A User's Guide to the evdbayes Package (Version 1.0)

Alec Stephenson

Department of Statistics,
Division of Economic and Financial Studies,
Macquarie University, NSW 2109,
Australia.

E-mail: alec_stephenson@hotmail.com
25th May 2004

1 Introduction

1.1 What is the evdbayes package?

The **evdbayes** package is an add-on package for the R (Ihaka and Gentleman, 1996) statistical computing system. It provides functions for the Bayesian analysis of extreme value models, using MCMC methods. There is no direct relationship between the **evd** (Stephenson, 2002) and **evdbayes** packages, but **evd** may be a "required package" for **evdbayes** in the future.

All comments, criticisms and queries on the package or associated documentation are gratefully received.

1.2 Obtaining the package/guide

The package can be downloaded from CRAN (The Comprehensive R Archive Network) at `http://cran.r-project.org/`. This guide (in pdf) will be in the directory `evdbayes/doc/` underneath wherever the package is installed.

1.3 Contents

This guide contains examples on the use of the **evdbayes** package. Section 2 introduces Bayes Theory. Section 3 describes the generalized extreme value distribution and the point process characterization of extremes. Prior distributions are constructed in Section 4. Posterior distributions and MCMC methods are discussed in Section 5. The heart of the guide is contained in Sections 6 and 7. In Section 6 the functions that implement the ideas of Sections 2 to 5 are introduced. Section 7 provides an introduction to more specialist topics.

1.4 Citing the package/guide

To cite this guide or the package in publications please use the following bibliographic database entry.

```
@Manual{key,
  title = {A User's Guide to the evdbayes Package (Version 1.0)},
  author = {Stephenson, A. G.},
  year = {2004},
  month = {May},
  url = {http://cran.r-project.org/}
}
```

1.5 Caveat

I have checked these functions as best I can but, as ever, they may contain bugs. If you find a bug or suspected bug in the code or the documentation please report it to me at `alec_stephenson@hotmail.com`. Please include an appropriate subject line.

1.6 Legalese

1.7 Acknowledgments

Thanks to Ole Christensen and Jonathan Tawn for their comments.

2 An Introduction to Bayes Theory

Let us assume that the data $\boldsymbol{x} = (x_1, \ldots, x_n)$ are independent realizations of a random variable whose density falls within the parametric family $\{f(x|\theta) : \theta \in \Theta\}$. The **likelihood** function is defined using

$$L(\theta; \boldsymbol{x}) = \prod_{i=1}^{n} f(x_i|\theta).$$

It is often easier to work with the **log-likelihood** function $l(\theta; \boldsymbol{x}) = \log\{L(\theta; \boldsymbol{x})\}$. The maximum likelihood estimate $\hat{\theta}(\boldsymbol{x})$ is the value at which $l(\theta; \boldsymbol{x})$ attains its maximum, as a function of θ.

In Bayes Theory we assume that, without reference to the data, it is possible to formulate beliefs about θ that can be expressed as a probability distribution. For example, if $\theta \in (0, 1)$, and you believe that any value in $(0, 1)$ is equally likely, your belief can be expressed using the probability distribution $\theta \sim U(0, 1)$. On the other hand, if $\theta \in \mathbb{R}^3$, you may be able to express your beliefs using a trivariate normal distribution. This requires the specification of nine parameters; the mean and variance of each marginal distribution, and the correlation coefficients between each pair. A distribution on θ, made without reference to the data, is called a **prior distribution**. The parameters of the prior distribution are called **hyperparameters**. The specification of a particular prior distribution requires the specification of all hyperparameters.

Let $\pi(\theta)$ denote the density of the prior distribution for θ. Bayes' theorem states that

$$\pi(\theta|\boldsymbol{x}) = \frac{\pi(\theta)L(\theta; \boldsymbol{x})}{\int_\Theta \pi(\theta)L(\theta; \boldsymbol{x})\,\mathrm{d}\theta} \propto \pi(\theta)L(\theta; \boldsymbol{x}), \tag{1}$$

where $\pi(\theta|\boldsymbol{x})$ is the density of the **posterior distribution**. The posterior distribution includes the additional information provided by the data $\boldsymbol{x}$. Point estimators can be derived by taking e.g. the mean of the posterior distribution (the posterior mean).

Computation of the normalizing constant $\int_\Theta \pi(\theta)L(\theta;\boldsymbol{x})\,\mathrm{d}\theta$ in (1) can be problematic, particularly for high-dimensional θ. Simulation methods can bypass this difficulty. In particular, Markov Chain Monte Carlo (MCMC) techniques seek to produce stationary sequences of simulated (vector) values with marginal density $\pi(\theta|\boldsymbol{x})$. These sequences can then be used to estimate features of the posterior distribution.

3 Likelihoods for Extremes

3.1 Generalized Extreme Value Distributions

The GEV (generalized extreme value) distribution function is given by

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

where (μ,σ,ξ) are the location, scale and shape parameters respectively, $\sigma > 0$ and $h_+ = \max(h,0)$. The case $\xi = 0$ (the Gumbel distribution) is defined by continuity.

Let $\boldsymbol{\theta} = (\mu,\sigma,\xi)$. If we assume that the data $\boldsymbol{x} = (x_1,\ldots,x_n)$ are independent realizations of a random variable distributed as GEV($\boldsymbol{\theta}$), the log-likelihood is

$$l(\boldsymbol{\theta};\boldsymbol{x}) = -n\log\sigma - (1+1/\xi)\sum_{i=1}^{n}\log\{1+\xi\left(x_i-\mu\right)/\sigma\} - \sum_{i=1}^{n}\left\{1+\xi\left(x_i-\mu\right)/\sigma\right\}^{-1/\xi}, \tag{3}$$

provided that $1+\xi\left(x_i-\mu\right)/\sigma$ is positive for each $i = 1,\ldots,n$. If any of these terms are non-positive the likelihood is zero (since the observed data falls beyond the end point of the GEV($\boldsymbol{\theta}$) distribution) and the log-likelihood is $-\infty$. The case $\xi = 0$ is again defined by continuity.

Due to an asymptotic argument (e.g. Coles, 2001) this model is often used when the data $\boldsymbol{x}$ consists of maxima (or negated minima) from some underlying process. Annual sea level maxima and annual temperature maxima are used in the examples of Sections 6.1 and 6.2 respectively.

3.2 Point Process Characterization

Let $X_1,\ldots,X_n$ be a series of independent random variables with common distribution function F. Suppose that n is large, so that the distribution of $M_n = \max\{X_1,\ldots,X_n\}$ can be approximated by the GEV(μ,σ,ξ) distribution (e.g. Coles, 2001), with (possibly infinite) end points* z_- and z_+. Then for large thresholds $u > z_-$ the sequence $\{X_1,\ldots,X_n\}$ viewed on the interval (u,z_+) is approximately a non-homogeneous Poisson process with intensity function

$$\lambda_{\boldsymbol{\theta}}(x) = \frac{1}{\sigma}\left\{1+\xi\left(\frac{x-\mu}{\sigma}\right)\right\}^{-(\xi+1)/\xi}, \qquad u < x < z_+,$$

where $\sigma > 0$ and $\boldsymbol{\theta} = (\mu,\sigma,\xi)$. The intensity measure on (u,z_+) is therefore given by

$$\Lambda_{\boldsymbol{\theta}}(u,z_+) = \int_u^{z_+}\lambda_{\boldsymbol{\theta}}(x)\mathrm{d}x = \left\{1+\xi\left(\frac{u-\mu}{\sigma}\right)\right\}^{-1/\xi}.$$

The mathematical details of the asymptotic approximation are given in Pickands (1971) and Smith (1989). The approximation yields a likelihood for $\boldsymbol{\theta}$ based on observed data $\boldsymbol{x}$ =

*If $\xi > 0$, $z_- = \mu - \sigma/\xi$ and $z_+ = \infty$. If $\xi < 0$, $z_- = -\infty$ and $z_+ = \mu - \sigma/\xi$. If $\xi = 0$, the expressions given are all defined by continuity, with $z_- = -\infty$ and $z_+ = \infty$.

$(x_1, \ldots, x_n)$. Suppose that n_u of the n observations exceed the threshold u. Let $x_{(i)}$ denote the ith exceedence, for $i = 1, \ldots, n_u$. The log-likelihood function can be derived (Coles, 2001) as

$$l(\boldsymbol{\theta}; \boldsymbol{x}) = -\Lambda_{\boldsymbol{\theta}}(u, z_+) + \sum_{i=1}^{n_u} \log\{\lambda_{\boldsymbol{\theta}}(x_{(i)})\},$$

provided that $1+\xi(u-\mu)/\sigma$ and $1+\xi(x_{(i)}-\mu)/\sigma$ for $i = 1, \ldots, n_u$ are positive. The interpretation of $\boldsymbol{\theta}$ depends on the value of n, because the approximate distribution of M_n is GEV($\boldsymbol{\theta}$). The following adjustment[†] to the log-likelihood $l(\boldsymbol{\theta}; \boldsymbol{x})$ avoids this problem.

$$l(\boldsymbol{\theta}; \boldsymbol{x}) = -n_y \Lambda_{\boldsymbol{\theta}}(u, z_+) + \sum_{i=1}^{n_u} \log\{\lambda_{\boldsymbol{\theta}}(x_{(i)})\}. \tag{4}$$

If the value n_y is the number of years of observation (excluding missing values), the annual maxima are distributed as GEV($\boldsymbol{\theta}$). More generally, if n_y is the number of periods of observation, the maxima over those periods are distributed as GEV($\boldsymbol{\theta}$). The asymptotic approximation assumes that there are a large number of observations within each period.

4 Construction of Prior Distributions

The likelihoods (3) and (4) are both functions of the parameter vector $\boldsymbol{\theta} = (\mu, \sigma, \xi)$. The construction of a prior distribution on $\boldsymbol{\theta}$ proceeds in the same manner for both models. We employ three different methods of construction. The first method uses the trivariate normal distribution. The second and third methods construct priors on the quantile space, for fixed probabilities, and on the probability space, for fixed quantiles.

The trivariate normal distribution, which contains nine hyperparameters, is very flexible but is difficult to elicit. At the other extreme, the construction on the probability space is relatively easy to elicit but is not very flexible, having only four hyperparameters with which to define a trivariate distribution.

The trivariate normal construction is the only construction of those presented below that enables the specification of independent parameters (e.g. Section 6.1). This specification is often used for a naive analysis, where there is no external information with which to formulate a dependence structure. On the other hand, increasing σ or ξ leads to a heavier tailed distribution, so *a priori* negative dependence between these parameters is expected (Coles and Tawn, 1996). The quantile space and probability space constructions induce a natural dependence structure using only a small number of hyperparameters. They also enable the elicitation of information using familiar quantities (e.g. Section 6.2).

4.1 Trivariate Normal Distribution

A trivariate normal prior distribution on $\boldsymbol{\theta}' = (\mu, \log\sigma, \xi)$ leads to the prior density

$$\pi(\boldsymbol{\theta}) \propto \frac{1}{\sigma} \exp\left\{-\frac{1}{2}(\boldsymbol{\theta}' - \boldsymbol{\nu})^T \Sigma^{-1} (\boldsymbol{\theta}' - \boldsymbol{\nu})\right\}. \tag{5}$$

This approach was used by Coles and Powell (1996). The mean vector $\boldsymbol{\nu}$ and the symmetric positive definite (3×3) covariance matrix Σ must be specified.

[†]The adjustment can be derived by introducing the multiplicative factor n_y to the intensity function.

4.2 Gamma Distributions for Quantile Differences

The following approach was used in Coles and Tawn (1996). Let $F(q_p) = 1 - p$, where $F(\cdot)$ is the GEV distribution function, given in expression (2). It follows that

$$q_p = \mu + \sigma(x_p^{-\xi} - 1)/\xi,$$

where $x_p = -\log(1-p)$. A prior distribution can be constructed in terms of the quantiles $(q_{p_1}, q_{p_2}, q_{p_3})$ for specified probabilities $p_1 > p_2 > p_3$. Since $q_{p_1} < q_{p_2} < q_{p_3}$ it is easier to work with the differences $(\tilde{q}_{p_1}, \tilde{q}_{p_2}, \tilde{q}_{p_3})$, so that $\tilde{q}_{p_i} = q_{p_i} - q_{p_{i-1}}$ for $i = 1, 2, 3$, where q_{p_0} is the physical lower end point of the process variable. The measurement scale can always be transformed to make the lower end point zero. The **evdbayes** package therefore assumes that $q_{p_0} = 0$. The priors on the quantile differences are taken to be independent, with

$$\tilde{q}_{p_i} \sim \text{gamma}(\alpha_i, \beta_i), \qquad \alpha_i, \beta_i > 0,$$

for $i = 1, 2, 3$. The differences $(\tilde{q}_{p_2}, \tilde{q}_{p_3})$ only depend on the scale and shape parameters (σ, ξ). The prior information on the location parameter μ arises only through $\tilde{q}_{p_1}$. The hyperparameters $(\alpha_1, \alpha_2, \alpha_3)$ and $(\beta_1, \beta_2, \beta_3)$, and the probabilities $p_1 > p_2 > p_3$, must all be specified. (By default the **evdbayes** package uses $p_i = 10^{-i}$ for $i = 1, 2, 3$.) This construction leads to the prior density

$$\pi(\boldsymbol{\theta}) \propto \mathrm{J} \prod_{i=1}^{3} \tilde{q}_{p_i}^{\alpha_i - 1} \exp\{-\tilde{q}_{p_i}/\beta_i\}, \tag{6}$$

provided that $q_{p_1} < q_{p_2} < q_{p_3}$. J is the Jacobian of the transformation from $(q_{p_1}, q_{p_2}, q_{p_3})$ to $\boldsymbol{\theta} = (\mu, \sigma, \xi)$, namely

$$\mathrm{J} = \sigma/\xi^2 \left| \sum_{\substack{i,j \in \{1,2,3\} \\ i<j}} (-1)^{i+j} (x_i x_j)^{-\xi} \log(x_j/x_i) \right|,$$

where $x_i = -\log(1 - p_i)$ for $i = 1, 2, 3$.

