threshold (not shown).

We can compare the parameter estimates of the GPD model fitted to the cluster maxima with those obtained by fitting the GPD to the original series:

```
evm(rain,th=13)

## Call: evm(y = rain, th = 13)
## Family:       GPD
##
## Model fit by maximum likelihood.
##
## Convergence: TRUE
## Threshold: 13
## Rate of excess: 0.07381
```

```
##
##   Log. lik   AIC
##   -4034.082  8072.164
##
##
## Coefficients:
##        Value    SE
## phi:   2.10088  0.03709
## xi:    0.01682  0.02457
```

The point estimates are not significantly different although the estimated standard errors are slightly larger for the model fitted to cluster maxima. The key difference between the fits is that the independence assumption underlinning the fitting of both of these models is not met in the case where the GPD is fitted to all threshold exceedances, whereas the assumption that cluster maxima are indpendent is satisfied.

The rate of excess in the GPD fitted to the whole series refers to the rate at which the original series exceeds the threhsold of 13mm. Note that the AIC values for these two fitted models are not comparable since they are fitted to different sets of data.

If required, the function `gpd` may be called by specifying a penalty or prior information on model parameters, or estimation may be carried out by using Markov Chain Monte Carlo. For example to simulate from the posterior distribution of the parameters, rather than the default estimation by using (penalised) maxium likelihood, we can do:

```
rain.mcmc <- evm(dc,method="simulate")
```

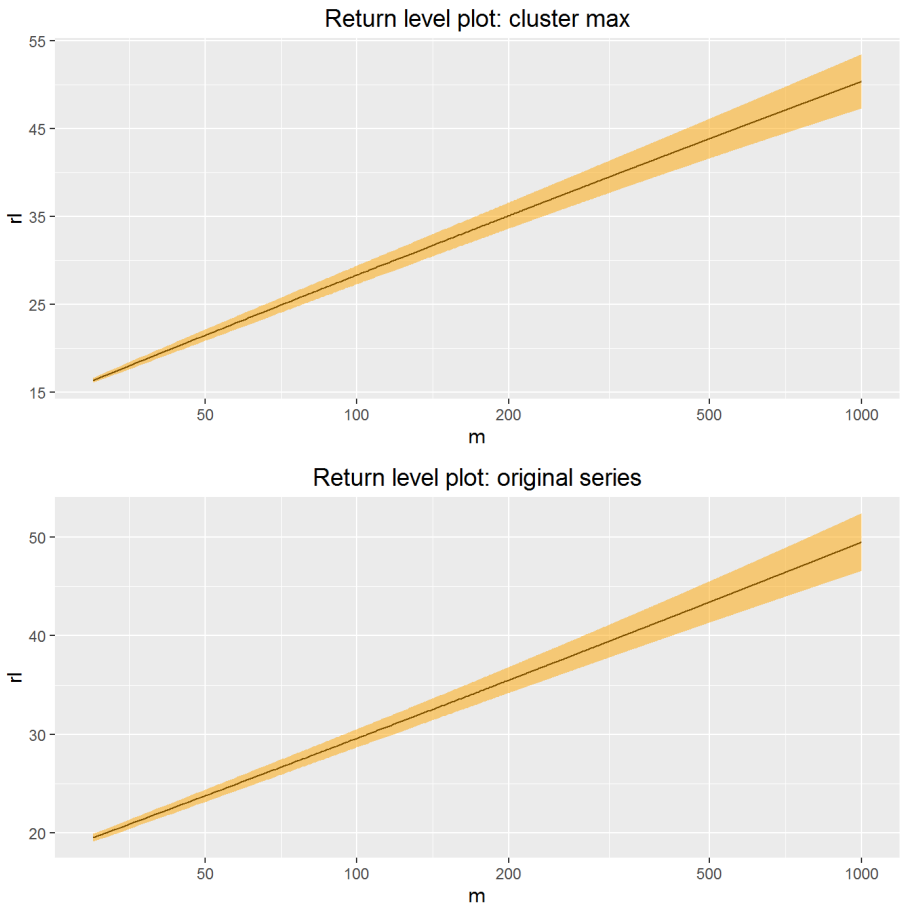
which returns an object of class `evmSim`. See documentation for the `evm` function for further information.

Estimation of return levels

Return levels can be computed from the GPD fitted to cluster maxima in the usual way. Note that the return levels computed from the declustered data refer to the occurrance of cluster maxima, rather than all threshold excesses and need to be interpreted accordingly.

```
M <- seq(30,1000)
p1 <- ggplot(predict(rain.gpd,M=M,ci.fit=TRUE),
             main="Return level plot: cluster max")
p2 <- ggplot(predict(evm(rain,th=13),M=M,ci.fit=TRUE),
             main="Return level plot: original series")
breaks <- c(20,50,100,200,500,1000)
grid.arrange(p1[[1]] + scale_x_continuous(trans="log",breaks=breaks),
             p2[[1]] + scale_x_continuous(trans="log",breaks=breaks),
```

```
                ncol=1)
```



The return levels calculated for the cluster maxima and the original series do not differ enormously. Close to the threhsold, excesses of a given level are rarer under the fitted cluster-max model than under the fitted original series model. This is is explained by the trivial observation that cluster maxima occur at a lower rate than do threshold exceedances. Away from the fitting threshold, the two curves give very similar estimates of return levels. This reflects the relatively weak dependence in the original series so that clusters are of short duration and the declustered series is not so different from the original one.

