Univariate Extreme Value Modelling using R

Harry Southworth and Janet E. Heffernan

September 28, 2016

1 Introduction

This document illustrates the use of the `texmex` package for performing extreme value analysis of some environmental data in `R`, [4]. This package vignette focusses on univariate extreme value modelling of threshold excesses using the generalized Pareto distribution (GPD), and of data arising as maxima, using the generalized extreme value (GEV) distribuion. The separate vignette `texmexMultivariate` examines multivariate extreme value modelling using a conditional threshold based approach. For extreme value modelling of temporally dependent Peaks over Threshold data by using declustering, see the package vignette `declustering`.

To cite this vignette, refer to Vignette name: `texmex1d` 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},
##   }
```

Extreme value statistical models are unusual among statistical models in that they are often required for extrapolation beyond levels observed in the data. As statisticians, we are told that extrapolation from statistical models is perilous: our models can only be trusted in regions where we have sufficient data to calibrate and check goodness of model fit. Extreme value modelling

has responded to a demand for extrapolation beyond this safe region. Since we can no longer rely on data as a check on our models' suitability, extreme value statisticians turn to mathematical arguments to bolster their confidence in their extrapolation. These arguments provide a justification for the use of a particular type of model to describe tail behaviour of random variables.

This is not a tutorial in Extreme Value Theory, for which we refer the reader to [1], which describes a range of methods for modelling the statistical properties of sample maxima, threshold excesses, extremes of dependent series and other aspects of tail behaviour.

1.1 Preliminaries

With `texmex` installed, use the `library` command to make the package available to the current session, set the colours used for graphics, and set the random seed so that results are reproducible on a given machine:

```
library(texmex)
library(gridExtra)
palette(c("black","purple","cyan","orange"))
set.seed(20130618)
```

The `gridExtra` package is used for laying out plots produced by `ggplot2`.

1.2 Data

The datasets used in this example analysis are contained in the `texmex` package. We give a detailed exposition of the fitting of the GPD without covariates to a daily rainfall dataset, `rain`, which appears in Coles (2001), [1]. We show how to extend the modelling framework to include covariates using the `winter` air pollution data from Heffernan and Tawn (2004), [3]. The modelling approach for fitting GEV models that we take within `texmex` is very similar to that for fitting the GPD, so we conclude with a brief demonstration of this by using the annual maxima sea-level dataset, `portpirie`, again from Coles [1]. More details of these datasets are given in their help files.

1.2.1 Rainfall data

```
head(rain)
```

```
## [1] 0.0 2.3 1.3 6.9 4.6 0.0
```

```
summary(rain)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   0.000   0.000   0.500   3.476   4.300  86.600
```

```
length(rain)
```

```
## [1] 17531
```

To plot the data (not shown here):

```
d1 <- ggplot(data=data.frame(rain=rain,index=1:length(rain)),
             aes(index,rain)) +
    geom_point(alpha=0.5,col=4)
d1
```

1.2.2 Winter air pollution data

```
head(winter)
```

```
##   O3 NO2  NO SO2 PM10
## 1 27  50 112  13   34
## 2 27  51 126  13   29
## 3 15  43  90  21   33
## 4  9  71 470  44  101
## 5 20  51 167  48   30
## 6  8  50 211  16   44
```

```
summary(winter,digits=3)
```

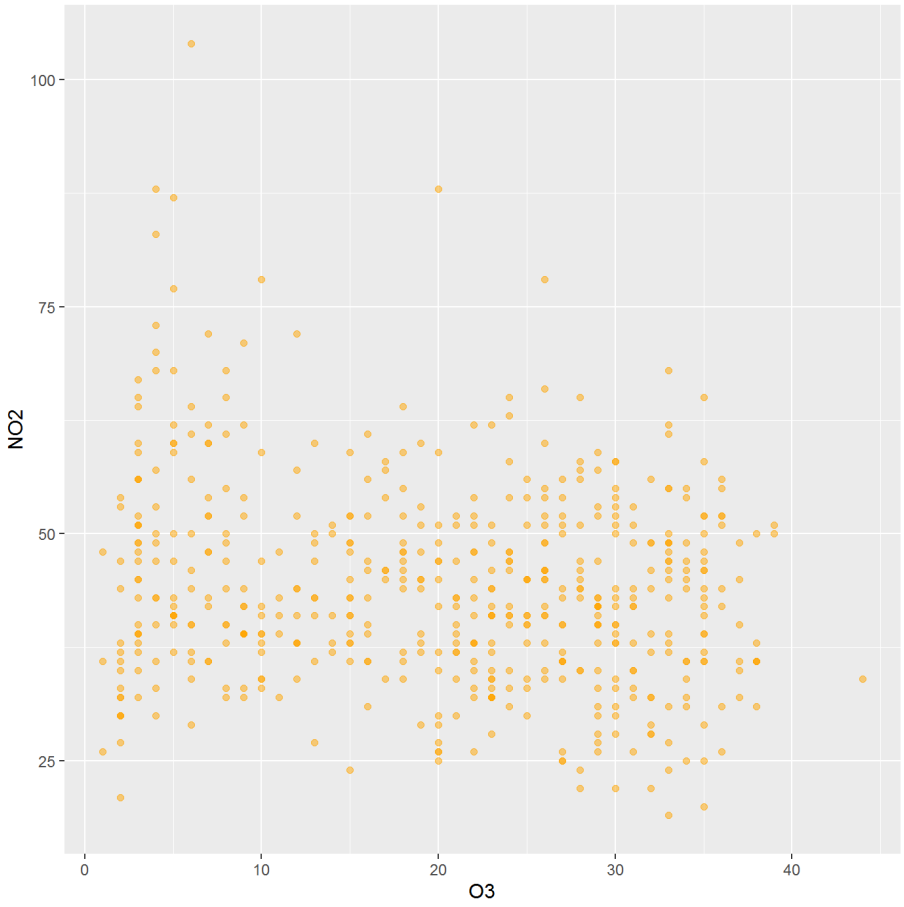
```
##        O3             NO2              NO            SO2
##  Min.   : 1.0   Min.   : 19.0   Min.   : 10   Min.   :  1
##  1st Qu.:10.0   1st Qu.: 36.8   1st Qu.: 64   1st Qu.:  8
##  Median :22.0   Median : 43.0   Median :112   Median : 15
##  Mean   :20.1   Mean   : 44.2   Mean   :135   Mean   : 21
##  3rd Qu.:29.0   3rd Qu.: 51.0   3rd Qu.:166   3rd Qu.: 26
##  Max.   :44.0   Max.   :104.0   Max.   :568   Max.   :200
##       PM10
##  Min.   :  7.0
##  1st Qu.: 29.0
##  Median : 40.0
##  Mean   : 48.4
##  3rd Qu.: 60.0
##  Max.   :177.0
```

```
dim(winter)
```

```
## [1] 532   5
```

We focus on the two variables Nitrogen Dioxide, `NO2` and ozone `O3` in the examples of covariate modelling that follow. A scatter plot of these two variables suggests a negative association:

```
d2 <- ggplot(winter,aes(O3,NO2)) + geom_point(alpha=0.5,col=4)
d2
```



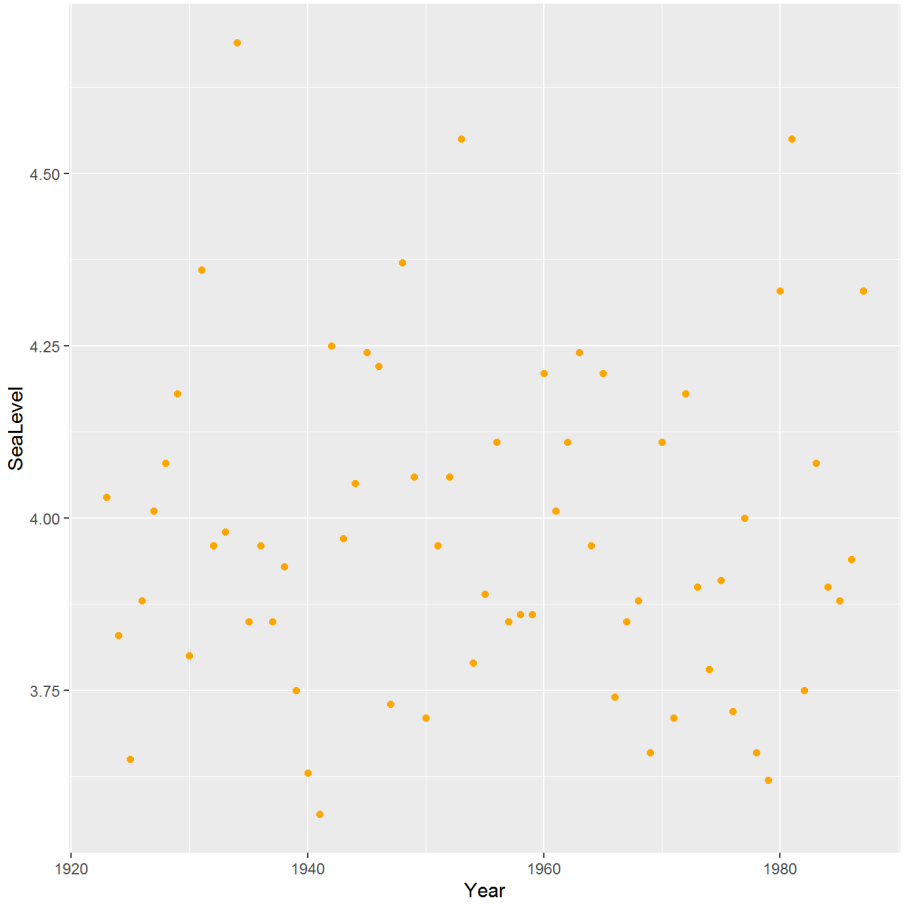
1.2.3 Portpirie data

```
head(portpirie)
```

```
##   Year SeaLevel
## 1 1923     4.03
## 2 1924     3.83
## 3 1925     3.65
## 4 1926     3.88
## 5 1927     4.01
## 6 1928     4.08
```

```
summary(portpirie)
```

```
##       Year         SeaLevel
##  Min.   :1923   Min.   :3.570
##  1st Qu.:1939   1st Qu.:3.830
##  Median :1955   Median :3.960
##  Mean   :1955   Mean   :3.981
##  3rd Qu.:1971   3rd Qu.:4.110
##  Max.   :1987   Max.   :4.690

dim(portpirie)

