in2extRemes: Into the R Package extRemes
Extreme Value Analysis for Weather and Climate Applications

Eric Gilleland
Richard W. Katz

NCAR Technical Notes
NCAR/TN-523+STR

National Center for Atmospheric Research
P. O. Box 3000
Boulder, Colorado
80307-3000
www.ucar.edu

NCAR | UCAR | National Center for Atmospheric Research

National Science Foundation
NCAR IS SPONSORED BY THE NSF

NCAR TECHNICAL NOTES

http://library.ucar.edu/research/publish-technote

The Technical Notes series provides an outlet for a variety of NCAR Manuscripts that contribute in specialized ways to the body of scientific knowledge but that are not yet at a point of a formal journal, monograph or book publication. Reports in this series are issued by the NCAR scientific divisions, serviced by OpenSky and operated through the NCAR Library. Designation symbols for the series include:

EDD – Engineering, Design, or Development Reports
Equipment descriptions, test results, instrumentation, and operating and maintenance manuals.

IA – Instructional Aids
Instruction manuals, bibliographies, film supplements, and other research or instructional aids.

PPR – Program Progress Reports
Field program reports, interim and working reports, survey reports, and plans for experiments.

PROC – Proceedings
Documentation or symposia, colloquia, conferences, workshops, and lectures. (Distribution maybe limited to attendees).

STR – Scientific and Technical Reports
Data compilations, theoretical and numerical investigations, and experimental results.

National Center for Atmospheric Research
P. O. Box 3000
Boulder, Colorado 80307-3000

NCAR/TN-523+STR
NCAR Technical Note

2013-December

in2extRemes: Into the R Package extRemes Extreme Value Analysis for Weather and Climate Applications

Eric Gilleland
Research Applications Laboratory, Weather Systems and Assesment Program
National Center for Atmospheric Research, Boulder, Colorado, U.S.A.

Richard W. Katz
Computational & Information Syatems Lab, Institute for Mathematics Applied to Geosciences
National Center for Atmospheric Research, Boulder, Colorado, U.S.A.

Research Applications Laboratory
Weather Systems and Assessment Program

NATIONAL CENTER FOR ATMOSPHERIC RESEARCH
P. O. Box 3000
BOULDER, COLORADO 80307-3000
ISSN Print Edition 2153-2397
ISSN Electronic Edition 2153-2400

Contents

List of Figures

List of Tables

PREFACE

This document serves as a tutorial/manual for the R package in2extRemes, which provides point-and-click windows for the extRemes (versions ≥ 2.0) package. The aim of in2extRemes and this document is to facilitate a shortened learning curve for performing statistical (univariate) extreme value analysis (EVA). Originally, the graphical user interfaces (windows) were provided by extRemes (versions < 2.0), which predominantly interfaced to the R package ismev, a software package that accompanies Coles (2001). Now, extRemes ≥ 2.0 is a command-line (no windows) EVA software package similar to ismev.

This document gives a brief overview of EVA, as well as a tutorial on how to install, start and use in2extRemes. Knowledge of R is not required to use in2extRemes apart from a few minor commands that are detailed here.

Table 1: Notation used in this manuscript.

EVA	Extreme Value Analysis
EVD	Extreme Value Distribution
EVT	Extreme Value Theory
df	(cumulative) distribution function
GEV	generalized extreme value
GP	generalized Pareto
PP	point process
POT	peak-over-threshold
qq-plot	quantile-quantile plot
iid	independent and identically distributed
μ	location parameter
σ	scale parameter
ϕ	logarithm of the scale parameter ($\phi = \ln \sigma$)
ξ	shape parameter ($\xi > 0$ implies a heavy tail df)
MLE	Maximum Likelihood Estimation/Estimate/Estimator
AIC	Akaike Information Criterion
BIC	Bayesian Information Criterion

1 Notation used in this tutorial

The notation used is designed to simplify later sections without having to show every GUI window. Therefore, it is important to familiarize yourself with this section. Some general abbreviations and notation used in this manuscript are given in Table 1 for ease of reference.