Discussion

The procedure described above, of declustering and then fitting the GPD to cluster maxima gives a valid statistical model whose underlying assumptions are met. However, it is interesting to note that in practice, the cluster maxima may not be of ultimate interest. For example, rainfall information can be helpful

if the assessment of flood damage is the ultimate goal. Here it may be more informative to analyse complete clusters and obtain an understanding of the aggregate rainfall over a rainy spell, rather than to focus on the largest daily value over that spell. This problem is inherently more difficult and requires a much more sophisticated solution; it is not attempted here.

A further point that deserves to be made concerns the choice of interexceedance time cut-off which divides those times into intra-cluster (short times) and inter-cluster times (long times). The above algorithm offers an automatic procedure for deciding the cut-off time, based on a statistical model whose assumptions can be examined. However in applications where there is a valid time horizon within which threshold exceedances should be considered to arise within the same cluster, then this statistical model based approach should not over-ride the science based argument in the case where the methods suggest different answers.

In cases where there is a clear *a priori* argument for selecting a cluster separation time, then the declustering can be carried out by using the so-called *runs declustering* algorithm (see Coles, 2001 [1], Chapter 5). Here we must specify the *run-length* `r`, the minimum number of consecutive values which must lie below the threshold before a cluster is deemed to be complete. In `texmex` we do:

```
declust(rain,th=13,r=5)

## declust.extremalIndex(y = ei, r = r)
##
## Threshold  13
## Declustering using the runs method, run length 5
## Identified 696 clusters.
```

Specifying a run length `r` forces the use of the *runs* declustering method. In contrast, when we fail to specify this then the intervals method automaticically fixes the run length following estimation of the extremal index:

```
declust(rain,th=13)

##
## Length of original series 17531
## Threshold 13
## Number of Threshold Exceedances 1294
## Intervals estimator of Extremal Index 0.7046188
## declust.extremalIndex(y = ei, r = r)
##
## Threshold  13
## Declustering using the intervals method, run length 3
## Identified 817 clusters.
```

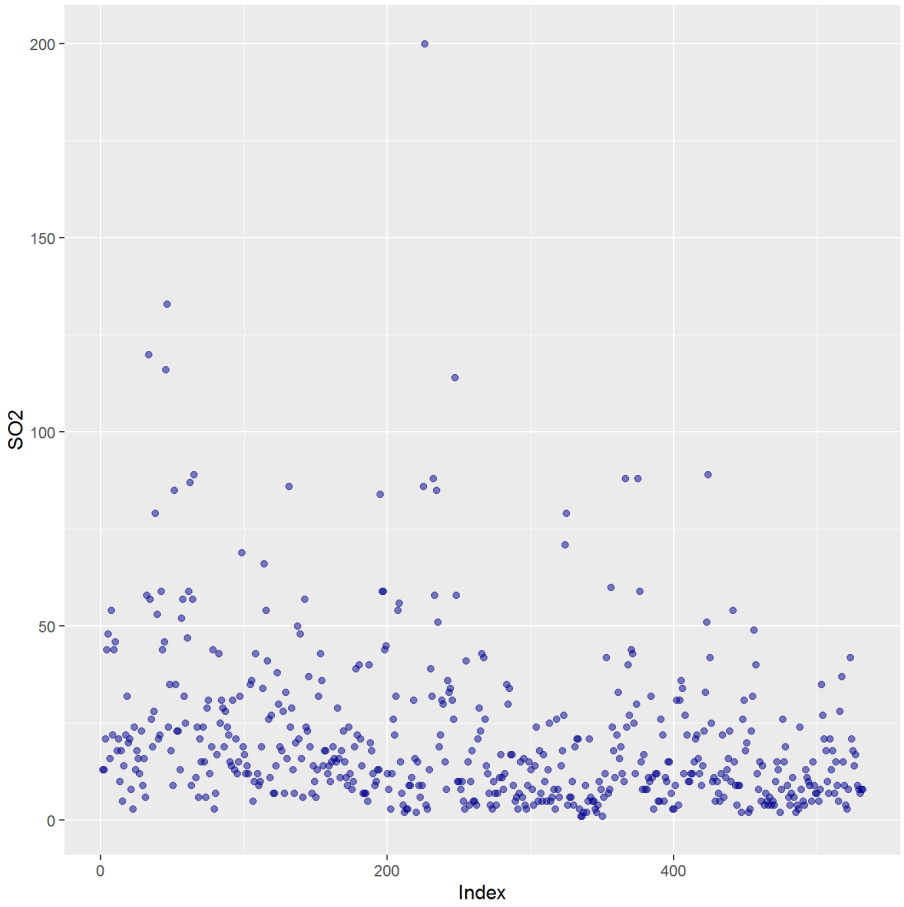
2.8 Modelling dependent series with covariates – air pollution data

We now look at the possibility of including covariates in our GPD model for cluster maxima. This is demonstrated by using the five-dimensional *winter air pollution* dataset included in `texmex`. For more information on this dataset type

```
help(winter)
```

We look at the sulphur dioxide component of this series which exhibits temporal dependence, and following declustering of this series attempt to model the excesses of a threshold by cluster maxima by using the remaining variables in the dataset as explanatory variables. The serial dependence in the SO2 data is apparent from a plot of the data in which the largest values appear clumped together:

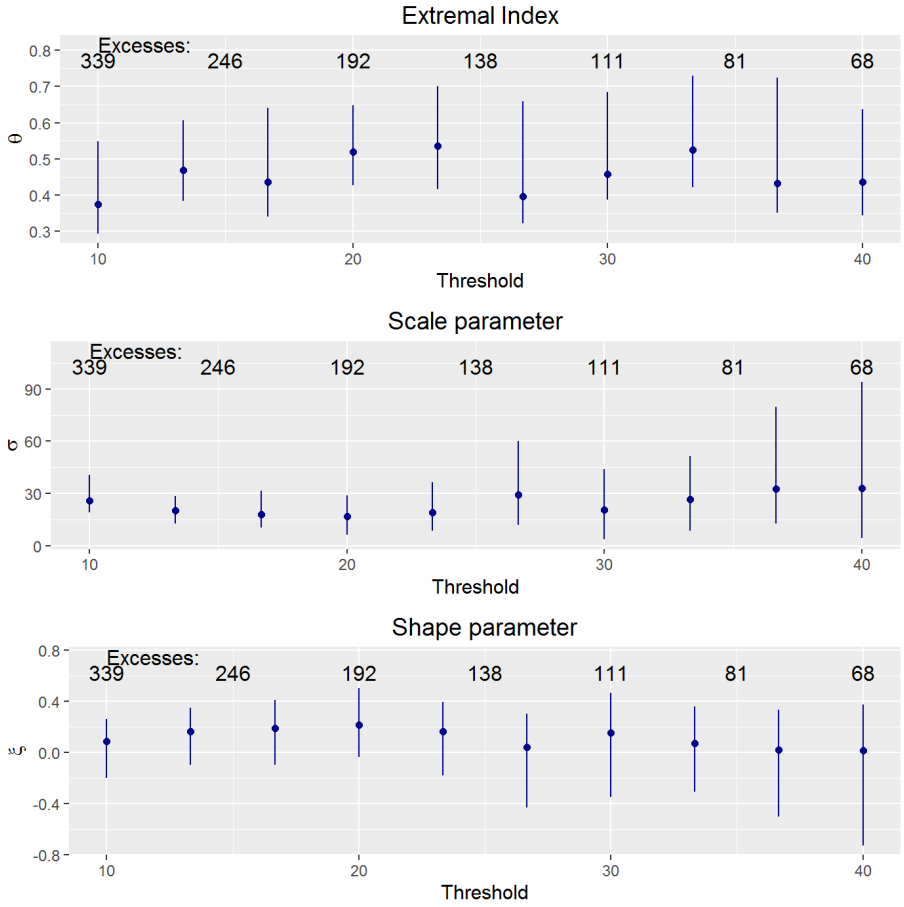
```
ggplot(data=data.frame(Index = 1:length(winter$SO2), SO2 = winter$SO2),
       aes(Index,SO2)) + geom_point(colour="dark blue",alpha=0.5)
```



Declustering

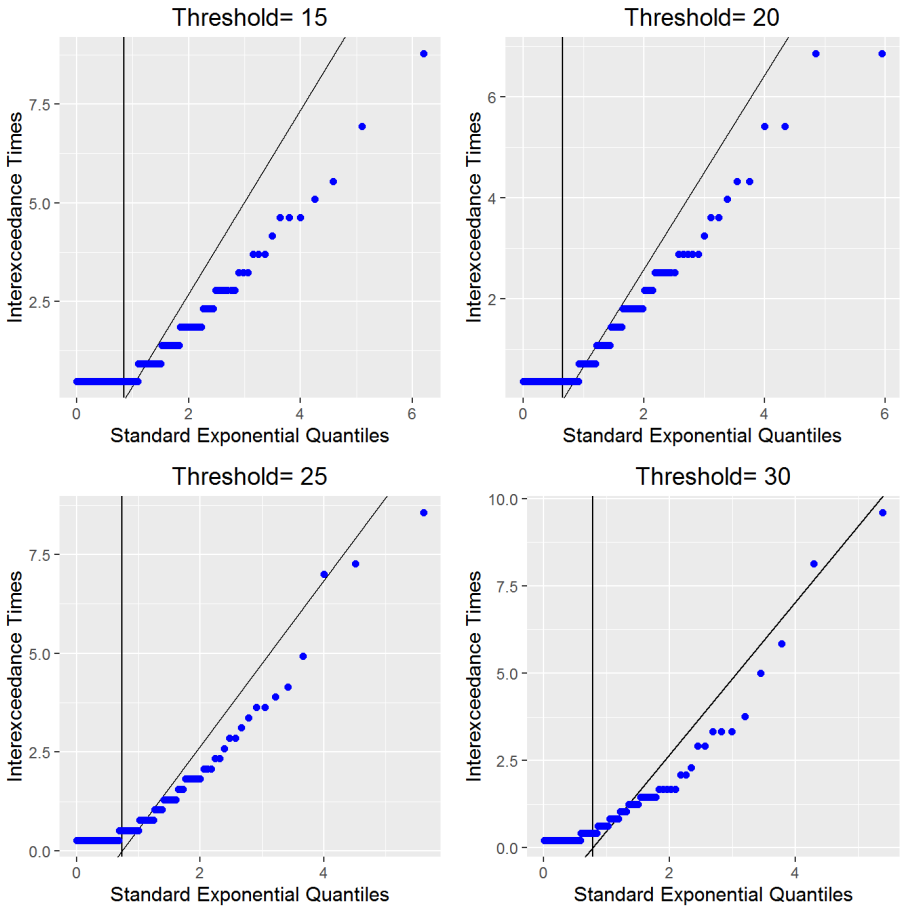
The SO2 variable is declustered whilst retaining the covariate structure of the data frame which holds the data. First we select a threshold:

```
erf <- extremalIndexRangeFit(SO2,data=winter,
                             umin=10,umax=40,verb=FALSE)
g <- ggplot(erf)
grid.arrange(g[[1]],g[[2]],g[[3]],ncol=1)
```