At $\xi = 0$ the prior distribution is defined by continuity, using

$$\lim_{\xi \to 0} q_{p_i} = \mu - \sigma \log x_i, \qquad i = 1, 2, 3,$$

and

$$\lim_{\xi \to 0} \mathrm{J} = \sigma/2 \left| \sum_{\substack{i,j \in \{1,2,3\} \\ i<j}} (-1)^{i+j} \log x_i \log x_j \log(x_j/x_i) \right|.$$

Derivations of the results given in this section are presented in detail in Section 5.6 of Stephenson (2003).

4.3 Beta Distributions for Probability Ratios

The following method of construction was proposed by Crowder (1992). Let $F(q) = 1 - p_q$, where $F(\cdot)$ is the GEV distribution function, given in expression (2). It follows that

$$p_q = 1 - \exp\left\{ -\left[1 + \xi\,(q - \mu)\,/\sigma\right]_+^{-1/\xi} \right\}.$$

A prior distribution can be constructed in terms of the probabilities $(p_{q_1}, p_{q_2}, p_{q_3})$ for specified quantiles $q_1 < q_2 < q_3$. Define $p_{q_0} = 1$ and $p_{q_4} = 0$. Since $p_{q_1} > p_{q_2} > p_{q_3}$ it is easier to work with the ratios $(\tilde{p}_{q_1}, \tilde{p}_{q_2}, \tilde{p}_{q_3})$, where $\tilde{p}_{q_i} = p_{q_i}/p_{q_{i-1}}$ for $i = 1, 2, 3$.

The priors on the probability ratios are then taken to be independent, with

$$\tilde{p}_{q_i} \sim \text{beta}\left(\sum_{j=i+1}^{4} \alpha_j,\, \alpha_i\right), \qquad i = 1, 2, 3.$$

The positive hyperparameters $(\alpha_i, \alpha_2, \alpha_3, \alpha_4)$ and the quantiles $q_1 < q_2 < q_3$ must all be specified. This construction leads to the prior density

$$\pi(\boldsymbol{\theta}) \propto \mathrm{J} \prod_{i=1}^{4} (p_{q_{i-1}} - p_{q_i})^{\alpha_i - 1}, \tag{7}$$

provided that $p_{q_1} > p_{q_2} > p_{q_3}$ and that $1 + \xi(q_i - \mu)/\sigma$ is positive for each $i = 1, 2, 3$. J is the Jacobian of the transformation from $(p_{q_1}, p_{q_2}, p_{q_3})$ to $\boldsymbol{\theta} = (\mu, \sigma, \xi)$, namely

$$\mathrm{J} = \sigma/\xi^2 \left\{ \prod_{i=1}^{3} f(q_i) \right\} \left| \sum_{\substack{i,j \in \{1,2,3\} \\ i<j}} (-1)^{i+j} (x_i x_j)^{-\xi} \log(x_j/x_i) \right|,$$

where $x_i = -\log(1 - p_{q_i})$ for $i = 1, 2, 3$, and $f(\cdot)$ is the density of the generalized extreme value distribution, so that $f(q_i) = x_i^{1+\xi} \mathrm{e}^{-x_i}/\sigma$.

Define $x_{i_0} = \lim_{\xi \to 0} x_i = \exp\{-(q_i - \mu)/\sigma\}$, for $i = 1, 2, 3$. At $\xi = 0$ the prior distribution is defined by continuity, using

$$\lim_{\xi \to 0} p_{q_i} = 1 - \mathrm{e}^{-x_{i_0}}, \qquad i = 1, 2, 3,$$

and

$$\begin{aligned} \lim_{\xi \to 0} \mathrm{J} &= \sigma/2 \left\{ \prod_{i=1}^{3} f_0(q_i) \right\} \left| \sum_{\substack{i,j \in \{1,2,3\} \\ i<j}} (-1)^{i+j} \log x_{i_0} \log x_{j_0} \log(x_{j_0}/x_{i_0}) \right| \\ &= \frac{1}{2\sigma^2} \left\{ \prod_{i=1}^{3} f_0(q_i) \right\} \left| \sum_{\substack{i,j \in \{1,2,3\} \\ i<j}} (-1)^{i+j} q_i q_j (q_i - q_j) \right|, \end{aligned}$$

where $f_0(\cdot)$ is the density of the Gumbel distribution, so that $f_0(q_i) = x_{i_0} \mathrm{e}^{-x_{i_0}}/\sigma$. Derivations of the results given in this section are presented in detail in Section 5.6 of Stephenson (2003).

5 Posterior Distributions

Given our prior density $\pi(\boldsymbol{\theta})$ and our likelihood $L(\boldsymbol{\theta}; \boldsymbol{x})$ the posterior density $\pi(\boldsymbol{\theta}|\boldsymbol{x})$ is defined by equation (1). Computing $\pi(\boldsymbol{\theta}|\boldsymbol{x})$ directly is problematic because it requires the computation of the integral $\int_\Theta \pi(\boldsymbol{\theta}) L(\boldsymbol{\theta}|\boldsymbol{x})\, \mathrm{d}\boldsymbol{\theta}$. Markov Chain Monte Carlo (MCMC) techniques can bypass this difficulty.

The **evdbayes** package produces a **Markov chain*** $\boldsymbol{\theta}_0, \ldots, \boldsymbol{\theta}_n$ with **equilibrium** distribution† $\pi(\boldsymbol{\theta}|\boldsymbol{x})$. Loosely speaking, this means that after the chain has been run for a certain length of time each subsequent sample within the chain will be (approximately) distributed as $\pi(\boldsymbol{\theta}|\boldsymbol{x})$,

*Loosely speaking, a (discrete-time) Markov chain is a stochastic process unfolding in time so that the past and future states are independent given the present.

†The equilibrium distribution is the distribution with density $\pi(\boldsymbol{\theta}|\boldsymbol{x})$, not $\pi(\boldsymbol{\theta}|\boldsymbol{x})$ itself. We will often refer to a distribution using the corresponding density function.

though the samples will not be independent. In this context $\pi(\boldsymbol{\theta}|\boldsymbol{x})$ is known as the **target** distribution of the Markov chain. The user must specify the **run length** n, and the initial value $\boldsymbol{\theta}_0 = (\mu_0, \sigma_0, \xi_0)$. After the Markov chain has been generated, the user must also decide when equilibrium has been reached by specifying the **burn-in** period b. The first b samples (including the initial value) are then discarded from the chain. Features of the posterior distribution are estimated using $\boldsymbol{\theta}_b, \ldots, \boldsymbol{\theta}_n$, which we assume to be a stationary sequence of (vector) values with marginal density $\pi(\boldsymbol{\theta}|\boldsymbol{x})$. For example, $\frac{1}{n-b+1}\sum_{t=b}^{n} \mu_t$ is a consistent (as $n \to \infty$) estimate of the posterior mean of μ. The dependence between the samples $\boldsymbol{\theta}_b, \ldots, \boldsymbol{\theta}_n$ influences the accuracy of these estimates. As the dependence becomes stronger, the run length n must be larger in order to achieve the same precision. Dependence exists both within the output for a single parameter (**autocorrelations**) and across parameters (**cross-correlations**).

Suppose the initial value of the chain is specified as $\boldsymbol{\theta}_0$. Given that the chain is at state $\boldsymbol{\theta}_t = (\mu_t, \sigma_t, \xi_t)$ at iteration t, the subsequent state $\boldsymbol{\theta}_{t+1}$ is generated using the following algorithm. $LN(\nu, \gamma^2)$ denotes the log-normal distribution, with mean $\exp(\nu + \gamma^2/2)$ and variance $\exp(2\nu + 2\gamma^2) - \exp(2\nu + \gamma^2)$, so that X is distributed as $LN(\nu, \gamma^2)$ if and only if the logarithm of X is distributed as $N(\nu, \gamma^2)$. The positive values $\boldsymbol{s} = (s_\mu, s_\sigma, s_\xi)$ should be specified to ensure that the chain has desirable properties. They should be large enough to ensure that the proposals are made throughout the sample space, but small enough to ensure that the proposed values are accepted often (e.g. Section 6.1).

Propose $\mu^* \sim N(\mu_t, s_\mu^2)$.
Set $\Delta = \frac{\pi(\mu^*, \sigma_t, \xi_t|\boldsymbol{x})}{\pi(\mu_t, \sigma_t, \xi_t|\boldsymbol{x})}$.
Set $\mu_{t+1} = \mu^*$ with probability $\min\{1, \Delta\}$, else set $\mu_{t+1} = \mu_t$.

Propose $\sigma^* \sim LN(\log \sigma_t, s_\sigma^2)$.
Set $\Delta = \frac{\pi(\mu_{t+1}, \sigma^*, \xi_t|\boldsymbol{x})}{\pi(\mu_{t+1}, \sigma_t, \xi_t|\boldsymbol{x})} \frac{\sigma^*}{\sigma_t}$.
Set $\sigma_{t+1} = \sigma^*$ with probability $\min\{1, \Delta\}$, else set $\sigma_{t+1} = \sigma_t$.

Propose $\xi^* \sim N(\xi_t, s_\xi^2)$.
Set $\Delta = \frac{\pi(\mu_{t+1}, \sigma_{t+1}, \xi^*|\boldsymbol{x})}{\pi(\mu_{t+1}, \sigma_{t+1}, \xi_t|\boldsymbol{x})}$.
Set $\xi_{t+1} = \xi^*$ with probability $\min\{1, \Delta\}$, else set $\xi_{t+1} = \xi_t$.

6 Examples

There are five main functions in the **evdbayes** package. The functions `prior.norm`, `prior.quant` and `prior.prob` construct the prior distributions presented in Section 4. The function `posterior` generates a Markov chain $\boldsymbol{\theta}_0, \ldots, \boldsymbol{\theta}_n$ with target distribution $\pi(\boldsymbol{\theta}|\boldsymbol{x})$. The function `mposterior` (locally) maximizes $\pi(\boldsymbol{\theta}|\boldsymbol{x})$, as a function of $\boldsymbol{\theta}$. This may be used to specify the initial value $\boldsymbol{\theta}_0 = \arg\max_{\boldsymbol{\theta}} \pi(\boldsymbol{\theta}|\boldsymbol{x})$.

This section presents three examples that illustrate these functions. The first and second examples use the generalized extreme value model of Section 3.1. The first example replicates the Bayesian analysis of sea level maxima from Section 9.1.3 of Coles (2001). The second example examines annual maximum temperatures recorded at Oxford, England. The third example uses the point process characterization of Section 3.2 for daily rainfall observations, following Coles and Tawn (1994). The datasets used in the first and second examples are available in the **evd** package (Stephenson, 2002).

The computations in the following sections were performed using a notebook containing a 1.2GHz Celeron processor and 256MB RAM. The generation times of each Markov chain are given in square brackets. The slowest generation time of all the chains generated within this section is about two seconds per 1000 iterations.

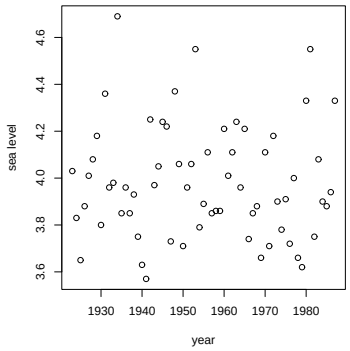

Figure 1: Annual maximum sea levels at Port Pirie, South Australia.

6.1 Port Pirie Sea Level Data

The numeric vector `portpirie` contains annual maximum sea levels (in metres) recorded at Port Pirie, South Australia, from 1923 to 1987. It is included in the **evd** package, and can be made available using `data(portpirie)`. The data are plotted in Figure 1, which can be reproduced using the code given below.

```
> data(portpirie) ; ptp <- portpirie
> plot(1923:1987, ptp, xlab = "year", ylab = "sea level")
```

A "naive Bayesian analysis" of the Port Pirie data is performed in Coles (2001, Section 9.1.3). He uses the word 'naive' because he has no external information with which to formulate a prior distribution and he makes little attempt to ensure that the generated Markov chain has desirable properties. I will begin by replicating his analysis. I will then examine the generated Markov chain and repeat the analysis to ensure that the generated chain does have desirable properties.

The prior specified by Coles (2001) can be constructed using the following code. (The function `diag` creates a diagonal matrix.)

```
> mat <- diag(c(10000, 10000, 100))
> pn <- prior.norm(mean = c(0,0,0), cov = mat)
```

The function `prior.norm` is used to construct a multivariate normal prior distribution on $(\mu, \log\sigma, \xi)$, with density (5). The off-diagonal elements of the covariance matrix of this distribution are zero, so the parameters are specified to be independent. The prior is therefore defined by the marginal distributions $\mu \sim N(0, 10^4)$, $\sigma \sim LN(0, 10^4)$ and $\xi \sim N(0, 10^2)$. The high variances lead to near-flat marginal priors, which reflect the absence of external information.