The command (`in2extRemes()` above) will open a GUI window similar to the one in figure 1 (actual appearance will depend on your specific operating system). This window will be referred to as the *main window*. At the top left are three menu items: **File**, **Plot** and **Analyze**. Clicking on any of these displays a submenu of choices (e.g., figure 2).

Sometimes submenu choices have a further submenu of selections that can be made. When this is the case, a small arrow appears to the right of the choice to indicate the existence of more choices (e.g., **Simulate Data** and **Transform Data** under **File** in figure 2).

If the instructions are to click on the **File** menu, then **Simulate Data**, followed

Figure 1: The main `in2extRemes` dialogue window. Actual appearance is system dependent.

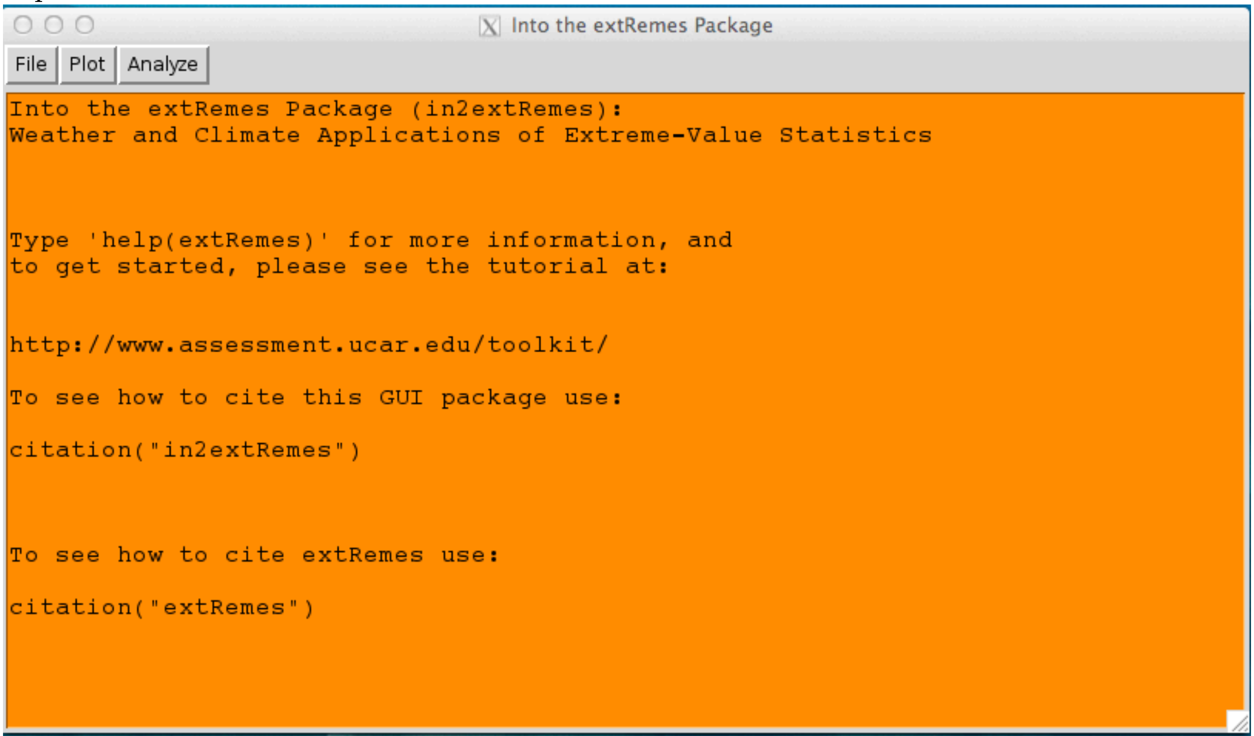
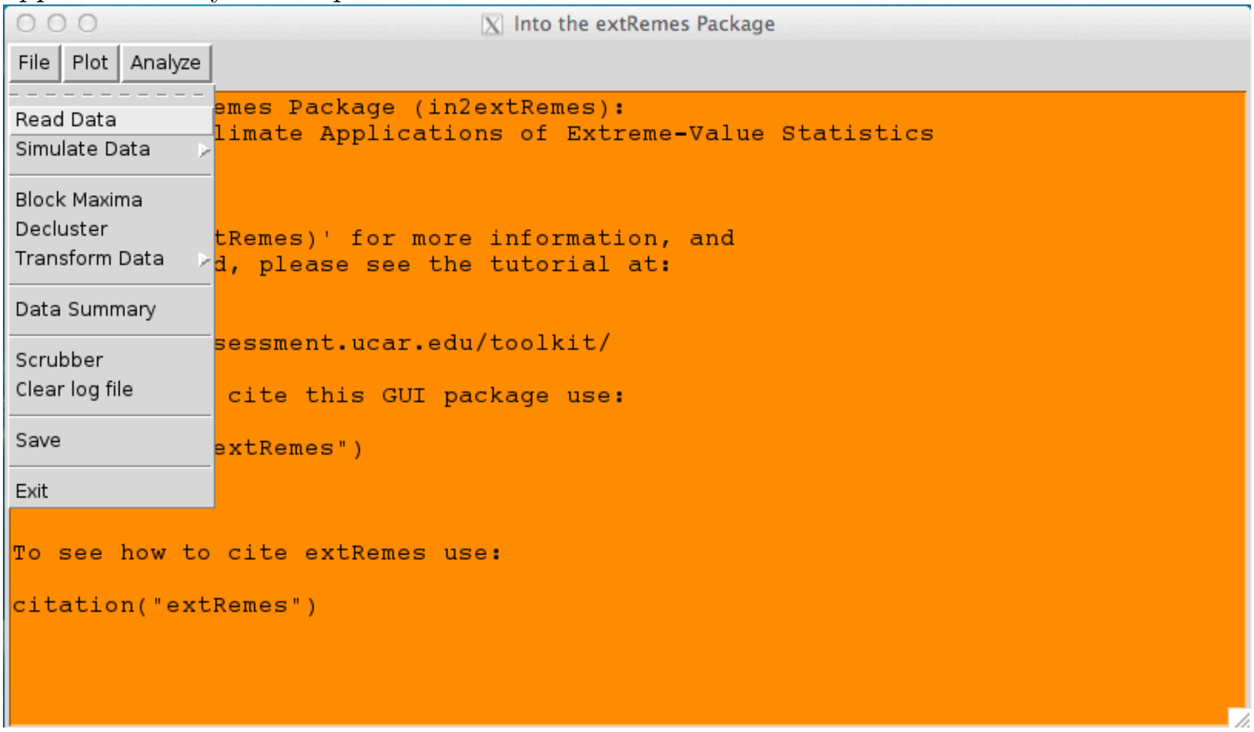