This suggests that we should not use a threshold of below 15. We further examine threshold choice by using Q-Q plots:

```
g1 <- ggplot(extremalIndex(SO2,data=winter,threshold=15))
g2 <- ggplot(extremalIndex(SO2,data=winter,threshold=20))
g3 <- ggplot(extremalIndex(SO2,data=winter,threshold=25))
g4 <- ggplot(extremalIndex(SO2,data=winter,threshold=30))
grid.arrange(g1,g2,g3,g4,ncol=2)
```

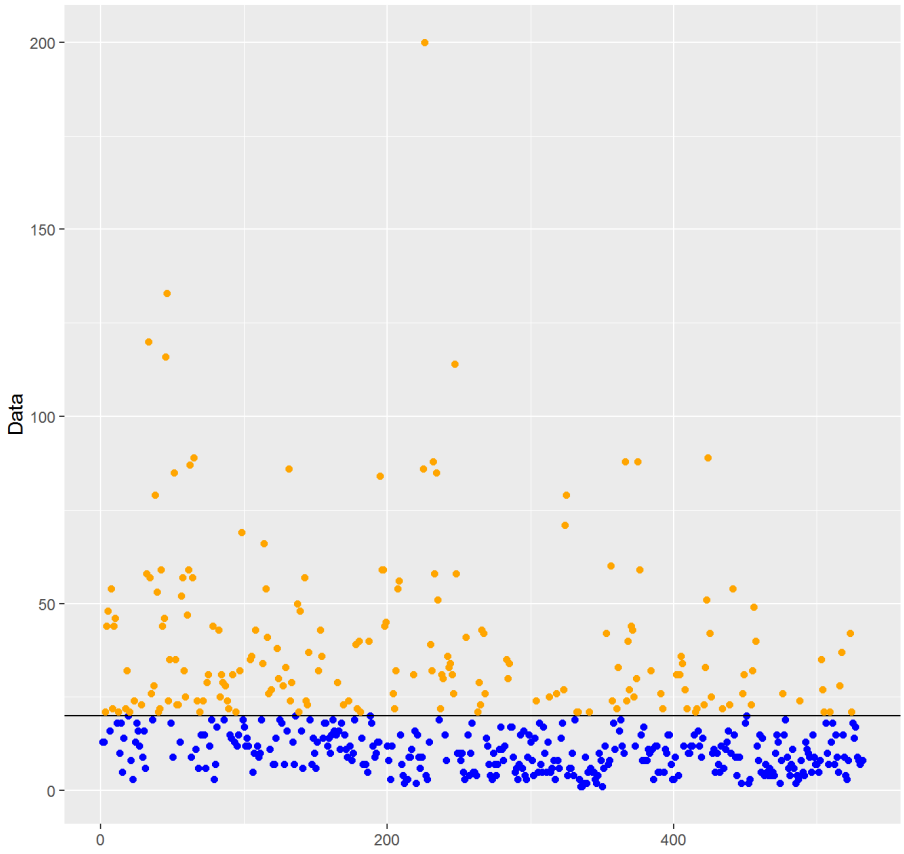


This suggests that a threshold of around 20 gives the best fit to the data. We proceed by using this value.

```
ei <- extremalIndex(SO2,data=winter,threshold=20)
ei

##
## Length of original series 532
## Threshold 20
## Number of Threshold Exceedances 192
## Intervals estimator of Extremal Index 0.5206914
```

The extremal index estimate of around 0.5 has the interpretation of cluters occuring with an average size of two. We can decluster using this estimate of the extremal index:

```
d <- declust(ei)
ggplot(d)
```




```
d

## declust.extremalIndex(y = ei)
##
## Threshold  20
## Declustering using the intervals method, run length 1
## Identified 77 clusters.
```

So of the 192 threshold exceedances, only 77 are independent cluster maxima.