The function `posterior` can now be used to generate a Markov chain $\boldsymbol{\theta}_0, \ldots, \boldsymbol{\theta}_n$ with target distribution $\pi(\boldsymbol{\theta}|\boldsymbol{x})$. Coles (2001) generates a chain of length 1000, using the initial values $\boldsymbol{\theta}_0 = (5, 1, 0.1)$ and the proposal standard deviations* $\boldsymbol{s} = (0.02, 0.1, 0.1)$. This chain can be reproduced using the following assignment [1.5 secs].

```
> n <- 1000 ; t0 <- c(5,1,0.1) ; s <- c(.02,.1,.1)
> ptpmc <- posterior(n, t0, prior = pn, lh = "gev", data = ptp, psd = s)
```

*Strictly speaking, s_σ is the standard deviation of the proposal distribution for $\log\sigma$, not for σ.

The data `ptp` consist of annual maxima, so the generalized extreme value likelihood of Section 3.1 is specified using `lh = "gev"`. The first argument to `posterior` is the run length. The second argument is the initial value $\boldsymbol{\theta}_0$, and the proposal standard deviations $\boldsymbol{s}$ should be passed to `psd`. The prior distribution which we constructed earlier using `prior.norm` is passed to the argument `prior`.

The object `ptpmc` is a matrix with 1001 rows and 3 columns, containing the Markov chain. The rows and columns are labelled using iteration numbers and parameter names respectively. The object also contains an attribute named `ar`, which is a matrix containing information regarding the acceptance of proposed values within the MCMC algorithm. The `ar` attribute is shown below. It can be printed using `attributes(ptpmc)$ar`.

```
            mu sigma   xi total
acc.rates 0.75  0.65 0.68  0.69
ext.rates 0.00  0.00 0.03  0.01
```

The first row contains acceptance rates (i.e. the number of times a proposal was accepted as a fraction of the run length) for each parameter and for the entire chain. If the acceptance rates are too low there may be substantial periods during which the chain does not move at all, because proposals are made that are too far away from the current state. If the acceptance rates are too high the chain may be exploring only a small fraction of the parameter space, because proposals are made that are too close to the current state. If the chain is jumping around, exploring all of the parameter space, we say that it is **mixing well**.

The proposal standard deviations $\boldsymbol{s} = (s_\mu, s_\sigma, s_\xi)$ can be used to tune the acceptance rates. Higher standard deviations give smaller acceptance rates, and vice-versa. It is difficult to give general advice on which acceptance rates represent the ideal, because results exist only for particular classes of target and proposal distributions (e.g. Gelman *et al.*, 1995). The behaviour of a chain for any given value of $\boldsymbol{s}$ can always be determined by plotting the sampled values. Acceptance rates of about 40 percent should lead to chains that mix well. The acceptance rates for this chain are quite large. If the proposal standard deviations were a bit higher, the chain would have better mixing properties.

The second row of the `ar` attribute contains the number of times a proposal was made for which the posterior density estimate was zero, as a fraction of the run length. This occurs when the upper/lower end point of the generalized extreme value distribution is less/greater than the largest/smallest data point. If these values are high, either the proposal standard deviations are too large, or the density of the target distribution is large near the boundary of the parameter space.

Once a Markov chain has been generated it needs to be analysed to ensure that it has desirable properties. I recommend that the R package **coda** is installed for this purpose. This package includes the function `mcmc`, which creates an `mcmc` object that **coda** can recognize as a Markov chain. The iterations of the chain, shown in Figure 2, can be plotted using the following snippet.

```
> ptp.mcmc <- mcmc(ptpmc, start = 0, end = 1000)
> plot(ptp.mcmc, den = FALSE, sm = FALSE)
```

Figure 2 recreates Figure 9.1 from Coles (2001). The two figures exhibit similar behaviour, though Coles (2001) plots the iterations of $\log \sigma$ rather than σ. Other differences are due to sampling variability.

The burn-in period seems to take about $b = 300$ iterations. Thereafter, the stochastic variations in the chain seem reasonably homogeneous. The starting value $\boldsymbol{\theta}_0 = (5, 1, 0.1)$ is relatively poor[†],

[†]I imagine that Coles (2001) deliberately selected a poor starting value, so that the burn-in period would be clearly depicted within Figure 2.

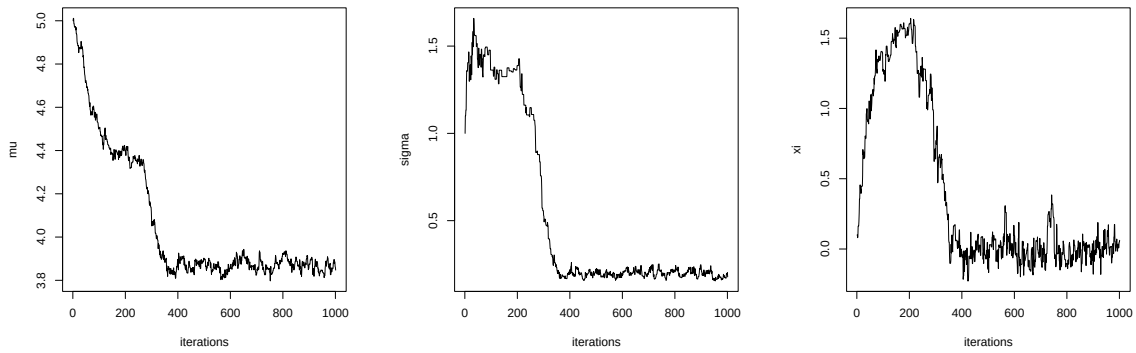

Figure 2: MCMC realizations of generalized extreme value parameters in a Bayesian analysis of the Port Pirie data, recreating Figure 9.1 from Coles (2001).

as it is not close to the centre of the posterior distribution. A good starting value $\boldsymbol{\theta_0}$ can be derived using the function `mposterior`, which (locally) maximizes $\pi(\boldsymbol{\theta}|\boldsymbol{x})$. The maximization is performed in the following snippet.

```
> maxpst <- mposterior(t0, prior = pn, lh = "gev", data = ptp)
> round(maxpst$par, 2)
[1]  3.87  0.20 -0.05
```

The arguments of `mposterior` are the same as those of `posterior`, except that the first argument is now the initial value for the optimization. The value returned from `mposterior` is a list of the same form as the value returned by the optimization function `optim`. The component `par` within the returned list contains $\arg\max_{\boldsymbol{\theta}} \pi(\boldsymbol{\theta}|\boldsymbol{x})$, which can be used as the initial value of the Markov chain. The above snippet suggests that we take $\boldsymbol{\theta_0} = (3.87, 0.2, -0.05)$.

In this example the prior densities are near-flat, reflecting the absence of prior information. It is therefore approximately true that $\pi(\boldsymbol{\theta}|\boldsymbol{x}) \propto L(\boldsymbol{\theta}; \boldsymbol{x})$, so we should expect $(3.87, 0.2, -0.05)$ to be close to maximum likelihood estimates. (In fact, they are the same when rounded to the second decimal place.) Maximum likelihood estimates[‡] often serve as good starting values.

An alternative approach is to generate a (short) Markov chain and examine the output to see where the posterior density is large. For example, the iterations shown in Figure 2 suggest taking $\boldsymbol{\theta_0} \approx (3.9, 0.2, 0)$. The initial value can then be used to generate a further (longer) chain. This is essentially the same as maximizing $\pi(\boldsymbol{\theta}|\boldsymbol{x})$ using a stochastic optimization routine, such as **simulated annealing**. Simulated annealing can be used to maximize $\pi(\boldsymbol{\theta}|\boldsymbol{x})$ by including the argument `method = "SANN"` in the call to `mposterior`.

Another approach entirely is to take multiple initial values, scattered about the parameter space. This generates multiple Markov chains which, loosely speaking, can be compared to see if they eventually produce the same behaviour (Gelman and Rubin, 1992). Multiple chains are discussed further in Section 7.1.2.

The code below generates the Markov chain again [1.5 secs] using the starting value $\boldsymbol{\theta_0} = (3.87, 0.2, -0.05)$. After some pilot runs, I have decided to take $\boldsymbol{s} = (.06, .25, .25)$.

[‡]The functions `fgev` and `fpot` in the **evd** package can calculate maximum likelihood estimates for the models of Section 3.

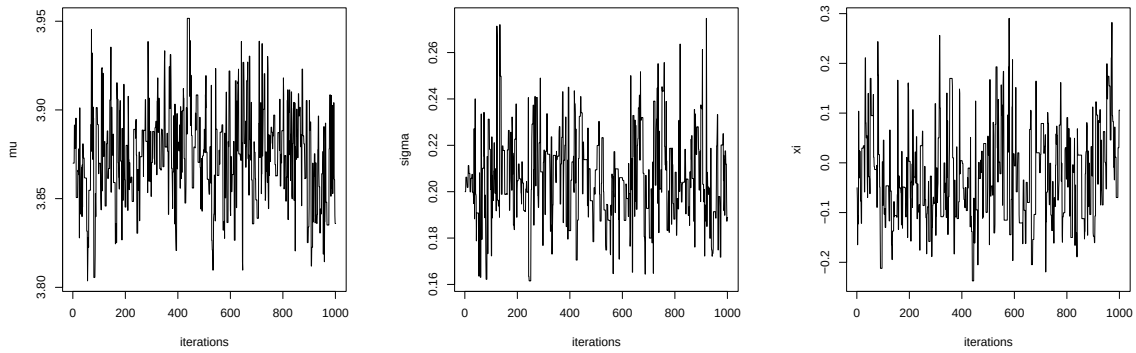

Figure 3: MCMC realizations of generalized extreme value parameters in a Bayesian analysis of the Port Pirie data, using different starting values and proposal standard deviations than those used to produce the realizations of Figure 2.

```
> t0 <- c(3.87,0.2,-0.05) ; s <- c(.06,.25,.25)
> ptpmc <- posterior(n, t0, prior = pn, lh = "gev", data = ptp, psd = s)
> ptp.mcmc <- mcmc(ptpmc, start = 0, end = 1000)
> plot(ptp.mcmc, den = FALSE, sm = FALSE)
```

Figure 3 shows the iterations of the chain. The starting value $\boldsymbol{\theta_0} = (3.87, 0.2, -0.05)$ yields a smaller burn-in period. The proposal standard deviations $\boldsymbol{s} = (.06, .25, .25)$ lead to improved mixing properties (although this is difficult to determine from Figures 2 and 3 because of the different scales on the y-axis).

The properties of the chains produced by `posterior` can be examined using statistical techniques. These techniques attempt to assess whether the chain is in equilibrium (or equivalently, whether the burn-in period is sufficiently long). There are also techniques that determine how long the chain should be in order to achieve a given aim. Reviews of these techniques are given in Cowles and Carlin (1996) and Brooks and Roberts (1998). The **coda** package contains functions that implement various diagnostics. The following paragraphs demonstrate the diagnostics introduced by Geweke (1992) and Raftery and Lewis (1992). The diagnostic of Gelman and Rubin (1992), designed for multiple chains, is illustrated in Section 7.1.2.

The diagnostic of Geweke (1992) is particularly simple. For each parameter, the means of the first and last parts of the chain are tested for equality. By default, the first 10% and the last 50% are used. The difference between the two means is divided by its estimated standard error. The estimation of the standard error attempts to take into account the autocorrelations. If the chain has reached equilibrium the distribution of each statistic is approximately standard normal. The code below implements this diagnostic on the Markov chain generated previously, where the first $b = 200$ samples are treated as the burn-in period and are discarded using the **coda** function `window`.

```
> ptp.mcmc <- window(ptp.mcmc, start = 200)
> geweke.diag(ptp.mcmc)

Fraction in 1st window = 0.1
Fraction in 2nd window = 0.5

     mu   sigma      xi
```

```
 0.4131 -0.0289 -0.3141

> geweke.diag(ptp.mcmc, 0.2, 0.4)

Fraction in 1st window = 0.2
Fraction in 2nd window = 0.4

     mu   sigma      xi
 0.1067 -0.6257  0.3706
```

The test statistics do not give any cause for concern. If any of the values are above two in absolute value, you may wish to increase the burn-in period and repeat the test. This is the basis for the plot produced by the function `geweke.plot`.

The diagnostic of Raftery and Lewis (1992) attempts to assess how long the chain should be in order to achieve a given aim. Specifically, it gives the number of samples that are needed to estimate a quantile (on each margin) within a certain accuracy with at least probability `s`. It is intended for use on short pilot runs. By default the quantile corresponds to the $\mathtt{q} = 0.025$ point of the distribution function and the probability $\mathtt{s} = 0.95$. The accuracy is defined so that the area to the left of the specified quantile be within a given margin $\pm\mathtt{r}$ of `q`. By default, $\mathtt{r} = 0.005$.

```
> raftery.diag(ptp.mcmc, r = 0.01, s = 0.75)

Quantile (q) = 0.025
Accuracy (r) = +/- 0.01
Probability (s) = 0.75

       Burn-in  Total Lower bound  Dependence
       (M)      (N)   (Nmin)       factor (I)
 mu    10       1033  323          3.20
 sigma 13       1342  323          4.15
 xi    12       1211  323          3.75
```

The first column gives the additional burn-in that would be useful next time you run the chain. The recommendations are often small, and appear to be of limited use in practice. The second column is of greatest interest. It specifies the length of chain (including the additional recommended burn-in, but excluding the 200 iterations already discarded) that is needed to achieve the designated aim, for each parameter. The third column gives number of samples `Nmin` that would be required if those samples were independent. If this number is greater than the length of the chain being analysed, which in this case is 801, the function simply returns a sentence stating the value of `Nmin`. (If the default arguments are used, $\mathtt{Nmin} = 3746$.) The final column gives the dependence factor, which is the ratio of the two preceding columns. The factor represents the extent to which the autocorrelation inflates the required sample size. Autocorrelations can be estimated and plotted using `autocorr` and `autocorr.plot`. (There exists similar functions for cross-correlations.) Large dependence factors occur when strong autocorrelations are present.