Figure 2: The main `in2extRemes` window showing the choices under **File**. Actual appearance is system dependent.



by **Generalized Extreme Value (GEV)**, then we will write

File $>$ **Simulate Data** $>$ **Generalized Extreme Value (GEV)**

Figures 2 - 4 show the submenus for each of the three main window choices. The choices are separated into groups of similar functionality. For example, the first two selections under **File** involve setting up data sets that can be handled by the various windows. In this case, a file can be read into the R session or data can be randomly generated from a GEV or GP df by selecting **Simulate Data**. In either case, an R object of class `"in2extRemesDataObject"` is created, which is a list object with certain set components. The important component being the `data` component, which contains the data frame object holding the data that has either been read from a file or simulated within R. It is possible to take an existing data object in R and convert it to an `"in2extRemesDataObject"` object using the `as.2extRemesDataObject` function. Only objects of class `"in2extRemesDataObject"` will be available for use within the `in2extRemes` windows. Note that unlike most GUI window programs, you must select which data set you want to use each time you open a new window; as opposed to selecting a data set and having each new window operate on the currently opened file.

The next three choices under the **File** menu manipulate existing data that have already been read into R and structured as `"in2extRemesDataObject"` class objects (i.e., data that are in a specific format and are recognized by the point-and-click windows). The first, **Block Maxima**, creates an entirely new data object, whereas the latter two simply add columns to the existing data set.

The **Block Maxima** option takes the maxima of one variable over blocks defined by another variable where both are columns of the same data frame (usually, a column containing the years is used for the blocks so that annual maxima are taken). A new `"in2extRemesDataObject"` object is created containing the block maxima data along with any other columns whose row entries correspond to the first instance of the block maximum within each block. An example of performing this step is given later in section 6.

The **Decluster** option will be discussed in more detail in section 10, and the options under **Transform Data** should be fairly straightforward.

Data Summary allows one to obtain a simple summary of a data set. Scrubber allows you to remove a data column or a fitted object from an existing `"in2extRemesDataObject"` object.

Clear log file erases the current `in2extRemes.log` file and replaces the contents with a message that the file has been cleared along with the date and time that it was cleared.

Save invokes the `save.image()` function to save the R workspace. It is a good idea to do this periodically.

Exit closes the main window, but does not quit the current R session (use `q("yes")`, `q("no")` or `q()` for that).

The remaining options under **Plot** and **Analyze** will be discussed in subsequent sections. However, to continue with the notation description, it is helpful to look at the **Scatter Plot** option under **Plot**. That is,

Plot > **Scatter Plot**

Figure 5 shows the window that pops up after the **Scatter Plot** selection is made. Most windows have the **Data Object** list box, which will display the available data objects. Note that if no `"in2extRemesDataObject"` class objects exist, then an error will be returned, which will show up in your R session window. A data object must be selected from this list box. In this example, a data set called `Phx` has been selected. In a grooved box just below the **Data Object** list box are two optional arguments. The first is a data entry field, which allows you to enter a value. In this case, the value must be a single character, and the default is a circle (the letter `"o"`). This option tells the `plot` function which character to

Figure 3: The main `in2extRemes` window showing the choices under **Plot**. Actual appearance is system dependent.