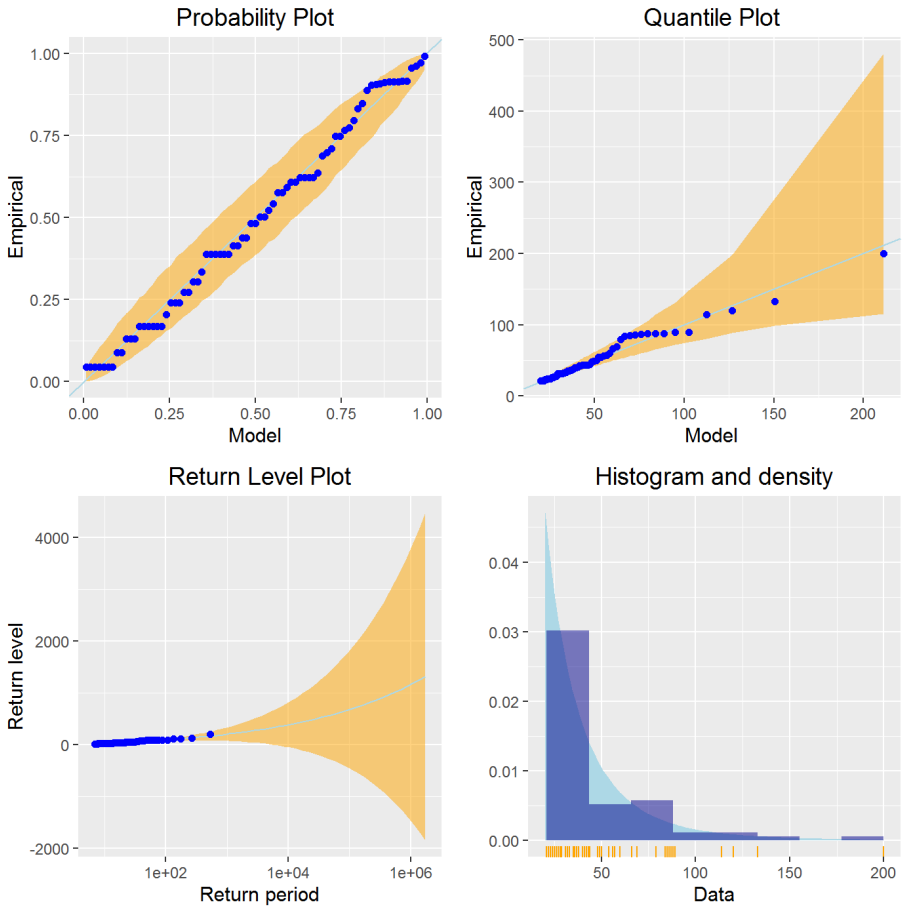
Fitting the GPD to the cluster maxima

We can now fit the GPD model to the excesses of cluster maxima above our threshold of 20. There are no covariates in the GPD model at this stage.

```r
so2.gpd <- evm(d)
so2.gpd
```

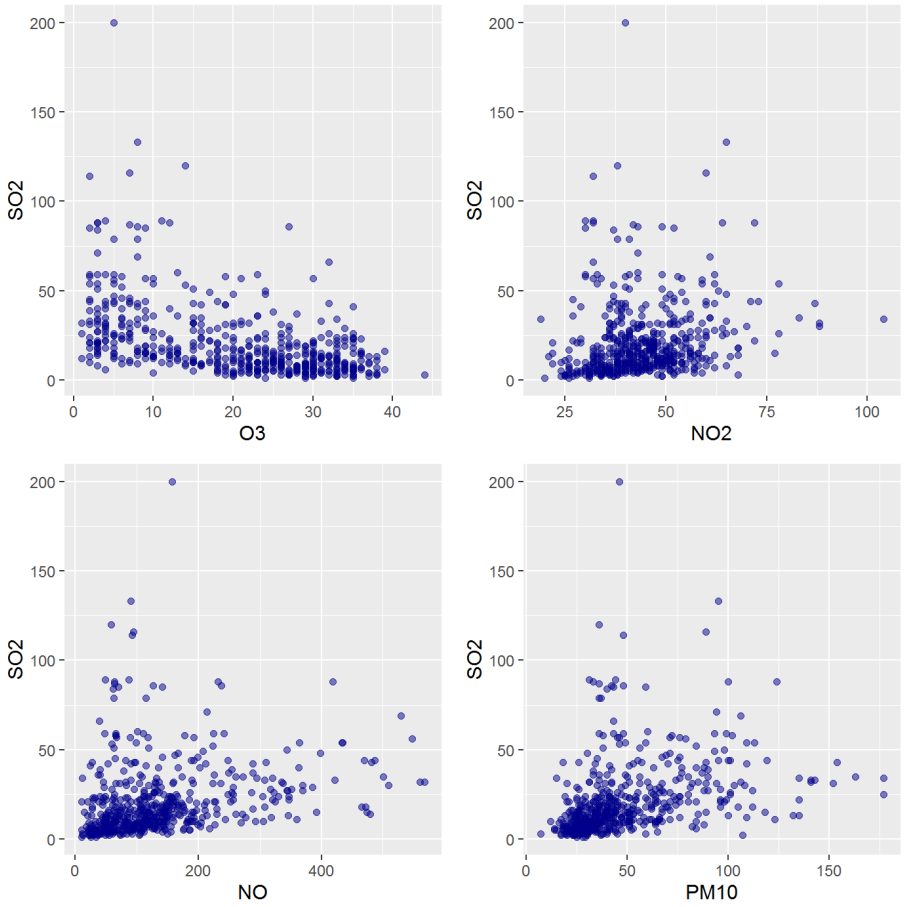
```
## Call: evm.declustered(y = d)
## Family:       GPD
##
## Model fit by maximum likelihood.
##
## Convergence: TRUE
## Threshold: 20
## Rate of excess: 0.1447
##
##   Log. lik   AIC
##   -328.5506  661.1012
##
##
## Coefficients:
##                   Value   SE
## phi: (Intercept)  3.0552  0.1935
## xi: (Intercept)   0.2121  0.1583
```

```r
ggplot(so2.gpd)
```



The diagnostic plots show a small kink in the Q-Q plot. We now look into whether we can improve upon this slight lack of fit by covariate modelling. We can examine the suitability of the remaining air quality variables as explanatory variables in linear predictors for the GPD model parameters by using scatter plots:

```
p1 <- ggplot(data=winter,aes(O3,SO2)) + geom_point(colour="dark blue",alpha=0.5)
p2 <- ggplot(data=winter,aes(NO2,SO2)) + geom_point(colour="dark blue",alpha=0.5)
p3 <- ggplot(data=winter,aes(NO,SO2)) + geom_point(colour="dark blue",alpha=0.5)
p4 <- ggplot(data=winter,aes(PM10,SO2)) + geom_point(colour="dark blue",alpha=0.5)
grid.arrange(p1,p2,p3,p4,ncol=2)
```