The diagnostics within the **coda** package should not be used as a substitute for the graphical examination of the sampled values. If you are going to use these diagnostics you should implement a range of methods, rather than a single test. I also recommend that you take some time to examine the theoretical details of each diagnostic that you implement. It is important to emphasize that there are inherent difficulties with all diagnostic procedures. In particular, no technique can be guaranteed to successfully diagnose convergence. Cowles and Carlin (1996)

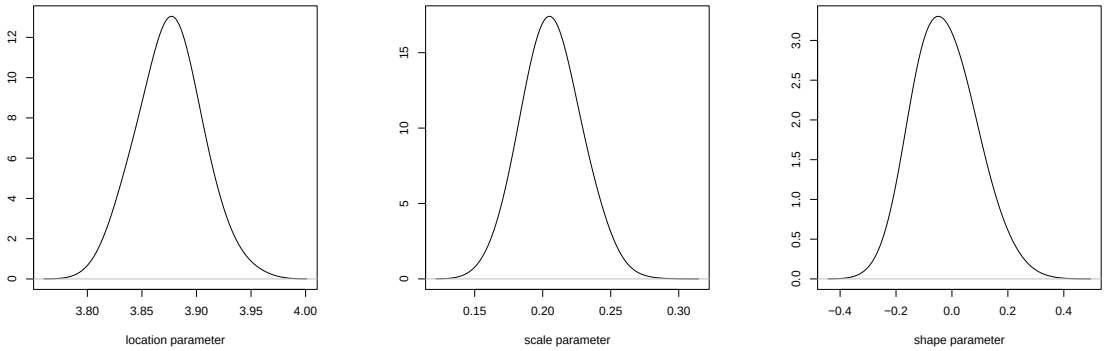

Figure 4: Marginal posterior density estimates for the generalized extreme value parameters μ, σ and ξ respectively, in a Bayesian analysis of the Port Pirie data.

point out that many statisticians rely heavily on such diagnostics, if for no other reason than "a weak diagnostic is better than no diagnostic at all".

The values $\boldsymbol{\theta}_b, \ldots, \boldsymbol{\theta}_n$ (with $b = 200$ and $n = 1000$) contained within the object `ptp.mcmc` can now be treated as (dependent) samples from the posterior distribution, with density $\pi(\boldsymbol{\theta}|\boldsymbol{x})$. In the following code, $\boldsymbol{\theta}_b, \ldots, \boldsymbol{\theta}_n$ are used to estimate features of $\pi(\boldsymbol{\theta}|\boldsymbol{x})$. The marginal density estimates, given in Figure 4, are created using `plot`.

```
> bwf <- function(x) sd(x)/2
> plot(ptp.mcmc, trace = FALSE, bwf = bwf)
> summary(ptp.mcmc)

Iterations = 200:1000
Thinning interval = 1
Number of chains = 1
Sample size per chain = 801

1. Empirical mean and standard deviation for each variable,
   plus standard error of the mean:

          Mean      SD  Naive SE Time-series SE
mu     3.87432 0.02683 0.0009479       0.001679
sigma  0.20347 0.02099 0.0007417       0.001299
xi    -0.02594 0.09790 0.0034590       0.006264

2. Quantiles for each variable:

         2.5%     25%      50%     75%  97.5%
mu     3.8239  3.8562  3.87592 3.88972 3.9281
sigma  0.1624  0.1896  0.20304 0.21612 0.2475
xi    -0.2027 -0.0937 -0.03075 0.03272 0.1982
```

The `summary` function presents summary statistics for each parameter. The first matrix gives empirical means and standard deviations. It also gives two consistent (as $n \to \infty$) estimates of the standard error of the mean. The `Naive SE` is the usual estimate, namely the empirical standard

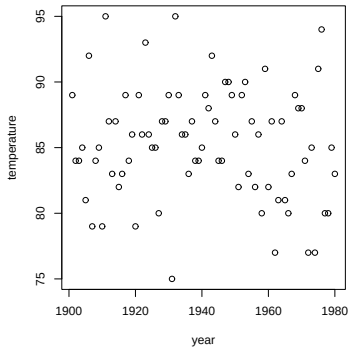

Figure 5: Annual maximum temperatures at Oxford, England.

deviation divided by the square root of the number of iterations. The `Time-series SE` is an estimate that attempts to account for the autocorrelation. The second matrix gives empirical quantiles. The empirical $100\alpha/2$ and $100(1-\alpha/2)$ percent quantiles form **posterior probability intervals**, which contain exactly $100(1-\alpha)\%$ of the posterior probability on each margin. The `summary` function also gives the iteration numbers, the thinning interval (see Section 7.1.1), the number of chains (see Section 7.1.2) and the length of the chain.

6.2 Oxford Temperature Data

The numeric vector `oxford` contains annual maximum temperatures (in degrees Fahrenheit) recorded at Oxford, England, from 1901 to 1980. It is included in the **evd** package, and can be made available using `data(oxford)`. The data are plotted in Figure 5, which can be reproduced using the code given below.

```
> data(oxford) ; ox <- oxford
> plot(1901:1980, ox, xlab = "year", ylab = "temperature")
```

Suppose that we have an expert who is prepared to give us his/her beliefs regarding annual temperature maxima at Oxford, without reference to the data. We will use the construction in Section 4.3, involving prior beta distributions for probability ratios. This induces dependence between the parameters (μ, σ, ξ), and is relatively easy to elicit. On the other hand, it is not very flexible, having only four hyperparameters with which to define a trivariate distribution. The remainder of this section uses the following properties. If $X \sim \text{beta}(a, b)$, with $a, b > 0$, then X has mean $\nu = a/(a+b)$ and variance $\nu(1-\nu)/(a+b+1)$. If $a, b > 1$, the density function has a mode at $(a-1)/(a+b-2)$.

Using the notation of Section 4.3, suppose we take $q_1 = 85$, $q_2 = 88$ and $q_3 = 95$. The corresponding probabilities are denoted by $p_{85} < p_{88} < p_{95}$. The probability ratios are then given by $\tilde{p}_{85} = p_{85}$, $\tilde{p}_{88} = p_{85}/p_{88}$ and $\tilde{p}_{95} = p_{88}/p_{95}$. The prior for $\tilde{p}_{85} = p_{85}$ should be elicited first. In other words, we need to elicit a prior distribution for the probability that the maximum annual temperature at Oxford will exceed 85 degrees Fahrenheit. Suppose that we elicit a beta(5,4) distribution for this probability. This means that we are satisfied that the beliefs of the expert correspond to the properties defined by this distribution. In particular, our expert believes that the maximum annual temperature at Oxford will exceed 85 degrees just over half the time, and he/she is 90% sure that the probability of exceedence is in the interval (0.28, 0.8).

As an aid to the elicitation process I have included a simple function called `ibeta`. This takes the arguments `mean` and `var`, or the arguments `shape1` and `shape2`, all of which can be vectors. It returns a vector or matrix containing the mean (`mean`), variance (`var`), mode (`mode`), and shape parameters (`shape1`/`shape2`) of the beta distribution(s) corresponding to the specified arguments. The code below gives two examples of its use.

```
> xx <- ibeta(shape1 = 5, shape2 = 4)
> round(xx, 2)
shape1 shape2   mean    var   mode
  5.00   4.00   0.56   0.02   0.57

> xx <- ibeta(mean = seq(0.1,0.9,0.2), var = 0.03)
> round(xx, 2)
  shape1 shape2 mean  var mode
1   0.20   1.80  0.1 0.03   NA
2   1.80   4.20  0.3 0.03  0.2
3   3.67   3.67  0.5 0.03  0.5
4   4.20   1.80  0.7 0.03  0.8
5   1.80   0.20  0.9 0.03   NA
```

The first example shows that the mean and variance of a beta(5,4) random variable are 0.56 and 0.02 respectively. The density function has a mode at 0.57. Quantiles and probabilities of beta distributions can be calculated using `qbeta` and `dbeta`. Densities can be calculated (and hence plotted) using `dbeta`. These tools aid elicitation, and help examine fully the elicited distribution.

We have elicited $\tilde{p}_{85} \sim \text{beta}(5, 4)$, so that $\alpha_1 = 4$ and $\alpha_2 + \alpha_3 + \alpha_4 = 5$. Suppose our expert thinks that half the annual maxima that exceed 85 degrees will also exceed 88 degrees. Furthermore, suppose he/she thinks that one tenth of the annual maxima that exceed 88 degrees will also exceed 95 degrees. We can equate the means of $\tilde{p}_{88} \sim \text{beta}(\alpha_3 + \alpha_4, \alpha_2)$, and $\tilde{p}_{95} \sim \text{beta}(\alpha_4, \alpha_3)$ to these ratios, giving $\alpha_3 + \alpha_4 = 0.5 \times 5 = 2.5$ and $\alpha_4 = 0.1 \times 2.5 = 0.25$. This yields $\boldsymbol{\alpha} = (\alpha_1, \alpha_2, \alpha_3, \alpha_4) = (4, 2.5, 2.25, 0.25)$. The parameter vector $\boldsymbol{\alpha}$ can now be used to construct the prior distribution.

The elicitation process demonstrated above is not only hypothetical, but also over-simplified. Elicitation of prior distributions is a notoriously difficult (and controversial) subject. In particular, you must obtain the expert's opinion of a number of different quantities in order to ensure that his/her beliefs can be represented by a specific distribution.

Given that our expert's opinion can be represented in the form of Section 4.3, with $\boldsymbol{\alpha} = (4, 2.5, 2.25, 0.25)$, the prior can be constructed as follows. The function `prior.prob` constructs a prior distribution with density (7).

```
> prox <- prior.prob(quant = c(85,88,95), alpha = c(4,2.5,2.25,0.25))
```

In the Port Pirie data, the prior distribution on $(\mu, \log\sigma, \xi)$ was taken to be trivariate normal. The prior marginal distributions of μ and ξ were therefore normal, and the marginal distribution of σ was log-normal. The densities of these marginals can easily be calculated (and therefore plotted) using `dnorm` and `dlnorm`. In this example, the prior marginal distributions of (μ, σ, ξ) are difficult to determine, since they involve the integration of expression (7). We can avoid this problem using MCMC methods! A Markov chain is generated as before, but now the target distribution has density $\pi(\boldsymbol{\theta})$, rather than $\pi(\boldsymbol{\theta}|\boldsymbol{x})$. This can be implemented by the function `posterior`, using the argument `lh = "none"`, meaning "likelihood is none".

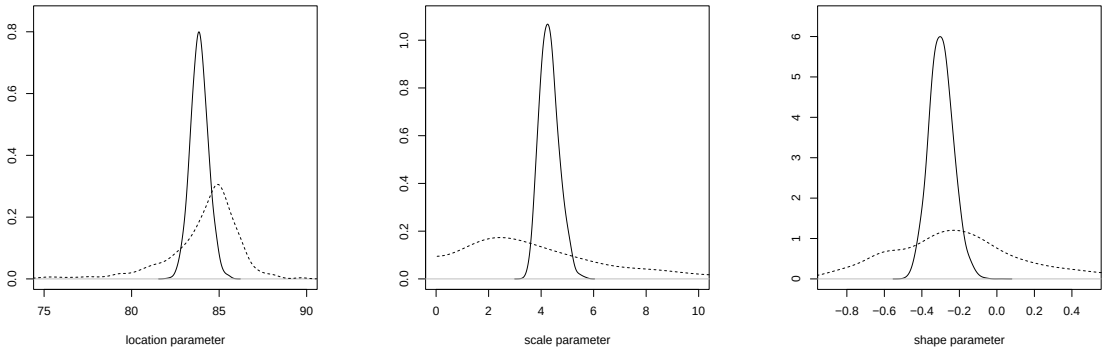

Figure 6: Marginal prior (dashed lines) and posterior (solid lines) density estimates for μ, σ and ξ respectively, in a Bayesian analysis of the Oxford data.