## [1] 65  2

d3 <- ggplot(portpirie,aes(Year,SeaLevel)) + geom_point(col=4)
d3
```



2 Generalized Pareto distribution models

We now proceed to fit, evaluate, choose between, and ultimately make predictions from generalized Pareto distribution (GPD) models.

2.1 Extreme value modelling and asymptotic motivation for the GPD

In this section, we show how to fit the generalised Pareto Distribution, GPD(σ, ξ) [2] to data points in excess of suitably chosen thresholds. The GPD has distribution function

$$F_{>u}(x) = 1 - \left\{1 + \xi\left(\frac{x-u}{\sigma}\right)\right\}^{-1/\xi} \quad \text{for } x > u, \tag{1}$$

where u is the threshold for fitting and $\sigma > 0$ and $\xi \in \mathbb{R}$ are the scale and shape parameters respectively. This is the conditional distributon of observations given that the observations exceed the fitting threshold u. The range of possible values taken by realisations from the GPD depends on the parameter values, with the distribution having a finite upper end point (short tailed) if the shape parameter is negative ($u < x \leq u - \sigma/\xi$ if $\xi < 0$) and an infinite tail otherwise ($u < x < \infty$ if $\xi \geq 0$). When $\xi = 0$, the GPD corresponds exactly to the Exponential distribution.

Extreme value theory tells us that under appropriate normalisation of the threshold excesses, as the threshold u tends to the distributional upper endpoint, the limiting distribution of the excesses must fall in the generalised Pareto family of distributions (given certain conditions concerning non-degeneracy of the limit distribution and smoothness of the distribution of the original variable). So whatever the original distribution of the measurements, provided we choose an appropriately high threshold, the distribution of values exceeding that threshold should be well approximated by a GPD. Diagnostic tools to aid the choice of suitable threshold are standard, and are described shortly – see also [1].

2.2 Parameterization

The usual parameterization of the GPD (as in Equation (1)) is in terms of its scale paramter σ and shape parameter ξ. There are, however, good reasons for reparameterizing in terms of $\phi = \log \sigma$:

- Experience has demonstrated that the numerical algorithms used for optimizing the log-likelihood tend to converge more reliably when working with ϕ;
- When including covariates in the model we are faced with the constraint that $\sigma > 0$ and working with a linear predictor specified in terms of $\phi = \log \sigma$ guarantees this constraint;

- When placing prior distributions on parameters, it is convenient to work with Gaussian distributions and ϕ is more likely to be close to Gaussian than is σ.

As such, some of the functions in `texmex` work with ϕ, not σ. In the case when inference is required for σ rather than ϕ, the point estimates can simply be exponentiated if maximum likelihood estimation is used. If a prior distribution is used, the point estimates are not invariant to transformation, so any transformed values should only be considered to be approximate.

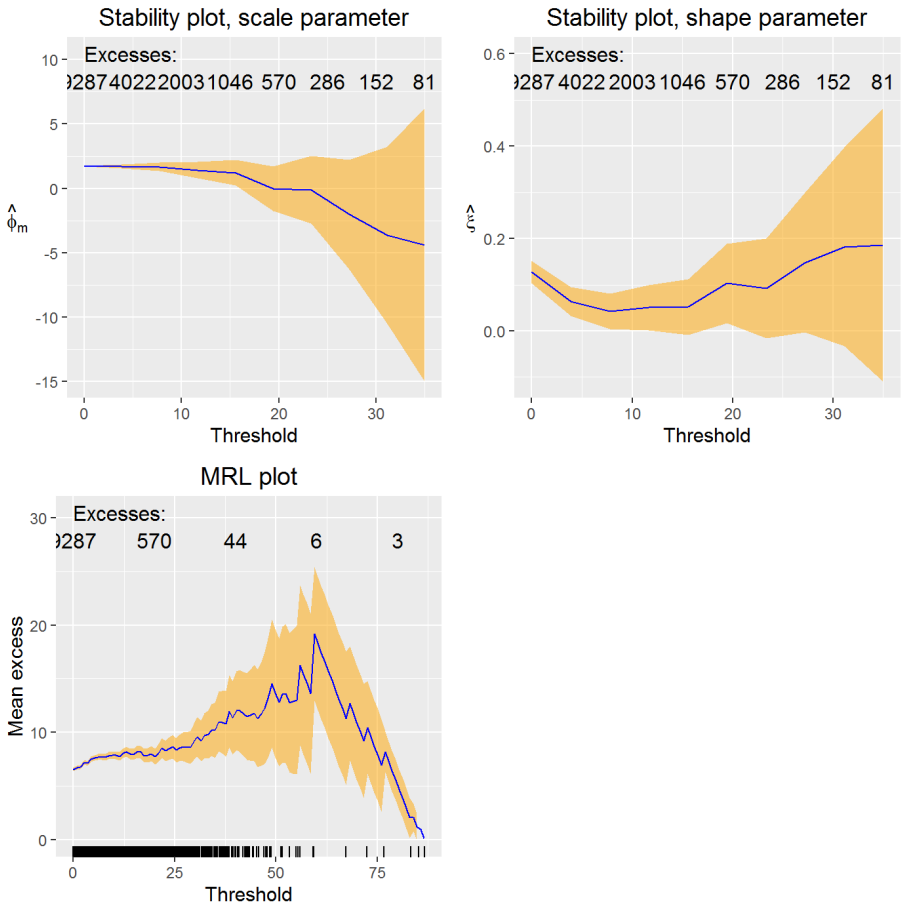
2.3 Threshold selection

GPD modelling proceeds by selecting a threshold above which the data appear to be well modelled. Standard tools for threshold selection that appear in the literature (see for example [1]) include the *mean residual life* (MRL) plot, and plots of parameters estimated using a range of thresholds, *threshold stability plots*.

For a suitably chosen threshold, the mean residual life plot should be linear and the parameter estimates in threshold stability plots constant above the chosen threshold (both of these requirements are assessed by taking account of sampling variability). The sign of the gradient in the linear part of the MRL plot corresponds to the sign of the shape parameter and hence indicates the shape of the tail – negative slope shows a short tailed distribution, a horizontal line (zero gradient) shows an exponential type tail and a positive slope suggests a heavy tailed distribution.

We illustrate the use of these diagnostics now for the `rain` data:

```
grfRain <- gpdRangeFit(rain,umax=35)
mrlRain <- mrl(rain)

g1 <- ggplot(grfRain)
g2 <- ggplot(mrlRain)

grid.arrange(g1[[1]] + ggtitle("Stability plot, scale parameter"),
             g1[[2]] + ggtitle("Stability plot, shape parameter"),
             g2 + ggtitle("MRL plot"),ncol=2)
```



The threshold stability plots (top) show both (log-)scale and shape parameter estimates to be stable for thresholds of around 20 and above.

The Mean Residual Life plot (bottom) has a linear form from values of around ten, the gradient being positive (up to a threshold of around 60 above which there are only a small handful of points). This indicates a heavy tail and positive shape parameter.

Note that this form of MRL plot is typical, with the very highest thresholds giving very erratic estimates with apparently narrow confidence bands. This commonly observed feature is due to the estimates for very high thresholds being based on a very small number of points - the very largest points in the data set. These are by construction close to the sample maximum and therefore MRL plots often have a sudden negative slope for very high values of threshold, which can be spurious as in this case.

For our example, a threshold around 20 therefore appears to be sensible. However, we will need to do some additional diagnostics to check this. We proceed by selecting the 97^{th} percentile as being the candidate threshold.

```
quantile(rain,0.97)

##  97%
## 20.6
```

The theory underpinning the GPD tells us that (if the underlying distribution satisfies our conditions) there exists a threshold above which the GPD fits the data, but the theory does not specify that the threshold necessarily must be high. Indeed, if the data are realisations from an Exponential distribution – which is a member of the GPD class, with shape parameter $\xi = 0$ – then a threshold equal to the minimum data point would be appropriate. In many cases of course, the threshold will be towards the top end of the observed data range, the motivation for the GPD as a tail model being asymptotic as the threshold goes to infinity. If sample sizes are too small, it may be the case that a suitable threshold cannot be chosen from within the range of the data with any degree of confidence.

2.4 GPD fitting in `texmex`

The generalised Pareto model for threshold excesses can be fit by using the `texmex` function `evm`, *Extreme Value Model*, which has a default `family=gpd` argument. We must specify the threshold to be used for fitting:

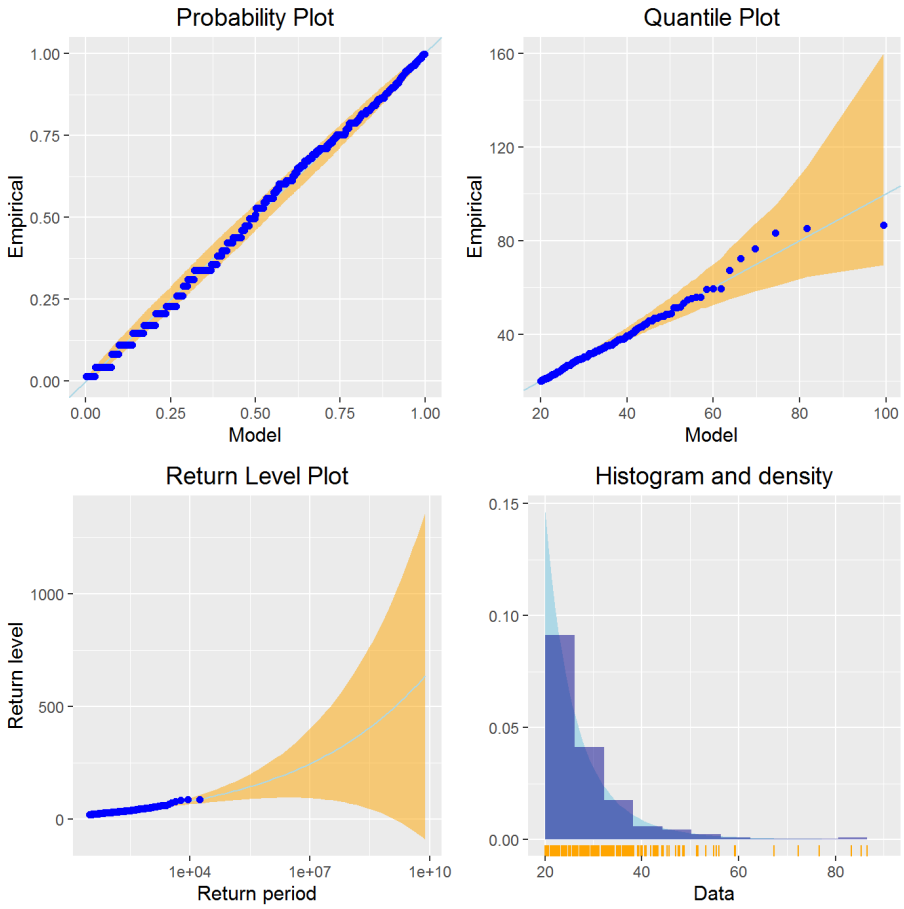
For the `rain` data:

```
rain.fit <- evm(rain,th=20)
rain.fit

## Call: evm(y = rain, th = 20)
## Family:       GPD
##
## Model fit by maximum likelihood.
##
## Convergence: TRUE
## Threshold: 20
## Rate of excess: 0.03251
##
##   Log. lik   AIC
##   -1740.834  3485.667
##
##
## Coefficients:
##        Value    SE
## phi:   1.92214  0.06348
## xi:    0.13224  0.04802
```

The estimated shape parameter shows us that with this threshold, the fitted GPD has a heavy tail, as $\hat{\xi} > 0$. This is in line with our expectations following inspection of the MRL plot.

We examine diagnostics plots to see whether these support our initial choice of threshold:

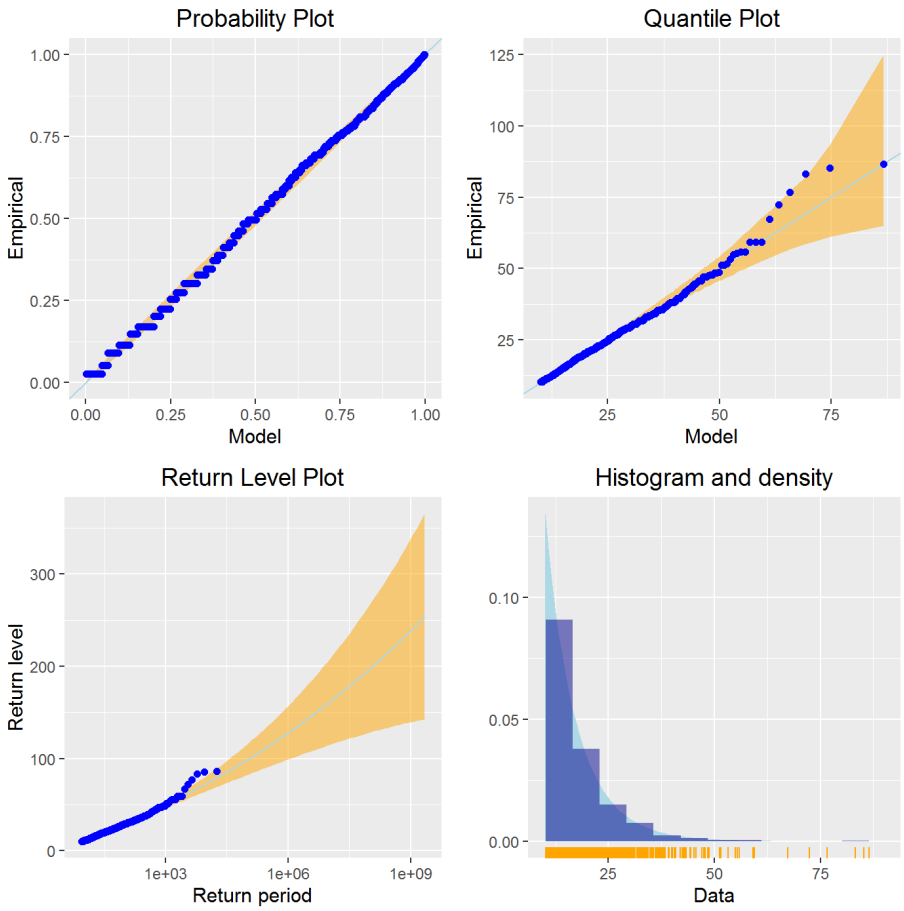
```
ggplot(rain.fit)
```



The shaded regions in the P-P and Q-Q plots indicate pointwise 95% tolerance intervals, based on 1000 simulated datasets. The shaded region in the return level plot shows 95% pointwise confidence intervals, based on a normal approximation.

The fit shown in these plots is good, with data and model agreeing well over the range of threshold exceedances. We can see how a lower threshold (perhaps that suggested by the MRL plot but not the threshold stability plot) gives an estimated slightly less heavy tail and a poorer fit:

```
ggplot(evm(rain,th=10))
```



2.5 Maximum penalized likelihood estimation

With small sample sizes, the GPD log-likelihood function often becomes flat and the optimiser can fail to converge. One way to overcome this is to penalize the likelihood by some function of the parameters. Experience suggests that the main problems may be overcome by putting fairly modest penalties on ξ.

Thus, rather than maximize the log-likelihood $l(\phi, \xi|X)$ we maximize

$$l(\phi, \xi) - \lambda\xi^2 \tag{2}$$

for some λ.

2.5.1 Choice of λ

If we exponentiate (2), the result can be written as

$$L(\phi, \xi|X)e^{-\xi^2/2\theta^2} \tag{3}$$

in which $\theta = \sqrt{\frac{1}{2\lambda}}$. The rightmost term in (3) is proportional to a Gaussian distribution centred at 0. Thus, maximum penalized likelihood estimation has a Bayesian interpretation and corresponds to maximum a posteriori estimation.

For the GPD, $\xi = -1$ corresponds to the distribution being uniform, $\xi = 0$ corresponds to it being exponential, and $\xi = 1$ corresponds to it being so heavy-tailed that its expectation is infinite. For many applications, values of $\xi = -1$ and $\xi = 1$ may be implausible, and we would expect values of ξ to be fairly close to 0. This implies a prior distribution that is Gaussian, centred at zero, with standard deviation $\theta = \frac{1}{2}$.

Since convergence issues are generally associated with ξ, we can choose a diffuse prior for ϕ, say $\phi \sim N(0, 10^4)$.

In general, we attempt to use MLE or penalized MLE with diffuse priors for both ϕ and ξ. Prior distribution $\xi \sim N(0, \frac{1}{4})$ independently of a diffuse prior on ϕ can be used when convergence issues arise:

```
pp <- list(c(0, 0), diag(c(10^4, .25)))
rain.pen <- evm(rain, qu=.97, priorParameters = pp, prior="gaussian")
```

in which `priorParameters` is a list containing the mean $(0, 0)^T$ and covariance matrix of the prior Gaussian distribution.

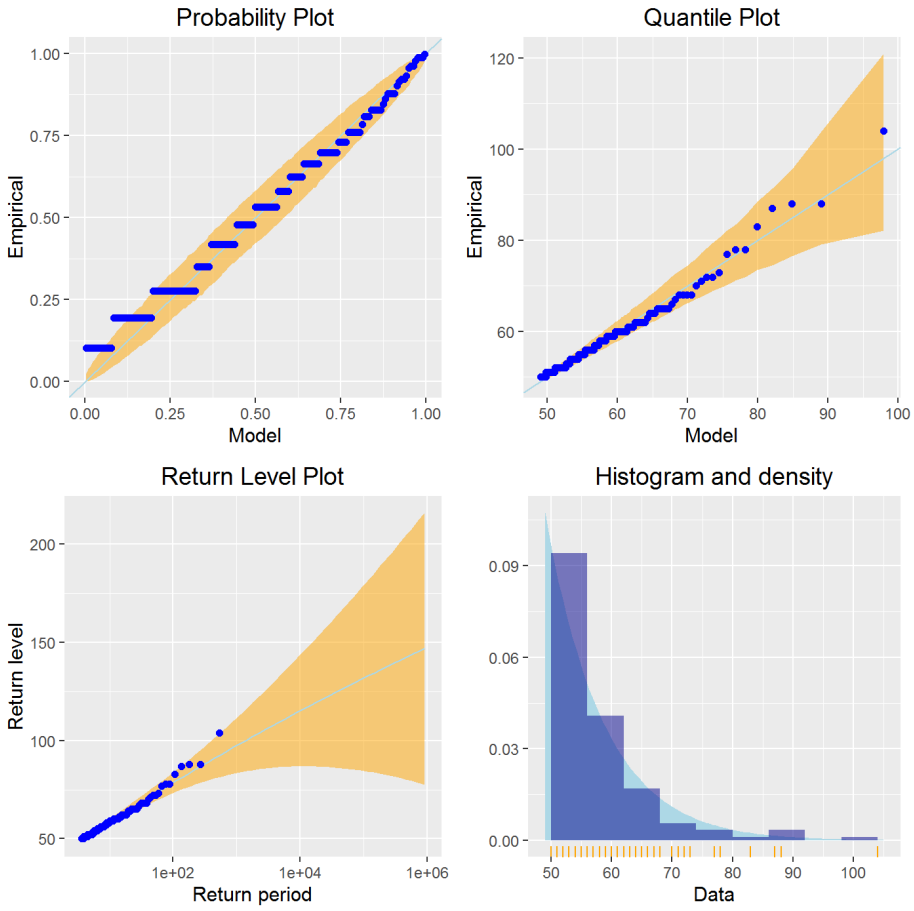
2.6 Covariate modelling

We can fit GPD models with covariates in ϕ, in ξ, in neither, or in both. We use the `winter` air pollution dataset to illustrate this model fitting, with NO2 as the response and O3 as the explanatory variable. Plots of this data in Section 1.2.2 suggested a negative association between these variables. A threshold corresponding to the 70% quantile of the NO2 variable was chosen using the diagnostic techniques in the previous section (output not shown here). As a starting point we fit the GPD with no covariates:

```
airpoll <- evm(NO2, data=winter, qu=.7, penalty="none", family=gpd)
airpoll

## Call: evm(y = NO2, data = winter, family = gpd, qu = 0.7, penalty = "none")
## Family:       GPD
##
## Model fit by maximum likelihood.
##
## Convergence: TRUE
```

```
## Threshold: 49
## Rate of excess: 0.2763
##
##   Log. lik   AIC
##   -470.9206  945.8413
##
##
## Coefficients:
##                    Value     SE
## phi: (Intercept)   2.23157   0.11010
## xi: (Intercept)   -0.02790   0.07297
```

```
ggplot(airpoll)
```

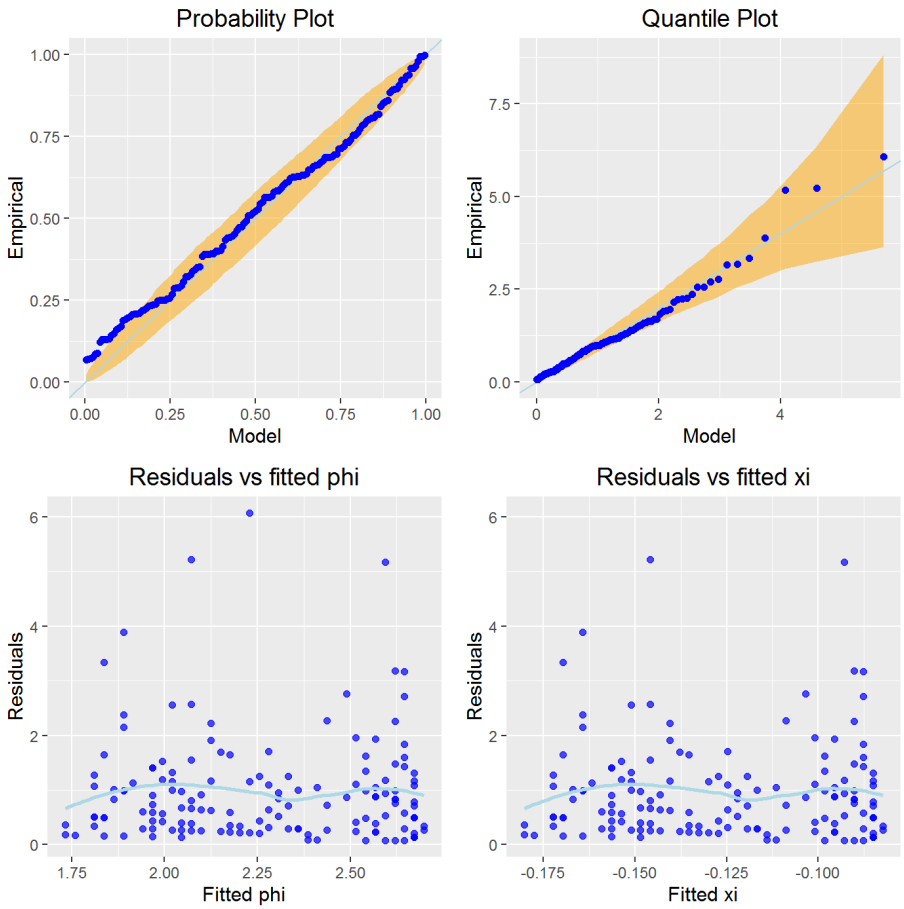


The plots show no systematic departure of the data from the model at this choice of threshold, so we proceed to fit various models with covariates:

```
airpoll1 <- update(airpoll, phi= ~O3, xi= ~O3)
airpoll2 <- update(airpoll, phi= ~O3)
airpoll3 <- update(airpoll, xi= ~O3)
```

The default model diagnostic plots for the model are different when there are covariates included in the model. Here we look at diagnostics for the model with O3 included in the linear predictors for both ϕ and ξ:

```
ggplot(airpoll1)
```



Since there is a covariate in the model the probability and quantile plots are constructed using the model residuals, which are exponential under the fitted model. We also have a plot of the residuals against the fitted parameters for any parameter that is modelled using a covariate (in this case the scale parameter ϕ and shape parameter ξ). A well fitting model should have homogeneity of residuals across different values of the fitted parameter. These diagnostic plots give no cause for concern. There are no return level or histogram/density plots

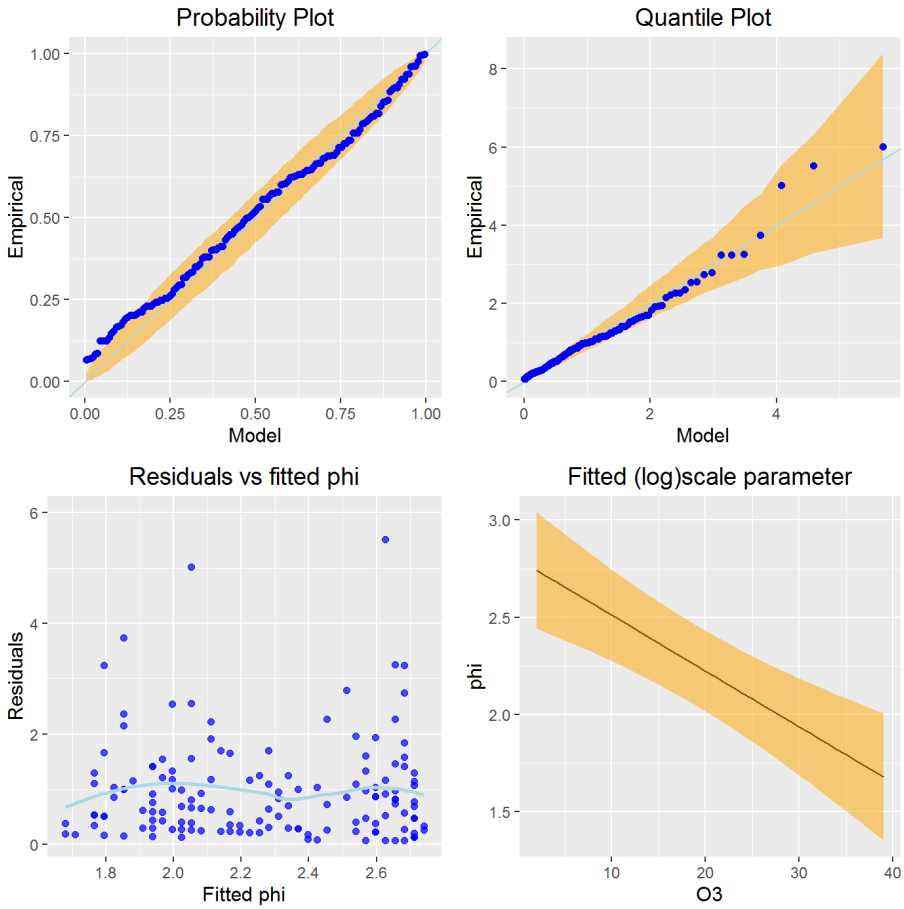
produced since the estimates of these quantities depend on the precise values taken by the covariates in the model.

```
AIC(airpoll)

## [1] 945.8413

AIC(airpoll1)

## [1] 932.6114

AIC(airpoll2)

## [1] 930.7401

AIC(airpoll3)

## [1] 937.6038
```

AIC is lowest for `airpoll2` which has O3 as a covariate affecting the scale parameter of the GPD, but not the shape parameter. Since `airpoll2` has the lowest AIC we prefer that model.

We now examine more detailed model diagnostics for our preferred model, `airpoll2`:

```
g3 <- ggplot(airpoll2,plot.=FALSE)
g4 <- ggplot(predict(airpoll2,type="lp",ci.fit=TRUE))[[1]] +
    ggtitle("Fitted (log)scale parameter")

grid.arrange(g3[[1]],g3[[2]],g3[[3]],g4,ncol=2)
```

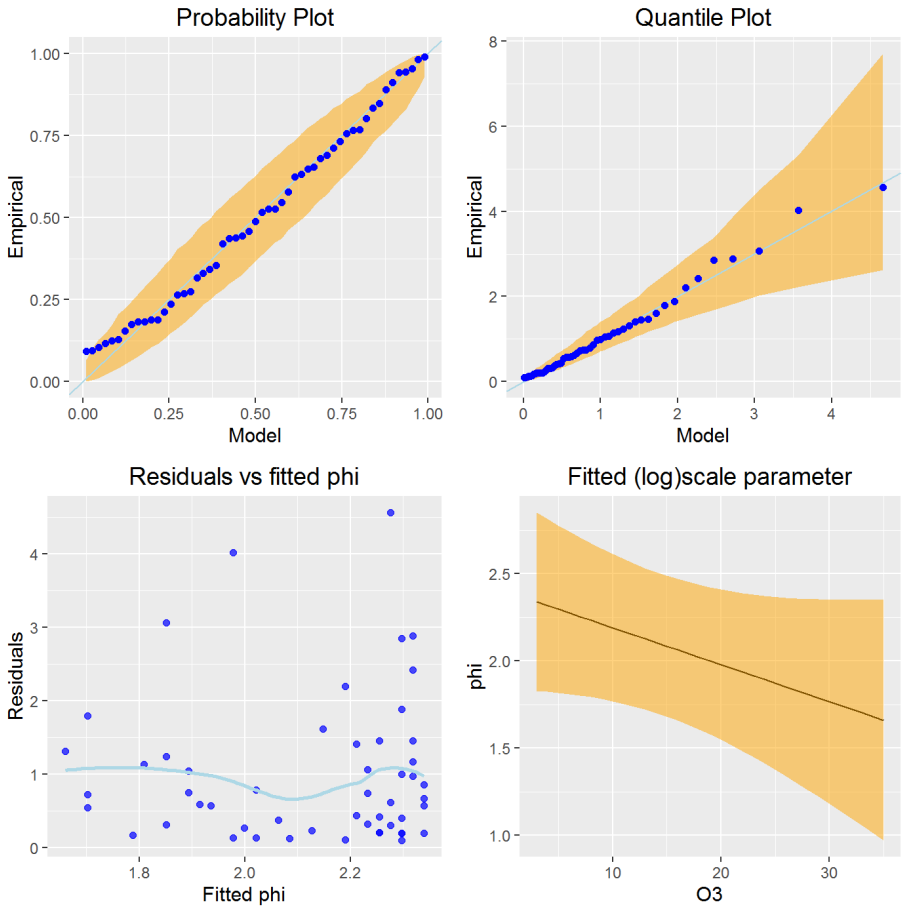



```
summary(airpoll2)

## Call: evm(y = NO2, data = winter, family = gpd, qu = 0.7, penalty = "none",
##     phi = ~O3)
##
## Family:       GPD
##
## Model fit by maximum likelihood.
##
## Convergence: TRUE
## Threshold: 49
## Rate of excess: 0.276
##
## Log-lik. AIC
## -462.3701   931
##
```

```
## Coefficients:
##                    Value     SE        t
## phi: (Intercept)   2.79794   0.16151  17.32411
## phi: O3           -0.02869   0.00646  -4.44422
## xi: (Intercept)   -0.12542   0.06637  -1.88968
##
## 1000 simulated data sets compared against observed data QQ-plot.
## Quantile of the observed MSE:  0.392
## 20 observations (13.605%) outside the 95% simulated envelope.
```

The `summary` command reveals an alarming number of points to lie outside of the 95% tolerance interval constructed for the Q-Q plot; the corresponding plot shows these points to be among those lying closest to the fitting threshold. This suggests that we should re-visit the whole model selection procedure again at a higher threshold (details not shown here). We eventually settle on the following model, fit at a threshold corresponding to the 90% quantile:

```
airpoll4 <- update(airpoll2,qu=0.9)
g5 <- ggplot(airpoll4,plot.=FALSE)
g6 <- ggplot(predict(airpoll4,type="lp",ci.fit=TRUE))[[1]] +
    ggtitle("Fitted (log)scale parameter")

grid.arrange(g5[[1]],g5[[2]],g5[[3]],g6,ncol=2)
```




```
summary(airpoll4)

## Call: evm(y = NO2, data = winter, family = gpd, qu = 0.9, penalty = "none",
##     phi = ~O3)
##
## Family:       GPD
##
## Model fit by maximum likelihood.
##
## Convergence: TRUE
## Threshold: 58
## Rate of excess: 0.0996
##
## Log-lik. AIC
## -166.49   339
##
```

```
## Coefficients:
##                   Value    SE       t
## phi: (Intercept)   2.4028   0.2886   8.3248
## phi: O3           -0.0212   0.0140  -1.5196
## xi: (Intercept)    0.0150   0.1547   0.0968
##
## 1000 simulated data sets compared against observed data QQ-plot.
## Quantile of the observed MSE:  0.008
## 1 observations (1.887%) outside the 95% simulated envelope.
```

The fit of this model is adequate, having apparently captured the dependence of the scale parameter on O3. The final plot shows the nature of this relationship, with larger values of O3 giving lower values of ϕ. This concurs with the observed negative relationship between the variables shown in the original scatter plot, Section 1.2.2.

2.7 GPD parameter uncertainty

We examine briefly here Information Matrix summaries and Bootstrap estimates of GPD parameter uncertainty, before going on to use a Bayesian simulation based approach to estimation of our GPD model parameters and associated uncertainty.

2.7.1 Information matrix based approaches

When the GPD model is fit by using the default maximum likelihood estimation, an estimate of the covariance matrix of model parameters is returned. The default procedure for estimating this covariance matrix is `cov="observed"` in which case the observed information matrix is used, as given in Appendix A of Davison and Smith [2]. The only other option is `cov = "numeric"` in which case a numerical approximation of the Hessian is used (see the help for `optim`). In some cases, particularly with small samples, the numerical approximation can be quite different from the closed form (`cov="observed"`) result, and the value derived from the observed information should be preferred.

For our fitted model, we compare the two approaches and find that the alternative methods give almost identical estimates of the Information matrix:

```
airpoll2$cov
```

```
##               [,1]          [,2]          [,3]
## [1,]  0.026083966 -7.906210e-04 -5.609481e-03
## [2,] -0.000790621  4.167378e-05  2.497265e-05
## [3,] -0.005609481  2.497265e-05  4.405064e-03
```

```
update(airpoll2,cov="numeric")$cov
```

```
##                [,1]          [,2]          [,3]
## [1,]  0.0260203685 -7.882263e-04 -0.0055858130
## [2,] -0.0007882263  4.159649e-05  0.0000237917
## [3,] -0.0055858130  2.379170e-05  0.0044028665
```

For small samples, the underlying log-likelihood may be far from quadratic, and the resulting estimates of standard errors derived using either of these methods are liable to approximate poorly the true standard errors.

2.7.2 Parametric Bootstrap approach

An alternative approach to uncertainty estimation is to use a parametric bootstrap – which does capture the asymmetry of the log-likelihood surface around the maximum likelihood estimates. This is carried out for our fitted model in `texmex` as follows:

```
boot <- evmBoot(airpoll2, trace=1001)
summary(boot)

## evmBoot(o = airpoll2, trace = 1001)
##                   phi: (Intercept)       phi: O3 xi: (Intercept)
## Original                 2.7979363 -0.0286897805     -0.12541904
## Bootstrap mean           2.8215086 -0.0288725201     -0.15717055
## Bias                     0.0235723 -0.0001827396     -0.03175151
## SD                       0.1741919  0.0065112370      0.09171290
## Bootstrap median         2.8272419 -0.0288481238     -0.15497853
##
## Correlation:
##                   phi: (Intercept)      phi: O3 xi: (Intercept)
## phi: (Intercept)         1.0000000 -0.698315136    -0.571992839
## phi: O3                 -0.6983151  1.000000000     0.001607933
## xi: (Intercept)         -0.5719928  0.001607933     1.000000000
```

We can compare these reported standard deviations with the correponding estimates derived from the Observed Information matrix estimate – these are close although not identical, with the largest disagreement occurring for the shape parameter. This is typical behaviour of the GPD model.

```
sqrt(diag(airpoll2$cov))

## [1] 0.161505314 0.006455523 0.066370657
```

We can also compare the bootstrap based estimate of the parameter correlation matrix with that derived from the Observed Information matrix:

```
cov2cor(airpoll2$cov)
```

```
##             [,1]        [,2]        [,3]
## [1,]  1.0000000 -0.75831579 -0.52331083
## [2,] -0.7583158  1.00000000  0.05828503
## [3,] -0.5233108  0.05828503  1.00000000
```

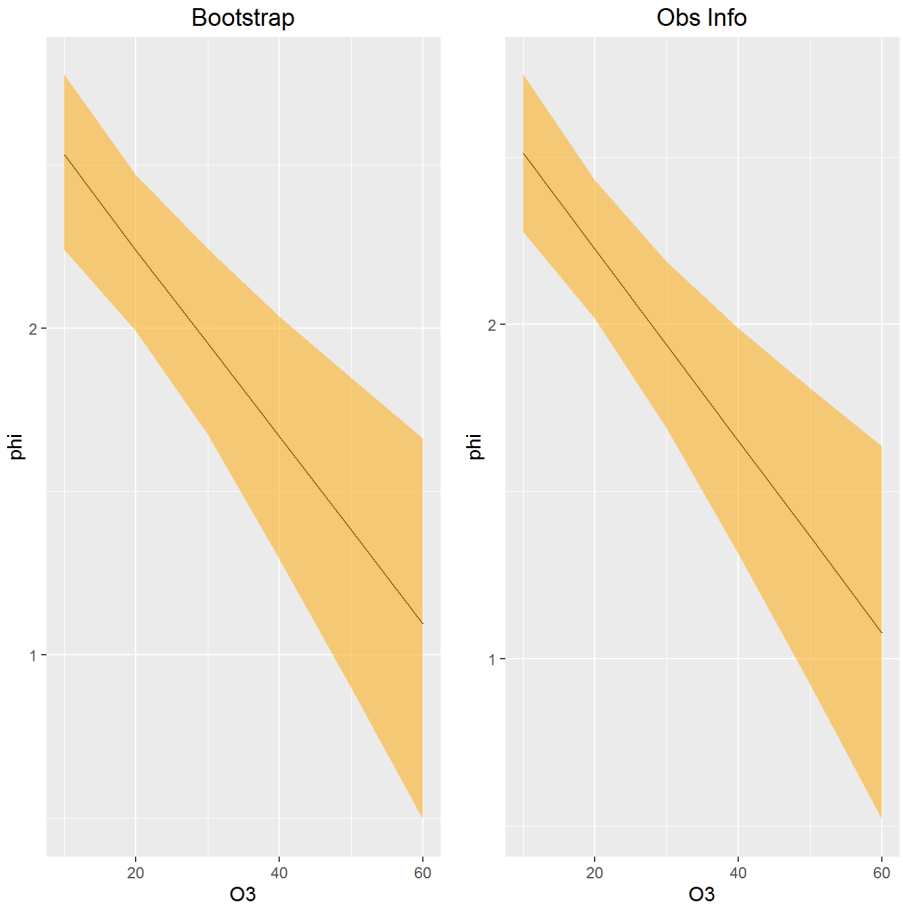
```
cov2cor(summary(boot)$covariance)
```

```
##                   phi: (Intercept)      phi: O3 xi: (Intercept)
## phi: (Intercept)        1.0000000 -0.698315136    -0.571992839
## phi: O3                -0.6983151  1.000000000     0.001607933
## xi: (Intercept)        -0.5719928  0.001607933     1.000000000
```

Estimates of this correlation matrix are similar although not identical, as anticipated.

Focussing on the covariance matrix of the parameter estimates is misleading and does not let us explore the asymmetric nature of the uncertainty about the parameter estimates. This can often be better seen in the bootstrap based confidence intervals for the model parameters shown in the following plots, although the asymmetry is not pronounced in this example:

```
O3 <- data.frame(O3=seq(10,60,len=6))

g7 <- ggplot(predict(boot,    newdata=O3,type="lp",
                     ci.fit=TRUE))[[1]] +
    ggtitle("Bootstrap")
g8 <- ggplot(predict(airpoll2,newdata=O3,type="lp",
                     ci.fit=TRUE))[[1]] +
    ggtitle("Obs Info")

grid.arrange(g7,g8,ncol=2)
```



2.8 Bayesian estimation

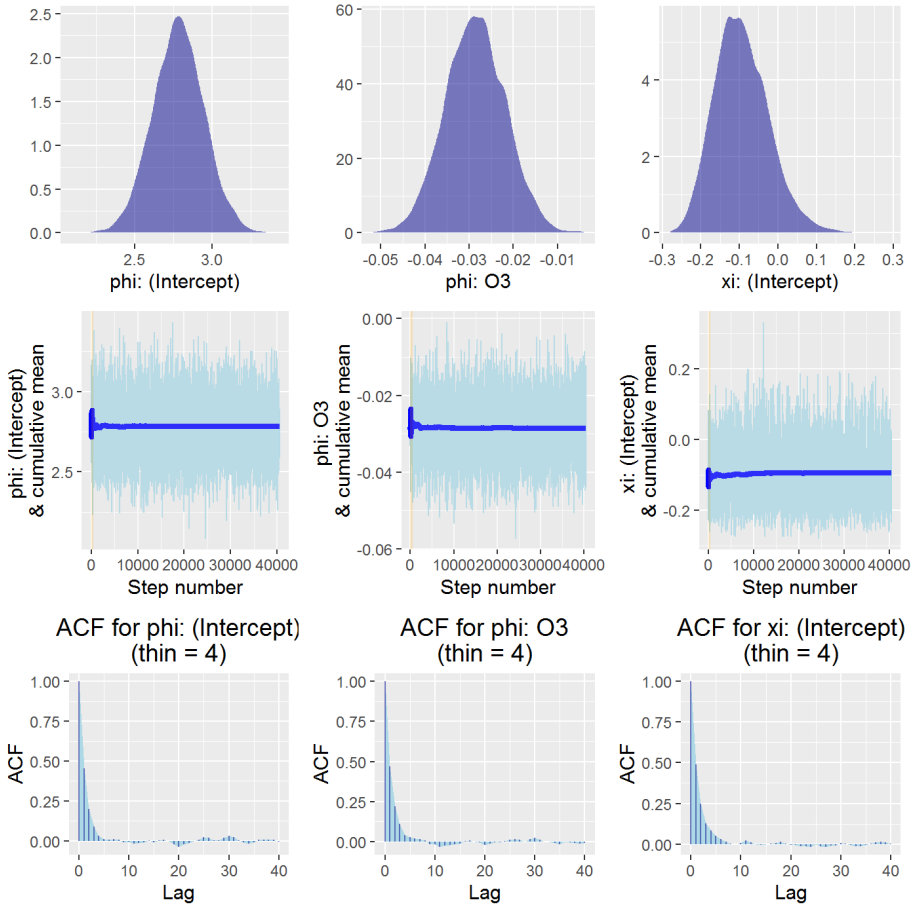
A further alternative approach to uncertainty estimation which accurately reflects the asymmetric nature of the uncertainty is offered by Bayesian simulation based methods. In `texmex` we can simulate from the posterior distributions of the parameters by using the `evm` function again, this time using `method = "simulate"` to tell the function to simulate from the joint posterior distribution of the parameters.

```
airpollSim <- evm(NO2, data=winter, qu=.7,
                  phi=~O3, method="simulate",
                  verbose=FALSE)
```

Equivalently, the Bayesian estimation based on MCMC can also be instigated by the use of the function `update` on the previously chosen model. The method of estimation is changed from `"optimize"` – under which estimation is carried

out using (penalized) maximum likelihood – to `"simulate"` – under which a Metropolis algorithm is used to simulate from the joint posterior distribution of the parameters. For our preferred model, `airpoll2`, this is implemented as follows:

```
airpollSim <- update(airpoll2,method="simulate",penalty="gaussian",
                     verbose=FALSE)
ggplot(airpollSim)
```




```
airpollSim

## evm(y = NO2, data = winter, family = gpd, qu = 0.7, penalty = "gaussian",
##     phi = ~O3, method = "simulate", verbose = FALSE)
## Family:       GPD
## Acceptance rate:  0.33
##
```

```
## MAP estimates:
## phi: (Intercept)          phi: O3  xi: (Intercept)
##       2.79793633      -0.02868978      -0.12541904
##
## Posterior means:
## phi: (Intercept)          phi: O3  xi: (Intercept)
##       2.78233246      -0.02861596      -0.09374727
```

The plots of the Markov chains ought to look like "fat hairy caterpillars" if the algorithm has converged on its target distribution. Also, the cumulative means of the chains should converge, the acceptance rate should not be too high or too low, and the autocorrelation functions should rapidly decay to zero. We conclude from the plots that there is no evidence against convergence of our Markov chains, although we should probably thin our output further to obtain a chain that is closer to independent (the default is to thin to every 4 observations). Here we retain the burn-in value of 500 but now discard all but every 20th observation, resulting in an autocorrelation function which decays more rapidly to zero. This results in the retention of 2000 values after discarding the burn-in and applying the thinning. (Note that the observations are not discarded destructively and we can use the `thinAndBurn` function repeatedly to examine the impact of using different values of `burn` and `thin`.)