Here we are not looking for linear regression type relationship but instead whether the scatter (scale) or tail behaviour (shape) of the SO2 variable depend on the candidate explanatory variable. The Ozone variabe looks like a possible candidate since the largest values of `SO2` are much more scattered for small values of Ozone than for large. We will look at the fitted models for each of the candidate explanatory variables, and examine the AIC (Akaike Information Criterion) for each in turn. A model that gives a better fit to the data – beyond that which we would expect to see simply due to the addition of more model parameters – will have a reduced values of AIC. We favour models with lower AIC. The absolute value of AIC is not of interest in itself as this is a function of the exact choice of data used to fit the model. The values of AIC for models fit to different sets of data (for instance to threshold excesses defined by different choices of threshold) are not comparable.

Covariate models are fitted to the declustered data object by using the *model formula* syntax, specifying the name of the parameter to be modelled with a covariate and also the name of the column of the original data frame containing the covariate. For example, to include the covariate `NO` in the linear predictor

for the log scale parameter, `phi`, we do:

```
evm(d,phi=~NO)
```

```
## Call: evm.declustered(y = d, phi = ~NO)
## Family:       GPD
##
## Model fit by maximum likelihood.
##
## Convergence: TRUE
## Threshold: 20
## Rate of excess: 0.1447
##
##   Log. lik   AIC
##   -328.4744  662.9488
##
##
## Coefficients:
##                    Value      SE
## phi: (Intercept)   3.138860   0.283911
## phi: NO           -0.000399   0.001013
## xi: (Intercept)    0.197205   0.160885
```

The model is estimated by using the *treatment contrasts* parameterisation, so that we estimate an *intercept* for the parameter `phi` (the value taken by this parameter for a zero value of the covariate) and also a *gradient*. Here the estimated gradient shows a very slight decline in `phi` for every positive unit change in `NO` (although this gradient parameter is not significantly different from zero).

We can examine the AIC for each of the candidate explanatory variables as follows:

```
AIC(evm(d,phi=~NO))
```

```
## [1] 662.9488
```

```
AIC(evm(d,phi=~NO2))
```

```
## [1] 662.8852
```

```
AIC(evm(d,phi=~O3))
```

```
## [1] 651.8719
```

```
AIC(evm(d,phi=~PM10))
```

```
## [1] 662.9997
```

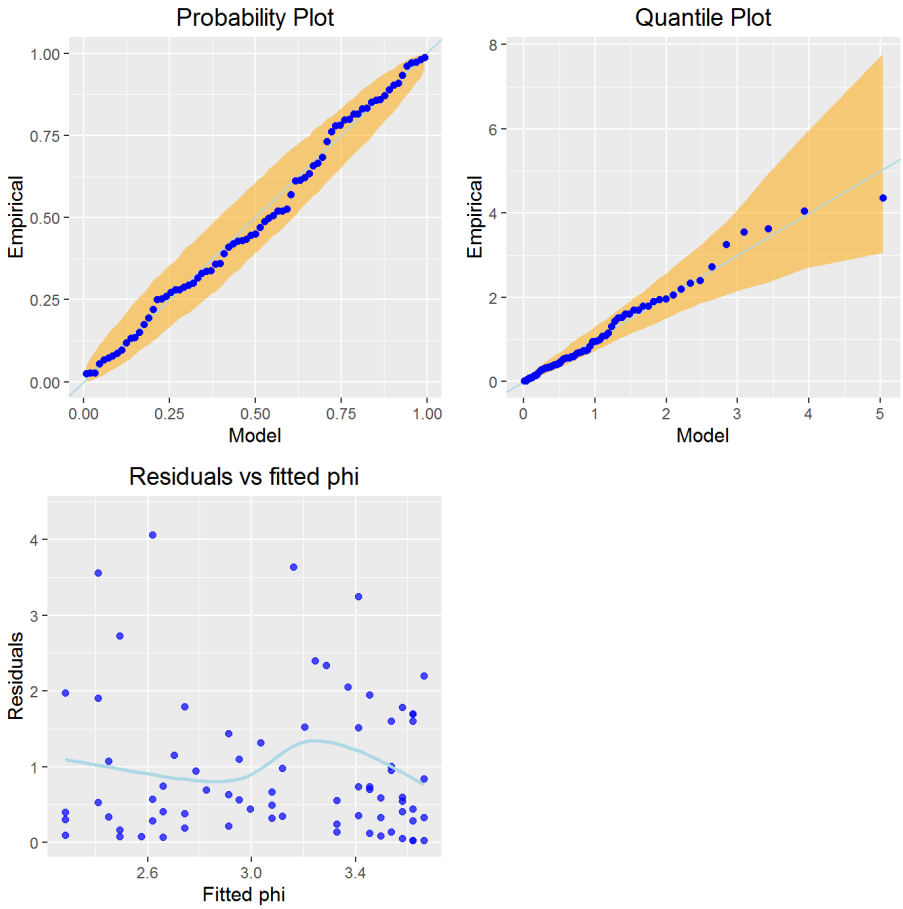
As suggested by the scatter plots, the Ozone variable has the best model fit, with by far the lowest AIC value, and a significant improvement over the model fitted with no covariates. We now examine whether we can obtain any further improvement in fit by including an additional covariate in the linear predictor for the shape parameter, `xi`.

```
AIC(evm(d,phi=~O3,xi=~NO))

## [1] 651.7845

AIC(evm(d,phi=~O3,xi=~NO2))

## [1] 653.2484

AIC(evm(d,phi=~O3,xi=~O3))

## [1] 653.6356

AIC(evm(d,phi=~O3,xi=~PM10))

## [1] 652.921
```

So we do not gain further by including any of these variables in an expression for `xi`. We reach the same conclusion (working not shown here) if we examine the inclusion of covariates only in the linear predictor for `xi`, and not in the (log) scale parameter.