The following code generates two Markov chains using `posterior`. The first chain [45 secs] samples from the prior density $\pi(\boldsymbol{\theta})$. The second chain [18 secs] samples from the posterior density $\pi(\boldsymbol{\theta}|\boldsymbol{x})$. Properties of generated Markov chains were discussed in the previous example, and I will not repeat the process in any detail. The chains have run lengths 50 000 and 10 000 respectively. A larger chain is generated for the prior distribution because the surface is more complex. The proposal standard deviations, initial values and burn-in periods have been determined by pilot runs. The function `posterior` allows the burn-in periods to be specified through the argument `burn`. The first `burn` iterations are discarded from the returned matrix.

```
> n <- 50000 ; t0 <- c(84, 1, 0) ; s <- c(5, 1, .5) ; b <- 5000
> ox.prior <- posterior(n, t0, prox, lh = "none", psd = s, burn = b)
> n <- 10000 ; t0 <- c(84,4.2,-0.3) ; s <- c(1.25,.2,.1) ; b <- 1000
> ox.post <- posterior(n, t0, prox, lh = "gev", data = ox, psd = s, burn = b)
```

Marginal prior and posterior density estimates are given in Figure 6. The figure can be produced using the code given below. The assignment statements within the code prevent the marginal density estimate of the scale parameter from being positive below zero. Density estimates can be plotted more easily using the **coda** package (e.g. Section 6.1). Unfortunately, the tools within the package do not make it any easier to create plots of the same form as Figure 6.

```
> plot(density(ox.post[,1],adj=2), xlim = c(55,90), ylim = c(0,0.85))
> lines(density(ox.prior[,1],adj=2), lty = 2)
> plot(density(ox.post[,2],adj=2), xlim = c(0,10), ylim = c(0,1.05))
> prsc <- density(c(ox.prior[,2], -ox.prior[,2]), adj=2)
> prsc <- list(x = prsc$x[prsc$x > 0], y = 2*prsc$y[prsc$x > 0])
> lines(prsc, lty = 2)
> plot(density(ox.post[,3],adj=2), xlim = c(-0.9,0.5), ylim = c(0,6.5))
> lines(density(ox.prior[,3],adj=2), lty = 2)
```

6.3 Rainfall Data

The numeric vector `rainfall` contains 20820 daily aggregate rainfall observations (in millimetres) recorded at a rain gauge in England over a period of 57 years, beginning on a leap year. Three

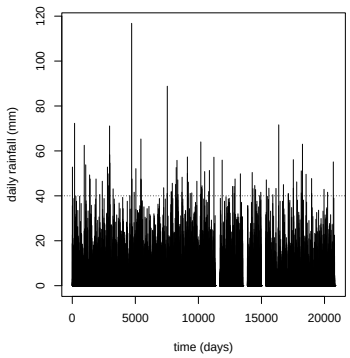

Figure 7: Daily aggregate rainfall recorded at a rain gauge in England over a period of 57 years. The dotted horizontal line represents the threshold used for the likelihood (4).

years contain only missing (`NA`) values, and the remaining 54 years contain 58 missing values in total. The vector is included in the **evdbayes** package, and can be made available using `data(rainfall)`. The data are plotted in Figure 7, which can be reproduced using the code given below.

```
> data(rainfall)
> plot(rainfall, type = "h")
> abline(h = 40, lty = 3)
```

In this section we will suppose that we have used an expert to elicit a prior distribution using the construction of Section 4.2. Suppose that we elicited the distributions $\tilde{q}_{p_1} \sim \text{gamma}(38.9, 1.5)$, $\tilde{q}_{p_2} \sim \text{gamma}(7.1, 6.3)$ and $\tilde{q}_{p_3} \sim \text{gamma}(47, 2.6)$, where $p_i = 10^{-i}$ for $i = 1, 2, 3$.

Quantiles and probabilities of gamma distributions can be calculated using `qgamma` and `pgamma`. Densities can be calculated (and hence plotted) using `dgamma`. The means and variances of the elicited distributions can be derived using `igamma`, as shown below.

```
> igamma(shape = c(38.9,7.1,47), scale = c(1.5,6.3,2.6))
  shape scale   mean     var   mode
1  38.9   1.5  58.35  87.525  56.85
2   7.1   6.3  44.73 281.799  38.43
3  47.0   2.6 122.20 317.720 119.60
```

The prior can be constructed using `prior.quant`. The probabilities do not need to be specified since they are taken as $p_i = 10^{-i}$ by default.

```
> prrain <- prior.quant(shape = c(38.9,7.1,47), scale = c(1.5,6.3,2.6))
```

The generalized extreme value likelihood is only appropriate for maxima. For the rainfall data, we use the point process characterization of Section 3.2. This is specified using `lh = "pp"` in the call to `posterior`. The threshold u within the likelihood (4) is specified using the argument `thresh`. The value n_y is specified using the argument `noy`. If the parameters are to represent the generalized extreme value model for *annual* maxima, n_y should be the number of years of observation (excluding missing values). In this case `noy` ≈ 54.

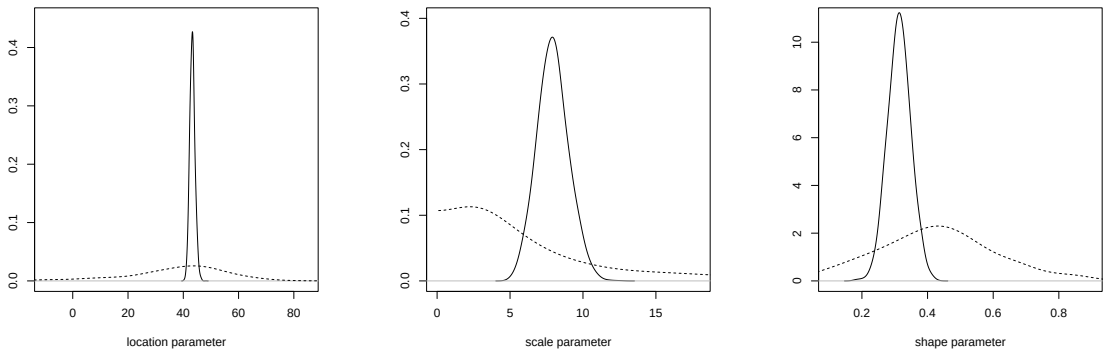

Figure 8: Marginal prior (dashed lines) and posterior (solid lines) density estimates for μ, σ and ξ respectively.

The following code generates two Markov chains using `posterior`. The first chain [10 secs] samples from the prior density $\pi(\boldsymbol{\theta})$. The second chain [20 secs] samples from the posterior density $\pi(\boldsymbol{\theta}|\boldsymbol{x})$. We take the threshold $u = 40$. The specification of the threshold is a standard topic in extreme value theory (e.g. Coles, 2001, Ch 4), and will not be discussed here. Both chains have run length $n = 10\,000$ and burn-in period $b = 2000$. Initial values have been derived using `mposterior`, and the proposal standard deviations $\boldsymbol{s}$ have been determined using pilot runs.

```
> n <- 10000 ; t0 <- c(50.8, 1.18, 0.65) ; s <- c(25, .35, .07) ; b <- 2000
> rn.prior <- posterior(n, t0, prrain, "none", psd = s, burn = b)
> t0 <- c(43.2, 7.64, 0.32) ; s <- c(2, .2, .07)
> rn.post <- posterior(n, t0, prrain, "pp", data = rainfall, thresh = 40,
      noy = 54, psd = s, burn = b)
```

These chains can be used to estimate features of the prior and posterior distributions. Figure 8 shows estimates of the prior and posterior densities. The figure can be constructed by adapting the code, given in Section 6.2, that was used to construct Figure 6.

7 Further Topics

This section provides an introduction to more specialist topics. Section 7.1 introduces the concept of thinning, and discusses the use of multiple Markov chains. Section 7.2 depicts posterior distributions of generalized extreme value quantiles. Predictive distributions are defined and illustrated in Section 7.3. Model diagnostics are implemented in Section 7.4, following Gelman *et al.* (1995). Three different extensions to the likelihoods of Section 3 are discussed in Section 7.5.

7.1 MCMC Topics

7.1.1 Thinning

Suppose that you create a Markov chain, but you only store every kth iteration. This process is called **thinning**. The integer k is called the **thinning interval**. The iterations that have been

stored, after an initial burn-in period, are (on assumption) sampled from the target distribution of the original chain, but the dependence between the samples will have been reduced.

Let us consider a more concrete example. Suppose we take the run length $n = 1000$, the burn-in period $b = 200$ and suppose that we only store every fifth ($k = 5$) iteration. Then we generate the 161 values $\boldsymbol{\theta}_{200}, \boldsymbol{\theta}_{205}, \ldots, \boldsymbol{\theta}_{1000}$. Because the values are only stored after every fifth iteration, they are not as dependent as the 161 values $\boldsymbol{\theta}_{200}, \boldsymbol{\theta}_{201}, \ldots, \boldsymbol{\theta}_{360}$, so they contain more information, and can estimate features of the target distribution more precisely. However, by thinning a chain you *always* loose information, because the 801 values $\boldsymbol{\theta}_{200}, \boldsymbol{\theta}_{201}, \ldots, \boldsymbol{\theta}_{1000}$ are more informative than $\boldsymbol{\theta}_{200}, \boldsymbol{\theta}_{205}, \ldots, \boldsymbol{\theta}_{1000}$. At this point, you may be asking why you would ever want to thin a chain. The main advantage of thinning a chain is one of storage. If you have a limited amount of storage space you can use thinning to throw away samples in such a way that only the minimum of information is wasted.

Thinning can be implemented by passing k to the argument `thin` of the function `posterior`. The following code continues the example of Section 6.3. The chain `rn.post2` is generated [40 secs] in the same manner as `rn.post`, except that we use a run length of $n = 20\,000$, and store only every fifth iteration.

```
> n <- 20000 ; t0 <- c(43.2, 7.64, 0.32) ; s <- c(2, .2, .07) ; b <- 2000
> rn.post2 <- posterior(n, t0, prrain, lh = "pp", data = rainfall, thresh = 40,
      noy = 54, psd = s, burn = b, thin = 5)
> rn.post2
             mu     sigma        xi
2000  43.00410  8.906956 0.3125027
2005  44.13408  8.599823 0.3489489
2010  44.13408  9.526747 0.3054351
[...]
19995 42.21891  7.679970 0.3158895
20000 44.50139  9.801986 0.2814907
```

7.1.2 Multiple Chains

In Section 6.1 we discussed the possibility of generating multiple Markov chains, with initial values scattered about the parameter space. This is the only way to ensure that the chain or chains have fully explored all regions of high probability, particularly when the target distribution is complex. I recommend generating a small number of chains and examining the iterations graphically. The burn-in period for each chain can easily be identified using this approach. In particular, it is possible to determine whether the chains have reached equilibrium (or not, in which case the burn-in period is larger than the current run length), which is very difficult to determine using only a single chain. On the other hand, many iterations will be discarded as there will often be a large burn-in period associated with each chain. Furthermore, if a single chain with run length 5 000 is generated in preference to five chains with length 1 000, the last 4 000 iterations will be sampled from a distribution that is likely to be closer to the target distribution than any of the samples that would have been generated in any of the smaller chains.

Continuing the example of Section 6.3, suppose that we generate [2 secs each] the chains `rna`, `rnb` and `rnc` in the same manner as `rn.post`, except that we use the starting values $\boldsymbol{\theta}_0^a = (40, 11, 0.2)$, $\boldsymbol{\theta}_0^b = (50, 5, 0.4)$ and $\boldsymbol{\theta}_0^c = (32, 6, 0.3)$. I have also reduced the run length to $n = 1000$ and omitted the burn-in period ($b = 0$). The iterations of the location parameter for each of the three chains are plotted in Figure 9.

The diagnostic of Gelman and Rubin (1992) is designed for multiple chains that have been run with starting values which are over-dispersed relative to the target distribution. (The starting

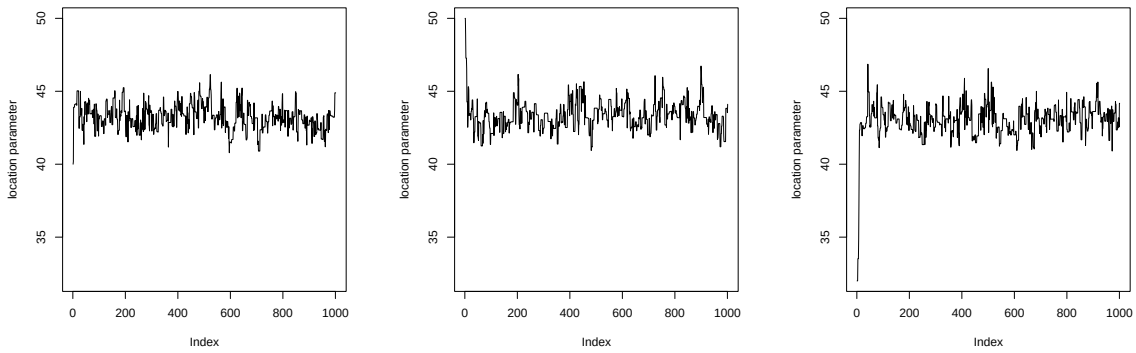

Figure 9: Iterations of the location parameter for each of the three chains `rna`, `rnb` and `rnc`. The starting values are $\mu_0^a = 40$, $\mu_0^b = 50$ and $\mu_0^c = 32$.

values in this example have been arbitrarily selected at points with low posterior density.) The diagnostic is implemented in **coda**, and can be performed using the following code. The function `mcmc.list` creates an object that **coda** can recognize as a list of Markov chains.

```
> rn.mcl <- mcmc.list(mcmc(rna), mcmc(rnb), mcmc(rnc))
> gelman.diag(rn.mcl, transform = TRUE)
Potential scale reduction factors:

     Point est. 97.5% quantile
[1,]       1.03           1.10
[2,]       1.03           1.09
[3,]       1.03           1.09

Multivariate psrf: 1.03
```

The diagnostics are based on estimates of the variance of the (margins of the) target distribution. If the chains have not reached equilibrium, the mean of the empirical variance within each chain (for each parameter) will underestimate the variance, because each chain will not have had the time to range over the target distribution. Similarly, the empirical between-chain variance (the variance of the empirical means of each chain, multiplied by the run length) will overestimate the variance, because the starting values are over-dispersed relative to the target distribution. The "point estimate" potential scale reduction factors (PSRF) are essentially the within-chain divided by the between-chain estimates of variance. If these factors are substantially larger than one, the simulated sequences may not have made a full tour of the target distribution. Gelman and Rubin (1992) recommend increasing the run length n until all the reduction factors are close to one, and then taking $b = n/2$, thus discarding the first half of the chain. This is the basis for the plot generated by `gelman.plot`. The condition of being "close" to one depends on the problem at hand; for most examples, values below 1.2 are acceptable (Gelman *et al.*, 1995).

The "97.5% quantile" PSRF is constructed in a similar manner, except that the variance ratio is replaced by the 97.5% quantile of its (estimated) sampling distribution. The multivariate potential scale reduction factor (MPSRF), due to Brooks and Gelman (1997), generalizes the original "point estimate" method to consider all parameters simultaneously.