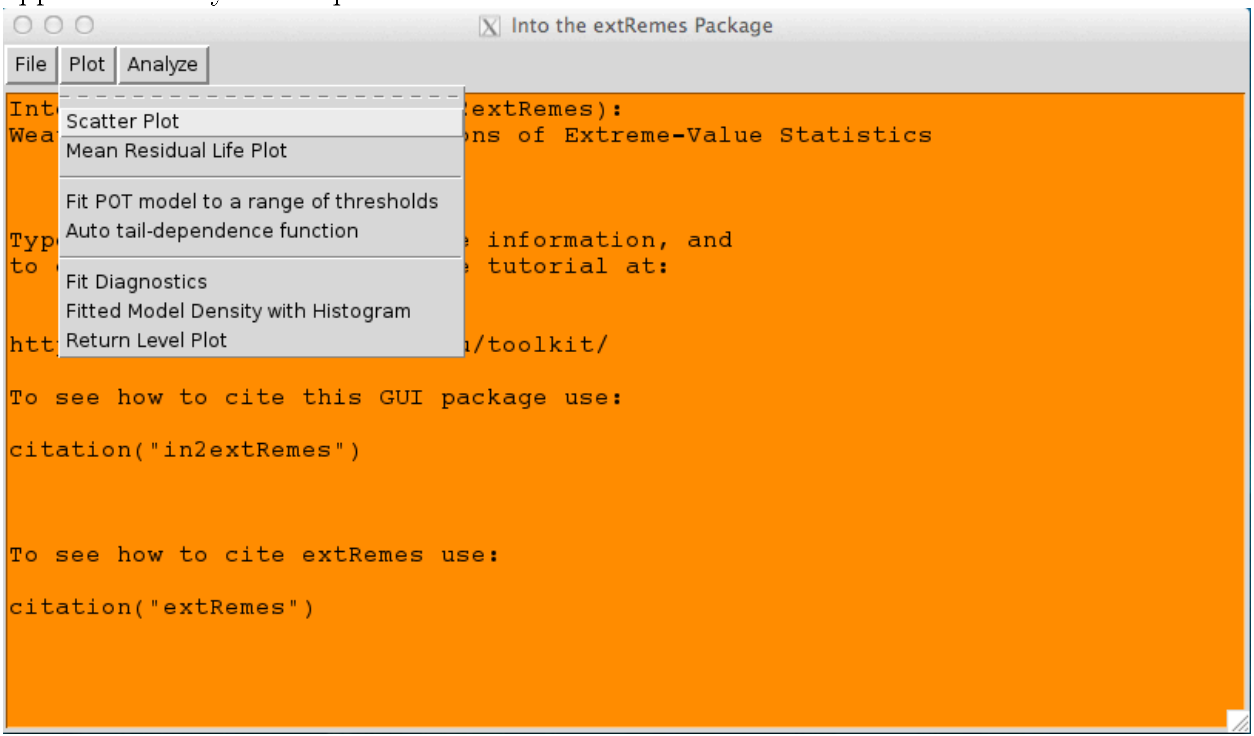


Figure 4: The main `in2extRemes` window showing the choices under **Analyze**. Actual