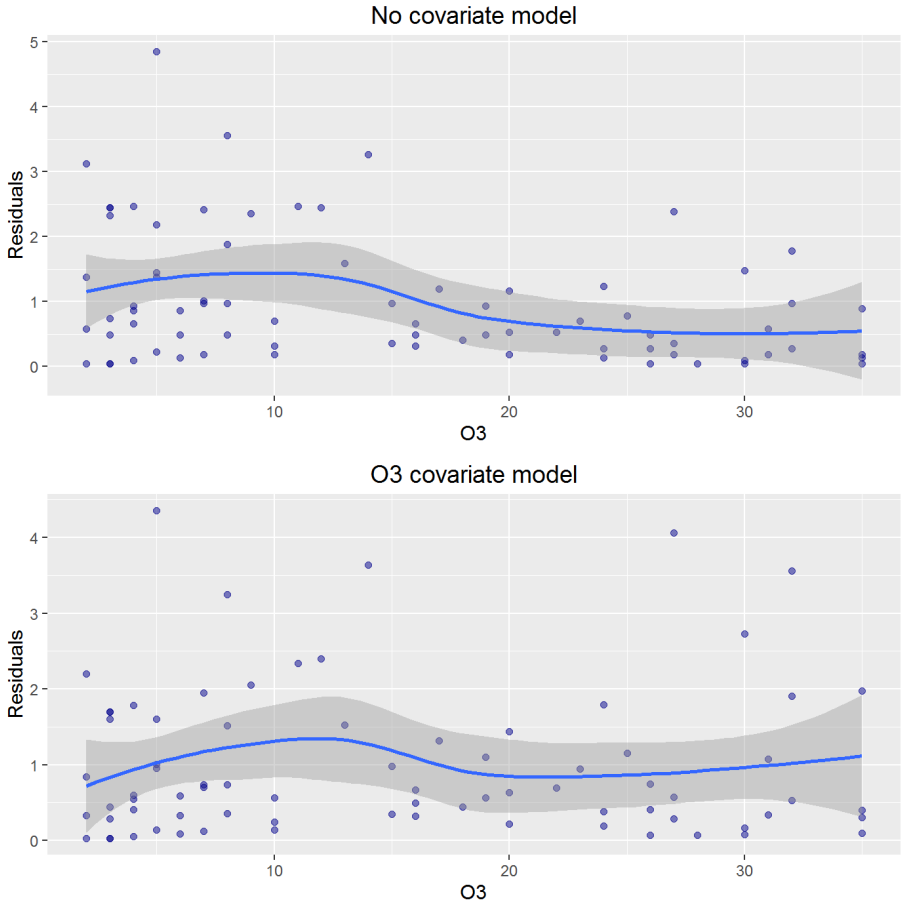
Finally we look at the model diagnostics for our preferred covariate model:

```
so2.o3.gpd <- evm(d,phi=~O3)
ggplot(so2.o3.gpd)
```



So we have managed to get rid of the previous kink in the Q-Q plot (Page 23) which appeared for the model when fitted with no covariates. We can also plot the residuals from each of our models against the O3 covariate:

```
O3 <- winter[winter$SO2>d$threshold,"O3"][d$isClusterMax]

p1 <- ggplot(data=data.frame(O3=O3, Residuals = resid(so2.gpd)),
             aes(O3,Residuals)) +
    geom_point(colour="dark blue",alpha=0.5) +
    labs(title="No covariate model") + geom_smooth()

p2 <- ggplot(data=data.frame(O3=O3, Residuals = resid(so2.o3.gpd)),
             aes(O3,Residuals)) +
    geom_point(colour="dark blue",alpha=0.5) +
    labs(title="O3 covariate model") + geom_smooth()
grid.arrange(p1,p2,ncol=1)
```



These plots show the covariate model captures the dependence of the no covariate model residual scatter on the Ozone variable.