The values shown here are sufficiently close enough to one to be acceptable. Incidentally, using

the run length $n = 10\,000$, as was used in Section 6.3, produced reduction factors that were all equal to one (to the number of decimal places printed by `gelman.diag`).

As with all diagnostics, there are criticisms. The diagnostic assumes that the initial values are sampled from a distribution that is over-dispersed relative to the target distribution. In practice the target distribution is unknown, so this is difficult* to achieve. The diagnostic also relies on a normal approximation to the samples of each parameter within the chain. The parameters can be transformed so that the normal approximation is more appropriate, but it remains a criticism.

7.2 Distributions of Quantiles

The Markov chains generated can be transformed in order to estimate other quantities of interest. In particular, the distributions of quantiles can be estimated. Let F be the GEV distribution function, and let $F(q_p) = 1 - p$, so that

$$q_p(\boldsymbol{\theta}) = \begin{cases} \mu - \frac{\sigma}{\xi}[1 - \{-\log(1-p)\}^{-\xi}] & \xi \neq 0 \\ \mu - \sigma \log\{-\log(1-p)\} & \xi = 0, \end{cases}$$

is the quantile corresponding to the upper tail probability p. For each p, the samples $\boldsymbol{\theta}_b, \ldots, \boldsymbol{\theta}_n$ can be substituted into the above expression to yield $q_p(\boldsymbol{\theta}_b), \ldots, q_p(\boldsymbol{\theta}_n)$. We can use these values to estimate features of the prior and posterior distributions of $q_p(\boldsymbol{\theta})$ in the same way that the values $\boldsymbol{\theta}_b, \ldots, \boldsymbol{\theta}_n$ have been used to estimate features of the prior and posterior distributions of $\boldsymbol{\theta}$.

Continuing the example of Section 6.3, prior and posterior density estimates of the quantiles $q_{0.1}$, $q_{0.01}$ and $q_{0.001}$ are shown in Figure 10. These are density estimates for the value that is exceeded by the annual maximum of daily rainfalls with probabilities 0.1, 0.01 and 0.001. The estimates can be plotted using the following code.

```
> poq <- mc.quant(rn.post, p = c(.1,.01,.001))
> prq <- mc.quant(rn.prior, p = c(.1,.01,.001))
> plot(density(poq[,1], adj = 2), xlim = c(20,100), ylim = c(0,.11))
> lines(density(prq[,1], adj = 2), lty = 2)
> plot(density(poq[,2], adj = 2), xlim = c(45,200), ylim = c(0,.05))
> lines(density(prq[,2], adj = 2), lty = 2)
> plot(density(poq[,3], adj = 2), xlim = c(125,350), ylim = c(0,.018))
> lines(density(prq[,3], adj = 2), lty = 2)
```

The function `mc.quant` takes two arguments. The first should be an object returned from `posterior`, which contains the values $\boldsymbol{\theta}_b, \ldots, \boldsymbol{\theta}_n$. If the second argument `p` $= p$, the function returns the vector $q_p(\boldsymbol{\theta}_b), \ldots, q_p(\boldsymbol{\theta}_n)$. If `p` $= (p_1, \ldots p_m)$ is a vector of length m, the function returns a matrix with jth column $q_{p_j}(\boldsymbol{\theta}_b), \ldots, q_{p_j}(\boldsymbol{\theta}_n)$, for $j = 1, \ldots, m$.

Although Figure 10 gives us density estimates for $q_{0.1}$, $q_{0.01}$ and $q_{0.001}$, it would be useful to have a graphical summary of the distributions of q_p for all (small) values of p. This can be done using a **return level plot**. A return level plot is a standard tool in extreme value theory. In the terminology of extreme value theory, return levels are simply quantiles. We would say that the value q_p is the **return level** associated with the **return period** $1/p$. A return level plot is a plot of q_p verses $-1/\log(1-p)$, for fixed values of (μ, σ, ξ), typically maximum likelihood

*But not impossible. The target distributions can be approximated using (mixtures of) multivariate normal or multivariate t distributions on $(\mu, \log\sigma, \xi)$, the parameters of which can be estimated using the output of `mposterior`, upon setting `hessian = TRUE`.

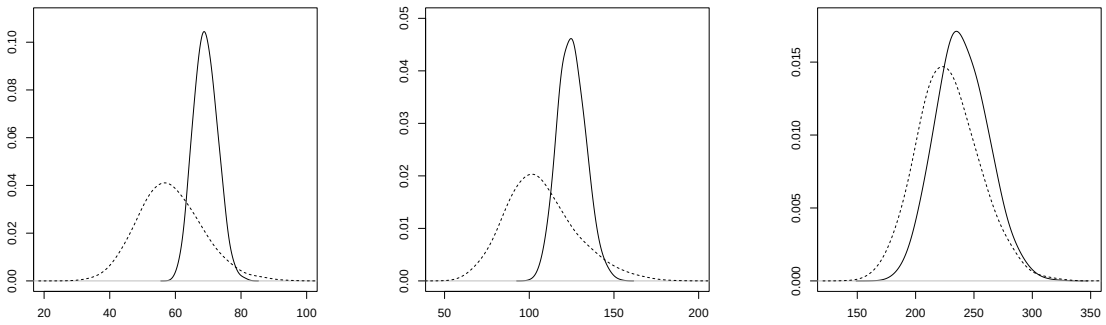

Figure 10: Prior (dashed lines) and posterior (solid lines) density estimates for the value (in millimetres) that is exceeded by the annual maximum of daily rainfalls with probabilities 0.1, 0.01 and 0.001 respectively.

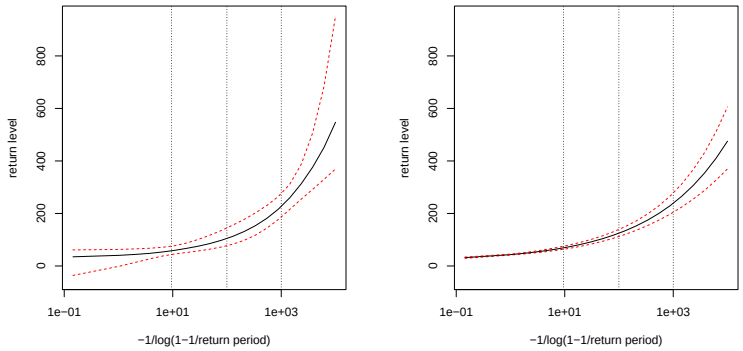

Figure 11: Return level plots of prior (left panel) and posterior (right panel) distributions for q_p. The curves within the plots represent medians (solid lines) and intervals containing 90% of the prior/posterior probability (dashed lines). Dotted vertical lines are drawn at $p = 0.1, 0.01, 0.001$.

estimates. The x-axis is plotted on a logarithmic scale. This emphasizes the values in the upper tail (small p), and makes the plot linear when $\xi = 0$, with slope σ and intercept μ. For small p, $-1/\log(1-p) \approx 1/p$, so the return level plot is approximately a plot of return levels verses return periods.

In our Bayesian framework, we can use a return level plot to illustrate the distributions of quantiles, or equivalently, return levels. For each p there is a corresponding sample $q_p(\boldsymbol{\theta}_b), \ldots, q_p(\boldsymbol{\theta}_n)$. We take a summary statistic for each sample, say the median, which we denote by $\check{q}_p$. We can then plot $\check{q}_p$ verses $-1/\log(1-p)$, again using a logarithmic scale on the x-axis. This gives us a curve of the medians of the prior/posterior distributions of q_p. We can also plot curves for other summary statistics, such as the empirical sample quantiles corresponding to the percentage points 0.05 and 0.95, which yield intervals containing 90% of the prior/posterior probability, for each q_p. Return level plots of this form are given in Figure 11. The plots depict the prior and posterior distributions of q_p. They can be created using `rl.pst(rn.prior)` and `rl.pst(rn.post)` respectively.

For a specific value of p, it may help to imagine a vertical line superimposed on the return level plot at $-1/\log(1-p) \approx 1/p$. The estimates for the quantiles of the prior/posterior distribution of

q_p are given by the y-coordinate of the intersection of this line with the plotted curves. In Figure 11 we have added vertical lines at $p = 0.1, 0.01, 0.001$, corresponding to the density estimates given in Figure 10.

7.3 Predictive Distributions

The primary objective of an extreme value analysis is often prediction. Let z denote a future observation with density function $f(z|\boldsymbol{\theta})$, where $\boldsymbol{\theta} \in \Theta$. The **posterior predictive density** of z, given observed data $\boldsymbol{x}$, is

$$f(z|\boldsymbol{x}) = \int_\Theta f(z|\boldsymbol{\theta})\pi(\boldsymbol{\theta}|\boldsymbol{x})\,\mathrm{d}\boldsymbol{\theta}. \tag{8}$$

If we are to observe a future observation z but we do not observe any data $\boldsymbol{x}$, our predictions are based on the **prior predictive density**

$$f(z) = \int_\Theta f(z|\boldsymbol{\theta})\pi(\boldsymbol{\theta})\,\mathrm{d}\boldsymbol{\theta}.$$

Predictive distributions reflect the uncertainty in the model and the uncertainty due to the variability of future observations.

Let $Z \sim \mathrm{GEV}(\boldsymbol{\theta})$, where $\boldsymbol{\theta} = (\mu, \sigma, \xi)$. Using expression (8), the posterior predictive distribution of a future observation z is given by

$$\Pr(Z \le z|\boldsymbol{x}) = \int_\Theta \Pr(Z \le z|\boldsymbol{\theta})\pi(\boldsymbol{\theta}|\boldsymbol{x})\,\mathrm{d}\theta,$$

where $\Pr(Z \le z|\boldsymbol{\theta})$ is the generalized extreme value distribution (2), evaluated at z. The prior predictive distribution $\Pr(Z \le z)$ is defined in a similar manner, replacing the posterior density $\pi(\boldsymbol{\theta}|\boldsymbol{x})$ with the prior density $\pi(\boldsymbol{\theta})$. Using our (prior and posterior) Markov chains $\boldsymbol{\theta}_b, \ldots, \boldsymbol{\theta}_n$, the predictive distributions can be estimated using

$$\frac{1}{n-b+1}\sum_{i=b}^{n} \Pr(Z \le z|\boldsymbol{\theta}_i). \tag{9}$$

Suppose that $\Pr(Z > z|\boldsymbol{x}) = p$, or that $\Pr(Z > z) = p$, so that z is the return level corresponding to the return period $1/p$. For each value of z, we can estimate p using expression (9). This information can be depicted in a return level plot (see Section 7.2). In other words, we can plot z verses the estimated values of $-1/\log(1-p) \approx 1/p$, using a logarithmic scale on the x-axis.

Continuing the example of Section 6.3, the lower curves within the return level plots of Figure 12 depict the prior and posterior predictive distributions, as described above. It may help to imagine a horizontal line superimposed on a return level plot at a specific value z. The x-coordinate of the point at which this line crosses the lower curve is (for sufficiently large z) approximately the inverse of the prior/posterior probability that the maximum daily rainfall over the next year will exceed z.

Let Z_L be the maximum daily rainfall over a future period of L years. The predictive distributions $\Pr(Z_L \le z)$ and $\Pr(Z_L \le z|\boldsymbol{x})$ can similarly be estimated using

$$\frac{1}{n-b+1}\sum_{i=b}^{n} \Pr(Z \le z|\boldsymbol{\theta}_i)^L,$$

which reduces to expression (9) when $L = 1$. The curves on the return level plots of Figure 12 depict the prior and posterior distributions of Z_L, for $L = 1, 2, 5$. The function `rl.pred` creates return level plots for predictive distributions. Figure 12 can be created using the following code. The values of L should be passed to the argument `period`. The vector `qlim` represents the quantiles at which the return level plot is evaluated.

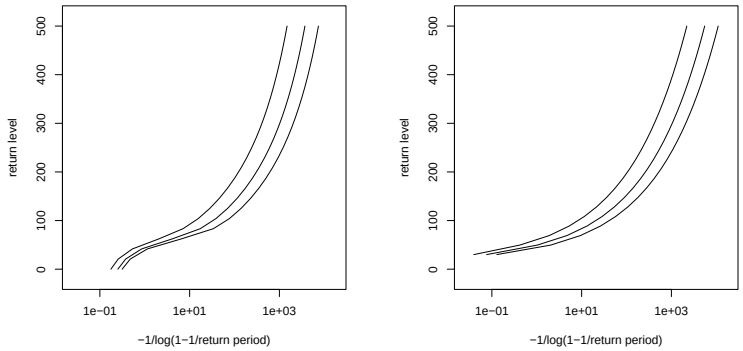

Figure 12: Return level plots of prior (left panel) and posterior (right panel) predictive distributions. The curves within the plots represent predictive distributions for maximum daily rainfall over a future period of one (lower), two (middle) and five (upper) years.

```
> rl.pred(rn.prior, period = c(1,2,5), qlim = c(0,500))
> rl.pred(rn.post, period = c(1,2,5), qlim = c(30,500))
```

7.4 Model Diagnostics and Sensitivity Analysis

Any analysis should include some check of the adequacy of the fit of the model to the data, and of the plausibility of the model for the purposes for which it will be used. In a Bayesian context, the model refers to both the prior distribution $\pi(\boldsymbol{\theta})$ and the likelihood $L(\boldsymbol{\theta}; \boldsymbol{x})$.

In practice, additional information is often available that is not included formally in the likelihood or the prior distribution. If this information suggests that posterior inferences are false, then more effort should be made to incorporate this information within the model. We can perform informal diagnostic procedures by comparing posterior distributions and posterior predictive distributions with aspects of reality that are not captured by the model. If there are any discrepancies, the model should be extended to include these aspects. Some possible extensions are discussed in Section 7.5.