appearance is system dependent.

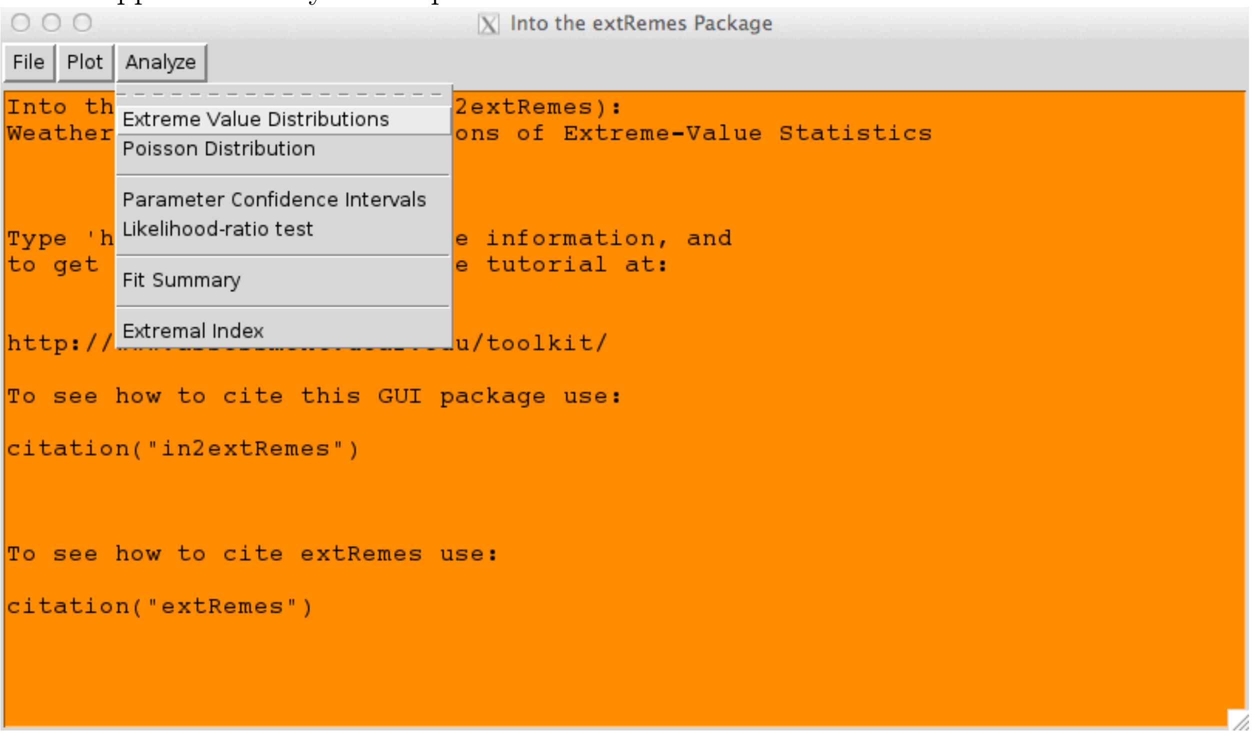
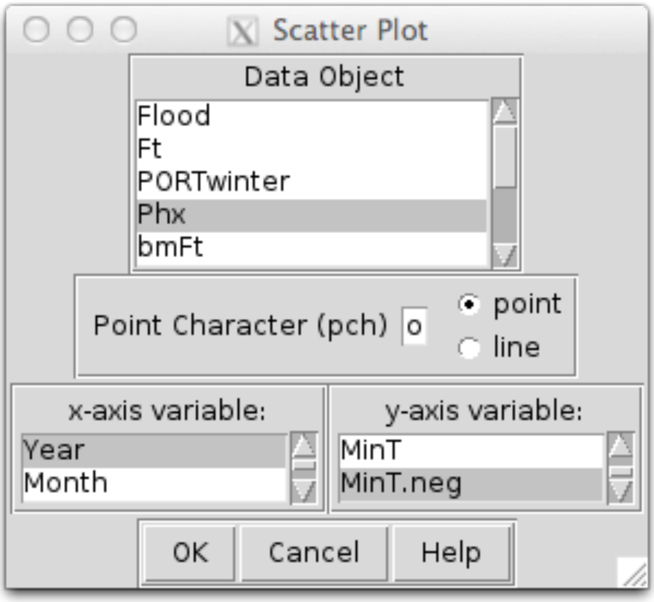