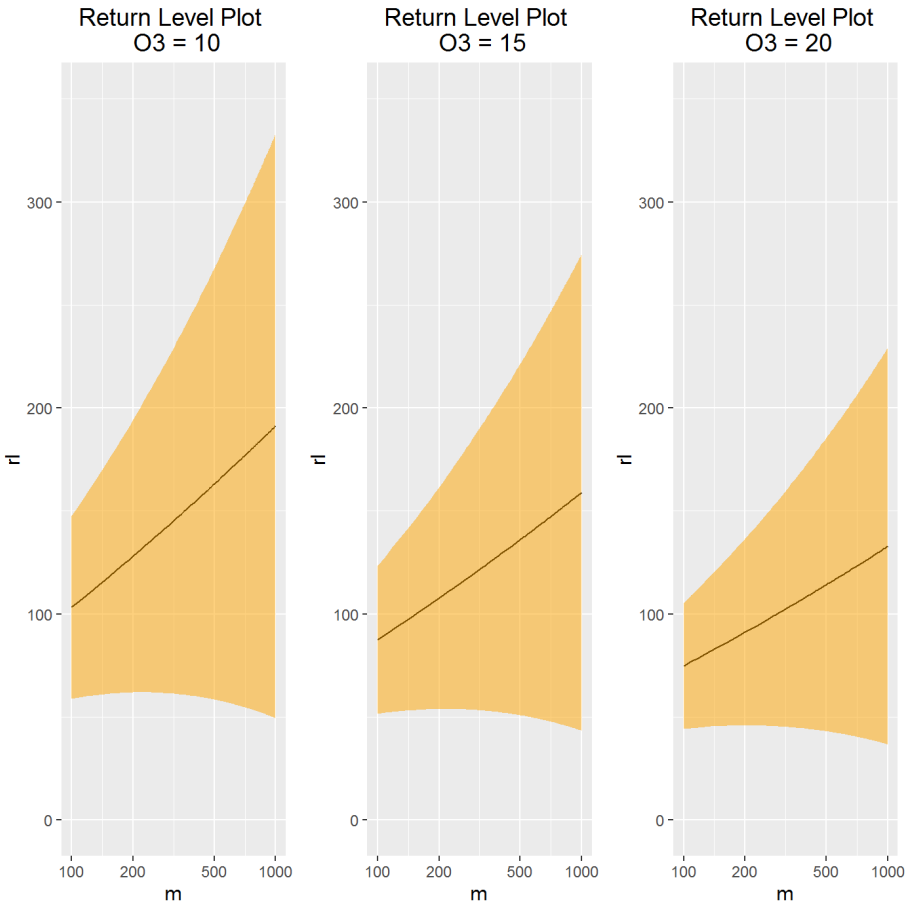
Return level estimation

Now that our preferred model for SO2 has model parameters that are specified as a function of a covariate – `O3` – we now have to specify a value of O3 to calculate a return level for SO2.

For instance, to look at `O3` values of 10, 15 and 20 and predicted return levels for each of these environmental conditions, we can do:

```
newdata <- data.frame(O3=c(10,15,20))
so2.o3.rl <- predict(so2.o3.gpd,M=seq(100,1000,len=100),ci.fit=TRUE,
                     newdata=newdata)
g <- ggplot(so2.o3.rl)
breaks <- c(100,200,500,1000)
grid.arrange(g[[1]] + scale_x_continuous(trans="log",breaks=breaks) +
```

```
    scale_y_continuous(limits=c(0,350)),
g[[2]] + scale_x_continuous(trans="log",breaks=breaks) +
    scale_y_continuous(limits=c(0,350)),
g[[3]] + scale_x_continuous(trans="log",breaks=breaks) +
    scale_y_continuous(limits=c(0,350)),ncol=3)
```



The unit of time to which the *return period* now refers is the *cluster*, since the GPD model has been fitted to cluster maxima.

Discussion

The interpretation of GPD models fitted with covariates included in the linear predictors for model parameters requires some delicacy. We put this in context by recalling the general approach taken within the statistical extreme value modelling paradigm. All extreme value statistical models are motivated by the limiting behaviour of random variables as we look at more and more extreme values of these variables. These limiting arguments provide statisticians with

some theoretical justification for using these models in regions of the sample space where data are scarce and unreliable for model validation; this is particularly helpful when we are required to extrapolate from our models, beyond levels that have been seen in the data for model fitting. We exploit stability properties of these models that arise as a consequence of the limiting arguments from which they are born.

These useful model properties and the interpretation of model parameters are somewhat muddied when we use covariates in the linear predictors for the parameters of the extreme value models. As for any regression type model, the models are conditional on the observed values of the covariate. This is satisfactory when the covariate can be fixed by design, and it makes sense to think of holding its value constant while we extrapolate the response variable. In the context of our winter air pollution example however, the Ozone variable was not set at a design value, but observed jointly with the other air quality variables. It is not at all clear from this setting whether it is appropriate to think about holding the value of Ozone fixed while examining ever higher values of SO2.

The issue of threshold selection also requires further consideration in the context of covariate modelling. For our simple example above, we used a constant threshold for the SO2 declustering and subsequent modelling of cluster maxima. The validity of this modelling practice for non-stationary processes is rather questionable. Our approach above asserts that the probablity of threshold excess is not affected by the covariate value. This is a very strong assumption and has been called into doubt in the literature for applications of this type. A possible solution is to use a variable threhsold, possibly also depending on the covariate. Further consideration of this issue is given in [4] and references therein. The current implementation of `texmex` does not support such sophisticated modelling.

References

[1] S. Coles. *An Introduction to Statistical Modelling of Extreme Values*. Springer, 2001.

[2] C.A.T. Ferro and J. Segers. Inferences for clusters of extreme values. *JRSS B*, 65:545–556, 2003.

[3] M.R. Leadbetter. Extremes and local dependence in stationary sequences. *Z. Wahrsch. Ver. Geb.*, 65:291–306, 1983.

[4] P.J. Northrop and P. Jonathan. Threshold modelling of spatially dependent non-stationary extremes with application to hurricane-induced wave heights. *Environmetrics*, 22:799–809, 2011.

[5] R Development Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2011. ISBN 3-900051-07-0.

[6] H. Southworth and J. E. Heffernan. *texmex: Threshold exceedences and multivariate extremes*, 2016. R package version 2.3.