A more formal diagnostic procedure compares the posterior predictive distribution to the data that have been observed (Gelman *et al.*, 1995). The basic technique is simple. We simulate samples from the posterior predictive distribution. These samples are then compared to the original data. Systematic discrepancies between the samples and the data correspond to features that are poorly fitted by the model. A balanced discussion of the advantages and disadvantages of this approach is given by Bayarri and Berger (1999, 2000). Further examples are given in Gelman *et al.* (1996).

Let us consider a specific example. Suppose we have data $\boldsymbol{x} = (x_1, \ldots, x_m)$, which we assume to be observed values of independent and identically distributed GEV($\boldsymbol{\theta}$) random variables. We need to simulate a sample from the posterior predictive distribution to which the data can be compared. This is done by generating m GEV($\boldsymbol{\theta}$) random variables, where $\boldsymbol{\theta}$ is *sampled from the posterior distribution*. Our Markov chain gives us $n - b + 1$ values $\boldsymbol{\theta}_b, \ldots, \boldsymbol{\theta}_n$, sampled from the posterior distribution. This leads to $n - b + 1$ samples of length m that can be compared to the actual data. If the point process likelihood (4) is used, the $n - b + 1$ samples can be compared to the period maxima derived from the actual data. Figure 13 demonstrates this process using the rainfall data from the example of Section 6.3. The plot on the left is a histogram of the

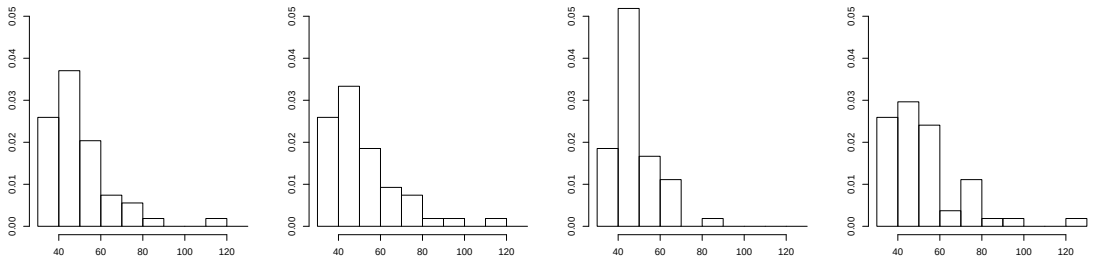

Figure 13: The left panel shows a histogram of the annual maxima of daily rainfall. The remaining three panels show histograms of samples from the posterior predictive distribution.

annual maxima of daily rainfalls. The remaining three plots depict samples from the posterior predictive distribution of annual maxima. They can be created using the following code, where the vector `rainmax` is constructed to contain the 54 annual rainfall maxima. The code includes the `rgev` simulation function, which is available in the **evd** package.

```
> yrs <- c(rep(c(366,365,365,365), 14), 366)
> yrs <- rep(1:57, yrs)
> myrs <- (yrs %in% c(32,38,42))
> rainmax <- tapply(rainfall[!myrs], yrs[!myrs], max, na.rm = TRUE)

> reprn <- cbind(matrix(0, nrow = 54, ncol = 3), rainmax)
> for(i in 1:3) {
    j <- 1000*(i-1) + 1
    reprn[,i] <- rgev(54, rn.post[j,1], rn.post[j,2], rn.post[j,3])
  }
> range(reprn) ; par(mfrow = c(2,2))
> for(i in 1:4) hist(reprn[,i], freq = FALSE, breaks = seq(30,130,10))
```

The three samples from the posterior predictive distribution are generated using the values $\boldsymbol{\theta}_{2000}$, $\boldsymbol{\theta}_{3000}$ and $\boldsymbol{\theta}_{4000}$ from the Markov chain `rn.post`. There are no clear systematic discrepancies between the samples and the data.

The code can easily be extended to create $n - b + 1$ posterior predictive samples using all the values $\boldsymbol{\theta}_b, \ldots, \boldsymbol{\theta}_n$, where $b = 2000$ and $n = 10000$. It is difficult to compare 8001 samples to the actual data using only graphical methods. Instead, we can define some function of the data $T(\cdot)$. We can then calculate the number of samples from the posterior predictive distribution for which the test statistic $T(\cdot)$ is greater than that for the actual data. In other words, if the replications are denoted by $\boldsymbol{x}^l$, for $l = b, \ldots, n$, we define p to be the proportion of the $n - b + 1$ simulations for which $T(\boldsymbol{x}^l) > T(\boldsymbol{x})$. If the value of p is close to zero or one, the test statistic $T(\cdot)$ corresponds to a feature that is poorly fitted by the model (Gelman *et al.*, 1995). The test statistic $T(\cdot)$ should be chosen to reflect aspects of the model that are relevant to the purposes to which the inference will be applied. In particular, $T(\boldsymbol{x}) = \max_j x_j$ will often be of particular importance for extreme value models.

The plots given in Figure 14 demonstrate this process, taking $T(\boldsymbol{x})$ as the largest value, the smallest value, the average and the standard deviation. The plot corresponding to $T(\boldsymbol{x}) = \max_j x_j$ can be created using the following code. The remaining plots can be constructed in a similar manner. None of the four test statistics yield a value of p close to zero or one.

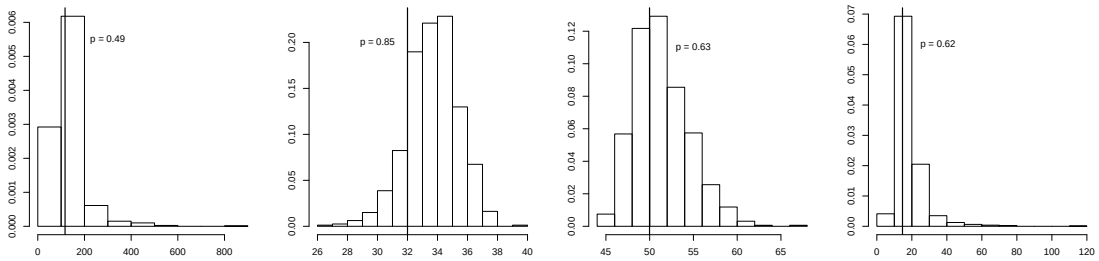

Figure 14: Histograms of test statistics of 8001 samples from the posterior predictive distribution. From left to right, the statistics are the largest value, the smallest value, the mean and the standard deviation. The corresponding values for the actual data are represented by vertical lines. The value p is given within each plot.

```
> reprn <- matrix(0, nrow = 54, ncol = 8001)
> for(i in 1:8001)
    reprn[,i] <- rgev(54, rn.post[i,1], rn.post[i,2], rn.post[i,3])
> repmax <- apply(reprn, 2, max)
> hist(repmax, freq = FALSE) ; abline(v=max(rainmax), lwd = 3)
> pv <- round(sum(repmax > max(rainmax))/8001, 2)
> text(300,.006, paste("p =", pv))
```

It is often the case that more than one model provides an adequate fit to the data. Sensitivity analysis determines by what extent posterior inferences change when alternative models are used. Alternative models may differ in the likelihood, or in terms of prior specification. The basic method of sensitivity analysis is to fit several models to the same problem. Posterior inferences from each model can then be compared. Posterior inferences will typically include marginal posterior distributions of the parameters (μ, σ, ξ), posterior distributions of GEV quantiles and posterior predictive distributions. The sensitivity of the marginal posterior density of the shape parameter ξ is often of particular interest.

7.5 Model Extensions

This section illustrates three extensions to the likelihoods of Section 3. Section 7.5.1 generalizes both the GEV and point process models to a frequently used form of non-stationarity. Section 7.5.2 discusses the implementation of a time-varying threshold within the point process characterization. Section 7.5.3 extends the GEV likelihood to incorporate upper order statistics. The likelihoods presented in this section are defined by continuity when $\xi = 0$.

7.5.1 Linear Trend for Location Parameter

The generalized extreme value log-likelihood (3) is based on the assumption that the data to be fitted are the observed values of independent random variables $X_1, \ldots, X_n$, where $X_i \sim \text{GEV}(\mu, \sigma, \xi)$ for each $i = 1, \ldots, n$. This assumption can be extended to $X_i \sim \text{GEV}(\mu_i, \sigma, \xi)$, where

$$\mu_i = \zeta + \eta t_i.$$

The parameters (ζ, η) are to be estimated, and the vector $\boldsymbol{t} = (t_1, \ldots, t_n)$ is specified by the user. It is assumed that $\boldsymbol{t}$ is approximately **centred** and **scaled**. If there is a linear trend present in

the data, t_i should be some centred and scaled version of the time of the ith observation. The log-likelihood (3) is extended to

$$-n \log \sigma - (1 + 1/\xi) \sum_{i=1}^{n} \log\{1 + \xi (x_i - \mu_i)/\sigma\} - \sum_{i=1}^{n} \{1 + \xi (x_i - \mu_i)/\sigma\}^{-1/\xi}.$$

The extension of the Poisson process log-likelihood (4) is similar. Recall that n_u of the n observations $x_1, \ldots, x_n$ exceed the threshold u, and $x_{(i)}$ denotes the ith exceedence, for $i = 1, \ldots, n_u$. The original log-likelihood is

$$-n_u \log \sigma - n_y \left\{1 + \xi \left(\frac{u - \mu}{\sigma}\right)\right\}_+^{-1/\xi} - \left(1 + \frac{1}{\xi}\right) \sum_{i=1}^{n_u} \log \left\{1 + \xi \left(\frac{x_{(i)} - \mu}{\sigma}\right)\right\}, \qquad (10)$$

provided that $1 + \xi(x_{(i)} - \mu)/\sigma$ for $i = 1, \ldots, n_u$ are positive. We again take $\mu_i = \zeta + \eta t_i$, for $i = 1, \ldots, n$. Let $\mu_{(i)}$ denote the location parameter that corresponds to the ith exceedence $x_{(i)}$. Then the log-likelihood is extended[†] to

$$-n_u \log \sigma - \frac{n_y}{n} \sum_{i=1}^{n} \left\{1 + \xi \left(\frac{u - \mu_i}{\sigma}\right)\right\}_+^{-1/\xi} - \left(1 + \frac{1}{\xi}\right) \sum_{i=1}^{n_u} \log \left\{1 + \xi \left(\frac{x_{(i)} - \mu_{(i)}}{\sigma}\right)\right\},$$

provided that $1 + \xi(x_{(i)} - \mu_{(i)})/\sigma$ for $i = 1, \ldots, n_u$ are positive.

To incorporate the linear trend term within a Bayesian analysis, a prior $\pi(\boldsymbol{\theta})$ must be specified on all four parameters $\boldsymbol{\theta} = (\zeta, \sigma, \xi, \eta)$. The construction of the prior proceeds in two stages. Firstly, a prior is constructed on (ζ, σ, ξ), using one of the techniques given in Section 4. Then we specify an independent prior normal distribution for η, with mean zero (since the vector $\boldsymbol{t}$ should be centred) and standard deviation `trendsd`, which is specified by the user.

When calling the function `posterior`, the initial value `init` must be extended to $\boldsymbol{\theta}_0 = (\zeta_0, \sigma_0, \xi_0, \eta_0)$, and the proposal standard deviations must be extended to $\boldsymbol{s} = (s_\zeta, s_\sigma, s_\xi, s_\eta)$. The vector $\boldsymbol{t}$ should be specified using the argument `trend`.

Continuing the example of Section 6.3, the following code generates Markov chains [13 and 32 secs respectively] with target distributions $\pi(\boldsymbol{\theta})$ and $\pi(\boldsymbol{\theta}|\boldsymbol{x})$, where $\boldsymbol{\theta} = (\zeta, \sigma, \xi, \eta)$. The initial values were derived using `mposterior`. The period 1932 – 1988 contains 20820 days, the 6576th of which is 1st January 1950. The trend parameter $\boldsymbol{t}$ is therefore specified so that ζ represents the location parameter on 1st January 1950 and η represents the increase (or decrease, if negative) over a period of 40 years (14610 days). We take `trendsd` = 10, representing a fairly flat marginal prior for η.

```
> shape <- c(38.9,7.1,47) ; scale <- c(1.5,6.3,2.6)
> prrain2 <- prior.quant(shape = shape, scale = scale, trendsd = 10)
> n <- 10000 ; t0 <-  c(50.8,1.18,0.65,0) ; s <- c(25,.35,.07,25) ; b <- 2000
> rn.prior2 <- posterior(n, t0, prrain2, lh = "none", psd = s, burn = b)
> t0 <- c(42.9,7.61,0.32,1) ; s <- c(2,.2,.07,4) ; tt <- (1:20820 - 6576)/14610
> rn.post2 <- posterior(n, t0, prrain2, lh = "pp", data = rainfall, thresh = 40,
      noy = 54, trend = tt, psd = s, burn = b)
```

[†]The term $\frac{1}{n}\sum_{i=1}^{n}\{1 + \xi(u - \mu_i)/\sigma\}^{-1/\xi}$ is an approximation to an integral. Since n is often very large, the package (by default) calculates $\frac{1}{|b|}\sum_{i \in b}\{1 + \xi(u - \mu_i)/\sigma\}^{-1/\xi}$ for an appropriate subset $b \subset \{1, \ldots, n\}$, with $|b| << n$. This behaviour can be overridden by setting `exact = TRUE`. This also applies to the likelihoods of Section 7.5.2.