```
airpollSim <- thinAndBurn(airpollSim, burn=500, thin = 20)
dim(airpollSim$param)

## [1] 2000    3

summary(airpollSim)

## [[1]]
## Family:       GPD
##
## [[2]]
##                   Posterior mean          SD
## phi: (Intercept)     2.78036128 0.164530801
## phi: O3             -0.02854285 0.006875537
## xi: (Intercept)     -0.09427419 0.069438660
##
## attr(,"class")
## [1] "summary.evm.sim"
```

We can use the `predict` method to obtain the linear predictors for the model parameters for each unique combination of any covariates that may be in the model.

```
O3 <- data.frame(O3=seq(20,50,by=10))
predict(airpollSim,newdata=O3,type="lp")

## Linear predictors:
##    phi      xi O3
## 1 2.21 -0.0943 20
## 2 1.92 -0.0943 30
## 3 1.64 -0.0943 40
## 4 1.35 -0.0943 50

predict(airpollSim,newdata=O3,M=1000)

## M = 1000 predicted return level:
##    res O3
## 1 89.4 20
## 2 79.5 30
## 3 72.1 40
## 4 66.6 50
```

To see linear predictors for the original dataset, simply omit the `newdata` argument (output not shown):

```
predict(airpollSim,type="lp")
predict(airpollSim,M=1000)
```

Setting the argument `all = TRUE` returns the linear predictors for all of the simulated parameter values in the (thinned) chains:

```
airpollParams <- predict(airpollSim, newdata=O3, type="lp", all=TRUE)
```

The returned object contains a list called `link` with one item for each unique value of the covariate(s). The following shows the first five simulated values of (ϕ, ξ) for covariate `O3 = 20`:

```
airpollParams$obj$link[[1]][1:5,]

##           phi           xi O3
## [1,] 2.141077 -0.003333331 20
## [2,] 2.187448 -0.094458714 20
## [3,] 2.179649 -0.096721497 20
## [4,] 2.280159 -0.139971066 20
## [5,] 2.127559 -0.060570982 20
```

2.9 Predicted return levels

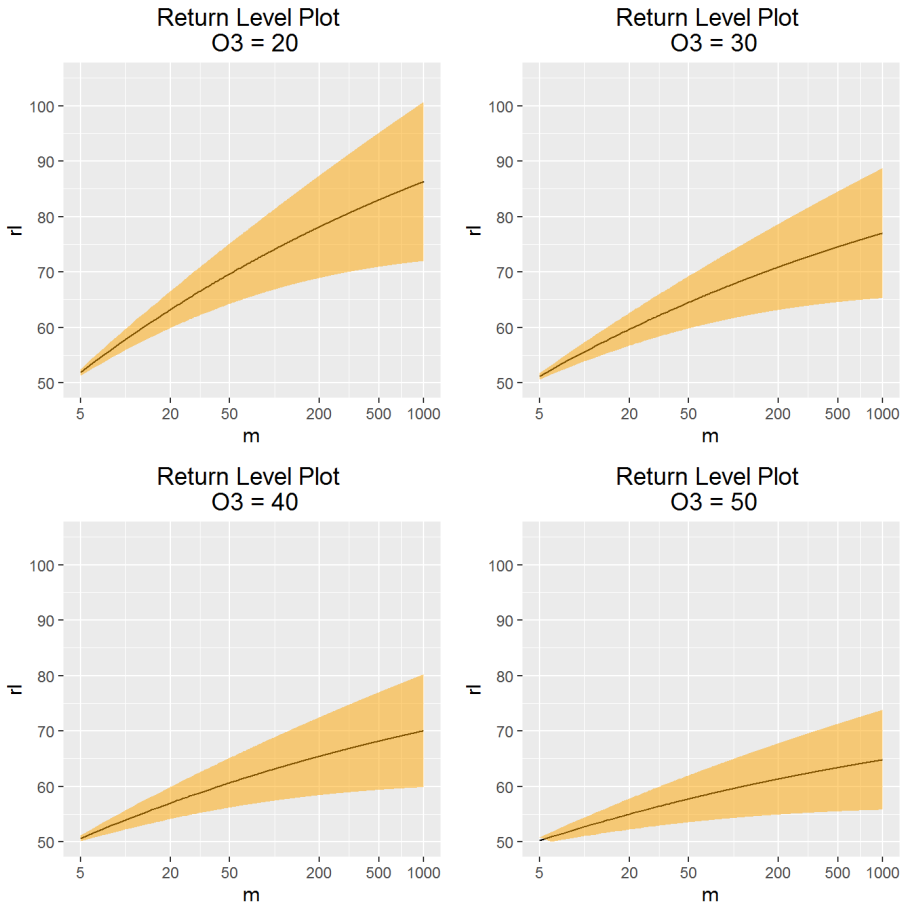
The general definition of an m-observation return level for the GPD is:

$$x_m = u + \frac{\sigma}{\xi}\{(mp)^{\xi} - 1\}. \tag{4}$$

Here p is the probability of exceeding the GPD fitting threshold u and m is a large value, so that x_m is termed the m-observation return level and represents the maximum value of x expected to be seen in m observations.

The effect of the O3 variable on the fitted GPD is seen clearly when we look at return levels and associated plots for different levels of this variable:

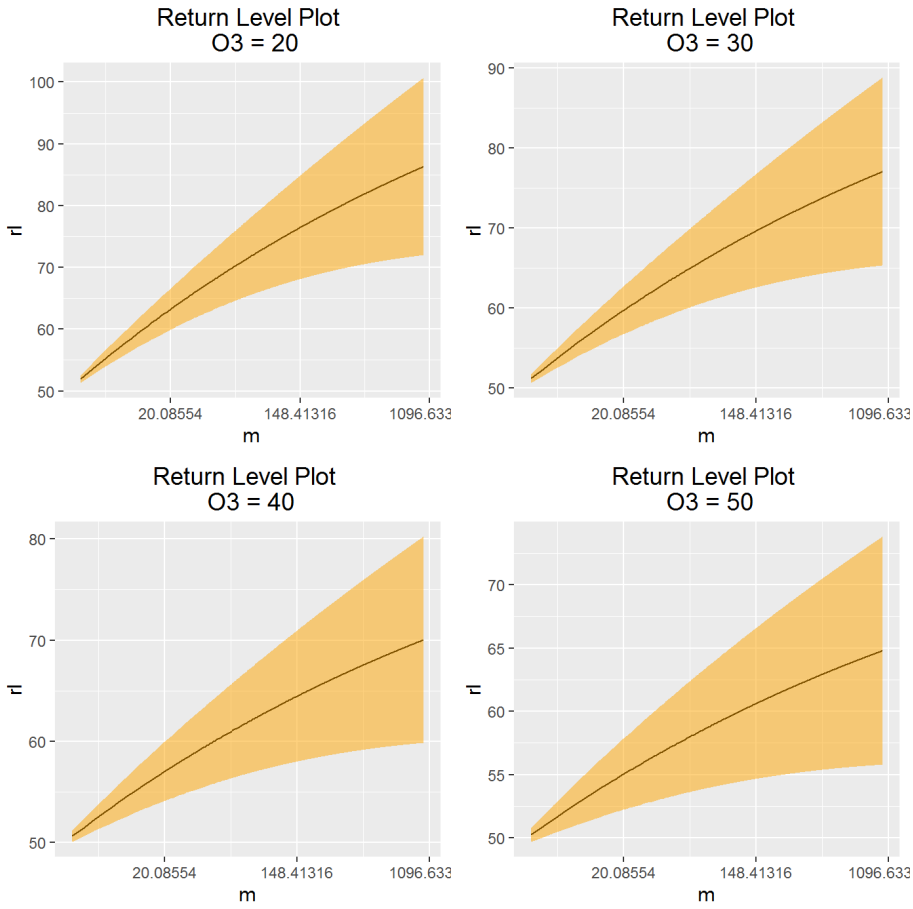
```
O3 <- data.frame(O3=seq(20,50,by=10))
pred <- predict(airpoll2,newdata=O3,M=5:1000,ci.fit=TRUE)
pred$obj[c(1,496,996)]

## $M.5
##         RL     2.5%    97.5% O3
## 1 51.92864 51.31966 52.53763 20
## 2 51.19821 50.64622 51.75019 30
## 3 50.64995 50.09070 51.20920 40
## 4 50.23843 49.68634 50.79052 50
##
## $M.500
##         RL     2.5%    97.5% O3
## 1 82.98659 70.90954 95.06364 20
## 2 74.50994 64.54292 84.47696 30
## 3 68.14747 59.35096 76.94398 40
## 4 63.37187 55.48075 71.26299 50
##
## $M.1000
##         RL     2.5%     97.5% O3
## 1 86.29465 71.95139 100.63791 20
## 2 76.99293 65.27717  88.70869 30
## 3 70.01117 59.83792  80.18442 40
## 4 64.77074 55.77008  73.77141 50

pred$call

## predict.evmOpt(object = airpoll2, M = 5:1000, newdata = O3, ci.fit = TRUE)

g9 <- ggplot(pred)
xAxis <- scale_x_continuous(breaks=c(5,20,50,200,500,1000),trans="log")
yAxis <- scale_y_continuous(limits=c(50,105))
grid.arrange(g9[[1]] + xAxis + yAxis,
             g9[[2]] + xAxis + yAxis,
             g9[[3]] + xAxis + yAxis,
             g9[[4]] + xAxis + yAxis,ncol=2)
```



The *Return period* is in units of *numbers of observations*, in this case, number of winter days. *Return level* is in the same units as NO2 variable to which the GPD model has been fit.

This plot shows how the different values of scale parameter affect the size of return levels associated with different return periods but not the shape of this function. The shape parameter ξ is common to each of the four models used for prediction here – this is emphasised if we allow the axes for plotting to differ for the four levels of O3:

```
grid.arrange(g9[[1]],g9[[2]],g9[[3]],g9[[4]],ncol=2)
```

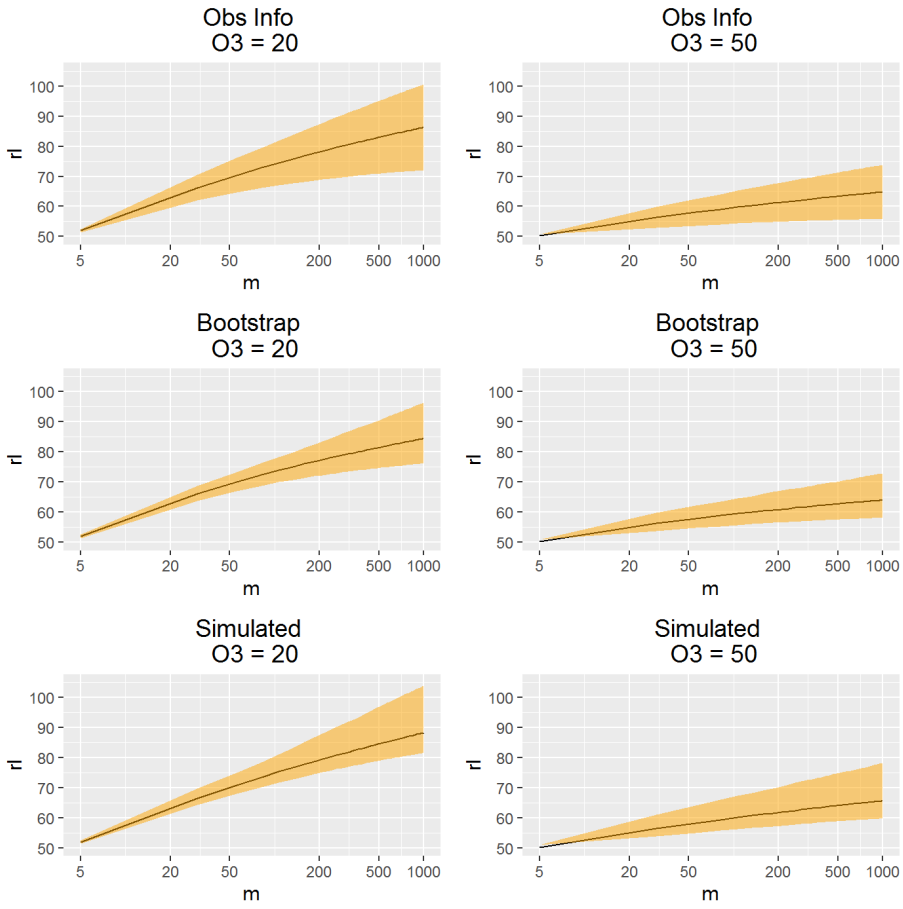


We can see that the underlying shapes of the four curves are identical, the main differences emphasised here is the greater uncertainty associated with the highest value of O3 examined here, which is beyond the range observed in the dataset.

The default method for estimating these confidence intervals is to use the Information Matrix and quadratic approximation, however this can lead to poor estimates as this approach gives symmetric intervals centered on the point estimates. If we are to extrapolate far beyond the range of the data, then it can be preferable to recognise the inherent reduction in certainty that occurs as we move away from the data where the information dwells. This is better reflected in the asymmetric confidence/credible intervals obtained by using either the bootstrap or Bayesian simuation based approach:

```
O3 <- data.frame(O3=c(20,50))
M <- seq(5,1000,len=40)
g11 <- ggplot(predict(airpoll2,newdata=O3,M=M,ci.fit=TRUE),main="Obs Info")
g12 <- ggplot(predict(boot,newdata=O3,M=M,ci.fit=TRUE),main="Bootstrap")
```

```
g13 <- ggplot(predict(airpollSim,newdata=O3,M=M,ci.fit=TRUE),main="Simulated")
grid.arrange(g11[[1]] + xAxis+yAxis,g11[[2]] + xAxis+yAxis,
             g12[[1]] + xAxis+yAxis,g12[[2]] + xAxis+yAxis,
             g13[[1]] + xAxis+yAxis,g13[[2]] + xAxis+yAxis,ncol=2)
```



The asymmetry in the bootstrap based confidence intervals and the Bayesian simulation based credible intervals for high return levels at large values of the covariate is marked here.

3 Generalized Extreme Value models

Whereas GPD models have an asymptotic motivation as models for threshold exceedances, the Generalised Extreme Value (GEV) distribution is derived as the limiting distribution for observations arising as maxima of IID observations.

We introduce the GEV distribution now, but refer to Coles [1] for more details of this family of distributions, and how it arises.

3.1 Extreme value modelling and asymptotic motivation for the GEV

In this section, we show how to fit the generalised Extreme Value Distribution, GEV(μ, σ, ξ) to data points arising as sample maxima. The GEV has distribution function

$$G(x) = \exp\left[-\left\{1 + \xi\left(\frac{x-\mu}{\sigma}\right)\right\}^{-1/\xi}\right], \tag{5}$$

for x satisfying $1 + \xi(x-u)/\sigma > 0$. The location parameter μ satisfies $-\infty < \mu < \infty$, scale parameter $\sigma > 0$ and shape parameter ξ satisfies $-\infty < \xi < \infty$. The range of possible values taken by realisations from the GEV depends on the parameter values, with the distribution having a finite upper end point (short tailed) if the shape parameter is negative ($x \leq \mu - \sigma/\xi$ if $\xi < 0$) and an infinite tail otherwise ($x < \infty$ if $\xi \geq 0$). When $\xi = 0$, the GEV corresponds exactly to the Gumbel distribution.

Extreme value theory tells us that under appropriate normalisation of the sample maxima, as the underlying sample size from which maxima are taken tends to infinity, the limiting distribution of the sample maxima must fall in the Generalised Extreme Value family of distributions (given certain conditions concerning non-degeneracy of the limit distribution and smoothness of the distribution of the original variable).

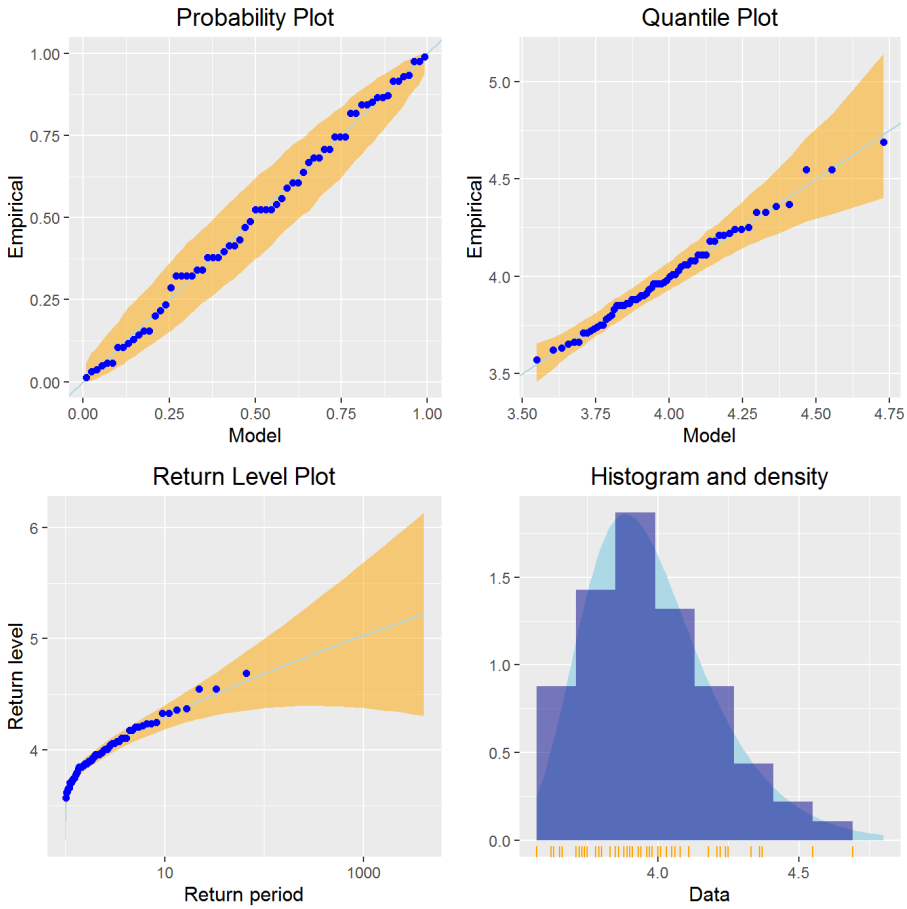
3.2 Parameterization

As for the GPD (Section 2.2), the usual parameterization of the GEV (equation (5)) is in terms of its location, scale and shape parameters μ, σ and ξ respectively. For the same reasons as those given for the GPD in Section 2.2, we choose to reparameterize in terms of $\phi = \log\sigma$. Comments made in this section regarding this parameterization in the context of the GPD apply equally to the GEV.

3.3 GEV fitting in `texmex`

We use the annual maxima sea-level observations in the dataset `portpirie` to illustrate the fitting of the GEV in `texmex`. The function `evm` (Extreme Value Model) is called, this time with the argument `family=gev`, which fits the GEV model rather than the default family, GPD. Diagnostic plots are constructed in the usual way:

```
port <- evm(SeaLevel,data=portpirie,family=gev)
ggplot(port)
```




```
port

## Call: evm(y = SeaLevel, data = portpirie, family = gev)
## Family:       GEV
##
## Model fit by maximum likelihood.
##
## Convergence: TRUE
##
##   Log. lik  AIC
##   4.339058  -2.678116
##
##
## Coefficients:
##                    Value     SE
## mu: (Intercept)    3.87473   0.02793
```

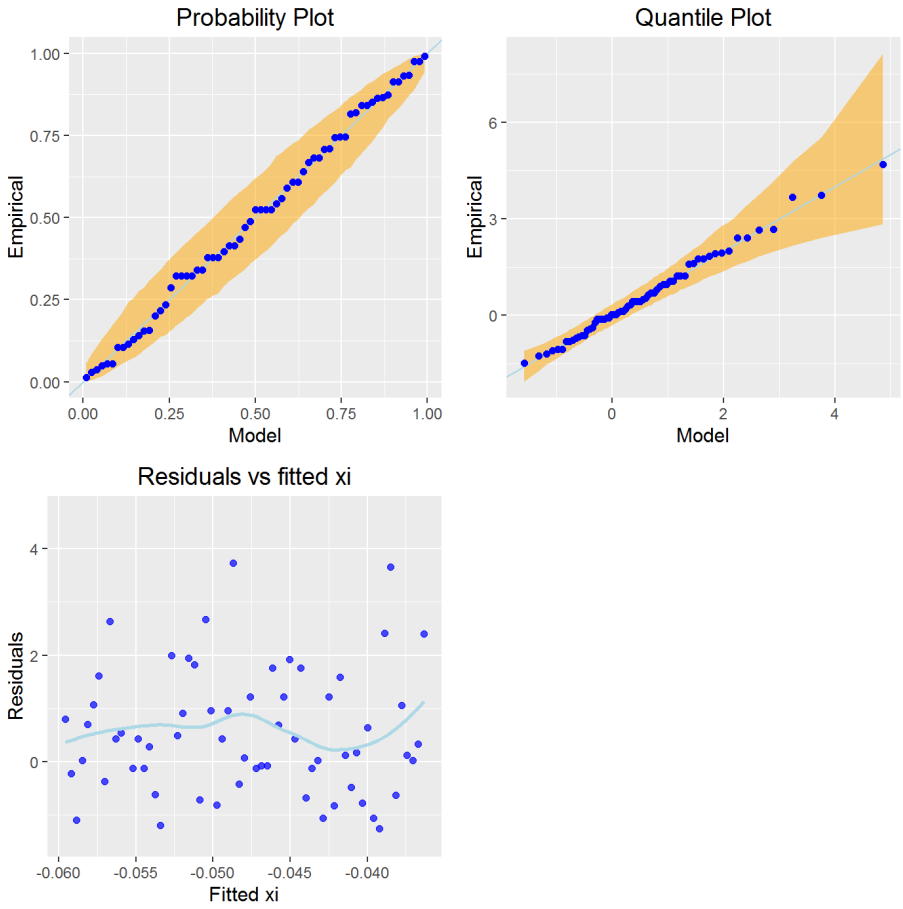
```
## phi: (Intercept)  -1.61930   0.10225
## xi: (Intercept)   -0.05014   0.09824
```

The fit of the GEV to the portpirie sea-level annual maxima appears to be good.

There is no analogue of threshold choice for the GPD (by using MRL plots and threshold stability plots), and as such there are fewer diagnostics to support the fitting of the GEV distribution. Where poor fit is observed, it may be due to having taken maxima of an insufficient number of observations (for example, if the data are annual maxima then it may be that one year of data is not a large enough sample size from which to draw maxima). Sometimes, but not always, it is possible to go back to the original data from which the maxima have been constructed and take maxima of a larger number of observations. This usually results in a smaller number of observed maxima so there is an obvious trade-off between bias and variance here.

Covariates may be included in the GEV models in the same manner as illulstrated in Section 2.6 for the GPD. As an exercise, we try fitting `Year` as a covariate to the `portpirie` sea level observations:

```
port1 <- update(port,mu=~Year)
port2 <- update(port,phi=~Year)
port3 <- update(port,xi=~I((Year-1955)/1955),start=c(coef(port),0.001))
ggplot(port3)
```



An examination of the AIC for each of these fitted models (`AIC(port)` etc.) reveals there to be no evidence of an effect of Year on any of the GEV model parameters.

3.4 Variants on basic GEV model fitting

The various different model estimation strategies outlined for the GDP model above can be applied equally to the estimation of GEV models. We can use MCMC to estimate the GEV model parameters, just as for the GPD in Section 2.8:

```
evm(SeaLevel,data=portpirie,family=gev,method="simulate")
```

Output may be processed and examined in the same manner as for the GPD model, and we refer to the details of Section 2.8 for outline examples in this area.

Informative priors or penalties may be applied to model parameters as follows:

```
pp <- list(c(0, 0, 0), diag(c(10^4, 10^4, .25)))
update(port, priorParameters = pp, prior="gaussian")
```

```
## Call: evm(y = SeaLevel, data = portpirie, family = gev, priorParameters = pp,
##     prior = "gaussian")
## Family:       GEV
##
## Model fit by penalized maximum likelihood.
##
## Convergence: TRUE
##
##   Log lik.  Penalized log lik.  AIC
##   4.338405  4.327974            -2.676810
##
##
## Coefficients:
##                    Value     SE
## mu: (Intercept)    3.87434   0.02776
## phi: (Intercept)  -1.62061   0.10152
## xi: (Intercept)   -0.04655   0.09535
```

This was illustrated in more detail for the GPD model in Section 2.5.

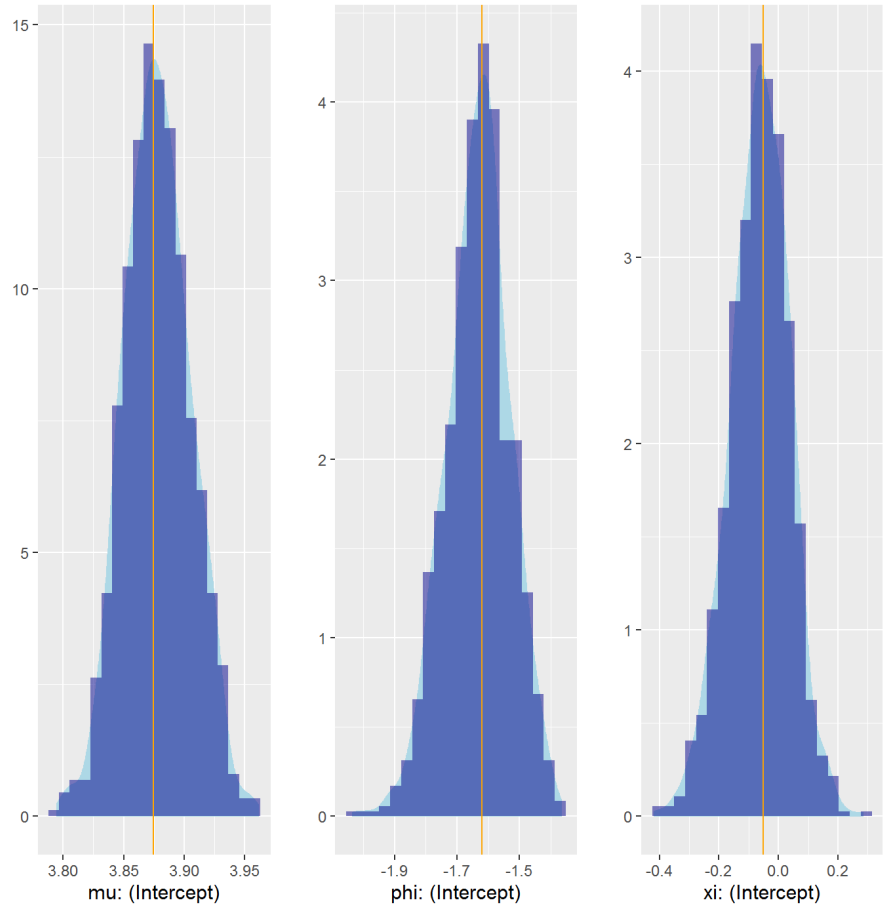
Currently, there is no implementation in `texmex` of the Observed Information Matrix estimator of the covariance matrix of parameter estimates, instead estimates of this matrix are obtained using a numerical approximation to the Hessian matrix.

Uncertainty estimation via the parametric bootstrap is carried out for our fitted model in `texmex` in the same manner as for the GPD, as follows:

```
boot <- evmBoot(port, trace=1001)
summary(boot)
```

```
## evmBoot(o = port, trace = 1001)
##                   mu: (Intercept) phi: (Intercept) xi: (Intercept)
## Original               3.874731480      -1.61930145     -0.05014022
## Bootstrap mean         3.877748549      -1.63265453     -0.06358499
## Bias                   0.003017069      -0.01335308     -0.01344477
## SD                     0.027184507       0.10164127      0.09916211
## Bootstrap median       3.876968083      -1.62688036     -0.06101311
##
## Correlation:
##                   mu: (Intercept) phi: (Intercept) xi: (Intercept)
## mu: (Intercept)         1.000000        0.3312010      -0.3898590
```

```
## phi: (Intercept)        0.331201        1.0000000       -0.3658741
## xi: (Intercept)        -0.389859       -0.3658741        1.0000000

ggplot(boot)
```



This plot shows the bootstrap distributions of the model parameter estimates, for the GEV fitted to the `portpirie` dataset.

3.5 Return level estimation

Quantiles of the fitted GEV distribution can be estimated by using the estimated model parameters in the following equation:

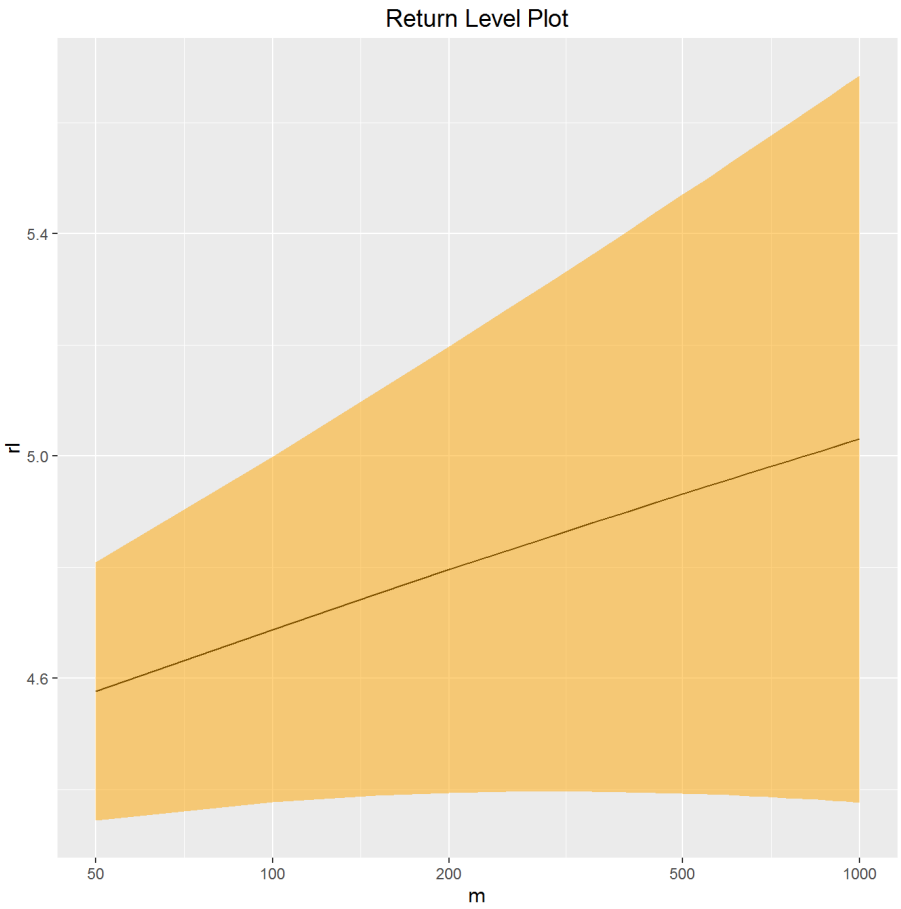
$$x_p = \begin{cases} \mu - \frac{\sigma}{\xi}[1 - \{-\log(1-p)\}^{-\xi}, & \text{for } \xi \neq 0; \\ \mu - \sigma \log\{-\log(1-p)\}, & \text{for } \xi = 0. \end{cases} \tag{6}$$

Here p is the probability satisfying $G(x_p) = 1 - p$ (where $G(x)$ is defined in equation (5)). This has the interpretation of x_p being the *return level* associated

with return period $1/p$. For example, for annual maxima data, it is the level which is expected to be exceeded on average once every $1/p$ years.

In `texmex`, return levels are estimated for the GEV as we showed for the GPD:

```
portRL <- predict(port,M=seq(50,1000,by=50),ci.fit=TRUE)
g14 <- ggplot(portRL)
g14[[1]] + scale_x_continuous(trans="log",breaks=c(50,100,200,500,1000))
```




```
portRL$obj[c(1,10,20)]

## $M.50
##           RL    2.5%    97.5%
## mu 4.576567 4.34373 4.809405
##
## $M.500
```

```
##          RL     2.5%    97.5%
## mu 4.932018 4.393014 5.471022
##
## $M.1000
##          RL     2.5%  97.5%
## mu 5.030884 4.376568 5.6852
```

References

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

[2] A. C. Davison and R. L. Smith. Models for exceedances over high thresholds. *Journal of the Royal Statistical Society, Series B*, 53:393 – 442, 1990.

[3] J. E. Heffernan and J. Tawn. A conditional approach for multivariate extreme values. *Journal of the Royal Statistical Society Series B*, 56:497 – 546, 2004.

[4] 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.