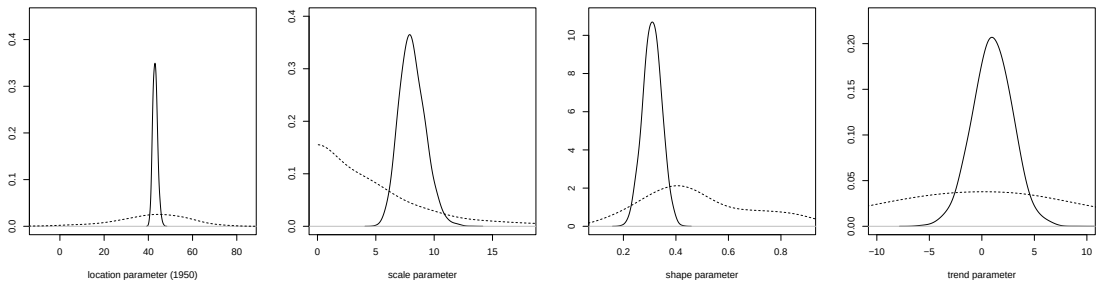

Figure 15: Marginal prior (dashed line) and posterior (solid line) density estimates for the generalized extreme value parameters ζ, σ, ξ and η respectively, in a Bayesian analysis of the rainfall data.

The marginal prior and posterior density estimates are shown in Figure 15. The marginal posterior density for η is approximately normal, with mean 1 and standard deviation 1.9. An increasing trend of one millimetre every 40 years does not represent a trend of any significance. Consequently, the prior and posterior distributions for (ζ, σ, ξ) are almost identical to those for (μ, σ, ξ) given in Figure 8.

7.5.2 Variable Thresholds

In the example of Section 6.3, the threshold for the Possion process likelihood (10) was chosen to be $u = 40$. We can extend this idea to allow variable thresholds. In other words, the threshold $\boldsymbol{u}$ can be a vector of length n, containing one value for each observation. The observation x_i is therefore an exceedence only if $x_i > u_i$. Let $x_{(i)}$ denote the ith exceedence. The log-likelihood is extended to

$$-n_u \log \sigma - \frac{n_y}{n} \sum_{i=1}^{n} \left\{ 1 + \xi \left(\frac{u_i - \mu}{\sigma} \right) \right\}_+^{-1/\xi} - \left(1 + \frac{1}{\xi} \right) \sum_{i=1}^{n_u} \log \left\{ 1 + \xi \left(\frac{x_{(i)} - \mu}{\sigma} \right) \right\},$$

provided that $1 + \xi(x_{(i)} - \mu)/\sigma$ for $i = 1, \ldots, n_u$ are positive. This likelihood can be implemented by passing a vector of length n to the argument `thresh`. If a shorter vector is passed to `thresh`, it is replicated until a vector of length n is created.

A linear trend term can also be included in the analysis, using the methods outlined in Section 7.5.1. In this case, the log-likelihood becomes

$$-n_u \log \sigma - \frac{n_y}{n} \sum_{i=1}^{n} \left\{ 1 + \xi \left(\frac{u_i - \mu_i}{\sigma} \right) \right\}_+^{-1/\xi} - \left(1 + \frac{1}{\xi} \right) \sum_{i=1}^{n_u} \log \left\{ 1 + \xi \left(\frac{x_{(i)} - \mu_{(i)}}{\sigma} \right) \right\}.$$

7.5.3 Order Statistics

Due to an asymptotic argument (e.g. Coles, 2001) the generalized extreme value log-likelihood (3) is often used when the data $\boldsymbol{x}$ consists of maxima from some underlying process. Suppose that the data $\boldsymbol{x}$ consists not only of maxima, but of the r largest order statistics. Specifically, suppose that $\boldsymbol{x} = (x_1^{(1)}, \ldots, x_1^{(r_1)}, x_2^{(1)}, \ldots, x_2^{(r_2)}, \ldots, x_m^{(r_m)})$, where $(x_i^{(1)}, \ldots, x_i^{(r_i)})$ are the largest r_i order statistics from year/period i, for $i = 1, \ldots, m$. It will often be the case that $r_1 = \cdots = r_m = r$. The same asymptotic argument used to justify the log-likelihood (3) for maxima leads to the

log-likelihood for order statistics

$$-\left(\sum_{i=1}^{m} r_i\right)\log\sigma - \sum_{i=1}^{m}\left\{1+\xi\left(\frac{x_i^{(r_i)}-\mu}{\sigma}\right)\right\}^{-1/\xi} - \left(1+\frac{1}{\xi}\right)\sum_{i=1}^{m}\sum_{k=1}^{r_i}\log\left\{1+\xi\left(\frac{x_i^{(k)}-\mu}{\sigma}\right)\right\},$$

provided that $1+\xi(x_i^{(k)}-\mu)/\sigma$ is positive for all $i=1,\ldots,m$ and $k=1,\ldots,r_i$. A linear trend term can also be included in the analysis, using the methods outlined in Section 7.5.1. In this case, the log-likelihood becomes

$$-\left(\sum_{i=1}^{m} r_i\right)\log\sigma - \sum_{i=1}^{m}\left\{1+\xi\left(\frac{x_i^{(r_i)}-\mu_i}{\sigma}\right)\right\}^{-1/\xi} - \left(1+\frac{1}{\xi}\right)\sum_{i=1}^{m}\sum_{k=1}^{r_i}\log\left\{1+\xi\left(\frac{x_i^{(k)}-\mu_i}{\sigma}\right)\right\},$$

where $\mu_i = \zeta + \eta t_i$ for $i=1,\ldots,m$.

The number of order statistics used within each year/period comprises a bias-variance trade-off: small values of r generate few data leading to high variance, whereas large values are likely to violate the asymptotic support for the model, leading to bias. The considerations involved in this choice are similar to those involved in the choice of threshold for the point process characterization. In practice, it is usual to select the r_i as large as possible, subject to adequate model diagnostics (Coles, 2001). For use in the **evdbayes** package, data of the form $(x_1^{(1)},\ldots,x_1^{(r_1)},x_2^{(1)},\ldots,x_2^{(r_2)},\ldots,x_m^{(r_m)})$ should be stored in a numeric matrix with m rows and $\max\{r_1,\ldots,r_m\}$ columns. The (i,j)th entry should contain $x_i^{(j)}$ if $j \le r_i$ and `NA` otherwise. If no order statistics are available within a particular year, the corresponding row should contain only `NA` values.

The numeric matrix `venice` contains the 10 largest sea levels (in centimetres) within each year in Venice for the period 1931–1981, except for the year 1935 in which only the six largest measurements are available. It is included in the **evd** package, and can be made available using `data(venice)`. The data are plotted in Figure 16, which can be reproduced using `matplot(1931:1981, venice)`. Figure 16 gives strong visual evidence for an increasing trend. We explicitly model this trend using $\mu_i = \zeta + \eta t_i$ for $i=1,\ldots,m$. (There also appears to be some cyclicity in the series, which we do not attempt to model.) We perform a naive Bayesian analysis, taking near-flat priors that reflect the absence of external information, in a similar manner to Section 6.1. The following code generates a Markov chain [40 secs] with target distribution $\pi(\boldsymbol{\theta}|\boldsymbol{x})$, where $\boldsymbol{\theta} = (\zeta,\sigma,\xi,\eta)$. The likelihood can be specified by setting `lh = "os"`, meaning "likelihood is order statistics". We take a run length $n = 10000$, a burn-in period $b = 2000$ and a thinning interval $k = 5$. The starting value has been derived using `mposterior`. The trend vector $\boldsymbol{t} = (t_1,\ldots,t_m)$ is specified so that ζ represents the location parameter in 1950 and η represents the increase (or decrease, if negative) in the location parameter over a period of 10 years. The proposal standard deviations have, as usual, been determined by pilot runs.

```
> mat <- diag(c(10000, 10000, 100))
> pv <- prior.norm(mean = c(0,0,0), cov = mat, trendsd = 100)
> t0 <- c(104, 11.7, -0.06, 0.48) ; tt <- (1:51 - 20)/10
> v.post <- posterior(10000, t0, pv, lh = "os", data = venice, trend = tt,
      psd = c(1.5, .05, .03, 1), burn = 2000, thin = 5)
```

The marginal posterior density estimates are depicted by the solid lines within Figure 17. The dashed lines give marginal posterior density estimates under the same model, but using only the five largest values within each year. The variances of the marginal posterior distributions inevitably increase when fewer order statistics are used.

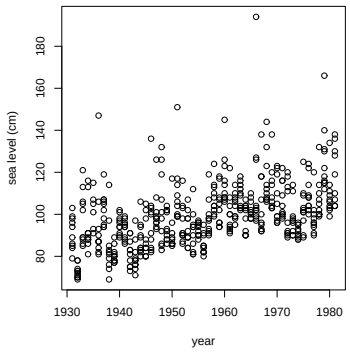

Figure 16: The 10 largest sea levels within each year in Venice for the period 1931–1981.

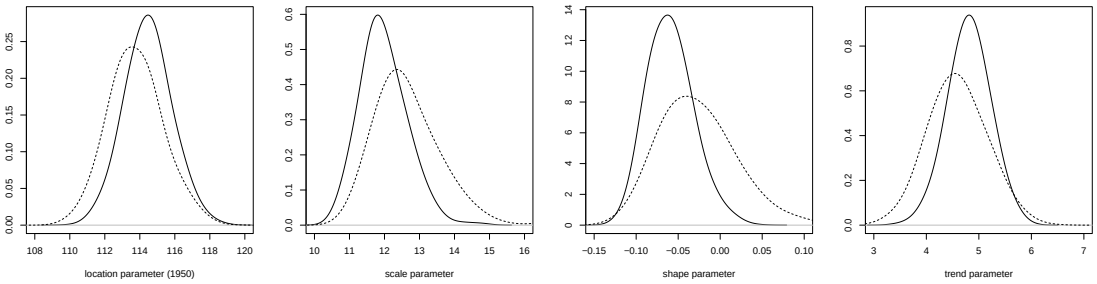

Figure 17: Marginal posterior density estimates for the generalized extreme value parameters ζ, σ, ξ and η respectively, in a Bayesian analysis of the venice data, using the ten largest (solid line) and five largest (dashed line) values within each year.

References

Bayarri, M. J. and Berger, J. O. (1999) Quantifying surprise in the data and model verification. In *Bayesian Statistics 6* (eds. J. M. Bernardo, J. O. Berger, A. P. Dawid and A. F. M. Smith). Oxford University Press, pp. 475–501.

Bayarri, M. J. and Berger, J. O. (2000) P-values for composite null models. *J. Amer. Statist. Assoc.*, **95**, 1127–1142.

Brooks, S. P. and Gelman, A. (1997) General methods for monitoring convergence of iterative simulations. *Journal of Computational and Graphical Statistics*, **7**, 434–455.

Brooks, S. P. and Roberts, G. O. (1998) Convergence assessment techniques for Markov chain Monte Carlo. *Statistics and Computing*, **8**, 319–335.

Coles, S. G. (2001) *An Introduction to Statistical Modeling of Extreme Values*. London: Springer–Verlag.

Coles, S. G. and Powell, E. A. (1996) Bayesian methods in extreme value modelling: a review and new developments. *Int. Statist. Rev.*, **64**, 119–136.

Coles, S. G. and Tawn, J. A. (1994) Statistical methods for multivariate extremes: an application to structural design (with discussion). *Appl. Statist.*, **43**, 1–48.

Coles, S. G. and Tawn, J. A. (1996) A Bayesian analysis of extreme rainfall data. *Appl. Statist.*, **45**, 463–478.

Cowles, M. K. and Carlin, B. P. (1996) Markov chain Monte Carlo convergence diagnostics: A comparative review. *J. Amer. Statist. Assoc.*, **91**, 883–904.

Crowder, M. (1992) Bayesian priors based on a parameter transformation using the distribution function. *Ann. Inst. Statist. Math.*, **44**, 405–416.

Gelman, A., Carlin, J. B., Stern, H. S. and Rubin, D. B. (1995) *Bayesian Data Analysis*. London: Chapman & Hall.

Gelman, A., Meng, X. L. and Stern, H. (1996) Posterior predictive assessment of model fitness via realized discrepancies (with discussion). *Statistica Sinica*, **6**, 733–807.

Gelman, A. and Rubin, D. B. (1992) Inference from iterative simulation using multiple sequences. *Statistical Science*, **7**, 457–511.

Geweke, J. (1992) Evaluating the accuracy of sampling-based approaches to the calculation of posterior moments. In *Bayesian Statistics 4* (eds. J. M. Bernardo, J. O. Berger, A. P. Dawid and A. F. M. Smith). Oxford University Press, pp. 169–193.

Ihaka, R. and Gentleman, R. (1996) R: A language for data analysis and graphics. *Journal of Computational and Graphical Statistics*, **5**, 299–314.

Pickands, J. (1971) The two–dimensional Poisson process and extremal processes. *J. Appl. Probab.*, **8**, 745–756.

Raftery, A. E. and Lewis, S. (1992) How many iterations in the Gibbs sampler. In *Bayesian Statistics 4* (eds. J. M. Bernardo, J. O. Berger, A. P. Dawid and A. F. M. Smith). Oxford University Press, pp. 763–773.

Smith, R. L. (1989) Extreme value analysis of environmental time series: an example based on ozone data (with discussion). *Statist. Science*, **4**, 367–393.

Stephenson, A. G. (2002) evd: extreme value distributions. *R News*, **2**, 31–32. URL `http://CRAN.R-project.org/doc/Rnews/`.

Stephenson, A. G. (2003) *Multivariate Extreme Value Distributions and their Application*. Ph.D. thesis, Lancaster University, Lancaster, UK.