Figure 5: Example of a GUI window. Results from **Plot** > **Scatter Plot**.



use to make the plot. Next are two radio buttons giving a choice between making a scatter plot or a line plot (the default being a scatter plot). Finally, the abscissa and ordinate axes variables are selected in the next two list boxes. The values in these lists will be filled in after a selection from **Data Object** is made.

The short-hand notation that will be used to describe the above procedure for making a scatter plot according to the choices in figure 5 is as follows.

Plot > **Scatter Plot**
Select:
Data Object > `Phx`
x-axis variable: > `Year`
y-axis variable: > `MinT.neg`

Because the default selections for **Point Character (pch)** and **point** vs **line** radio buttons are used, they are not explicitly stated in the above short-hand instructions. Note that the different choice fields (e.g., **Data Object**) are given in bold face, while the specific selection choices are shown in `this font`.

2 Introduction

Extreme value analysis (EVA) refers to the use of extreme value theory (EVT) for analyzing data where interest is in rare, or low probability, events (e.g., annual maximum precipitation, temperature excesses over a very high threshold, wind speeds exceeding a high threshold, etc.). Not all high-impact events fall under the category of EVA, but typically the events under consideration also have a high impact on human life, economic stability, infrastructures, the environment, etc.

As an example, the recent heavy rain and consequent flooding in September 2013 in Colorado resulted in several deaths, the loss of property, severe damage to numerous roads, among other problems (http://wwa.colorado.edu/resources/front-range-floods/assessment.pdf). A single flooding event in the area is not unprecedented (e.g., Boulder, Colorado, where some of the worst flooding occurred, is named for the large boulders deposited by previous floods at the base of Boulder Creek), it is rare enough for the asymptotic assumptions of EVT to provide a good approximation. In particular, the Boulder Creek flooding was on the scale of a 50- or 100-year event meaning that the probability of seeing an event of such magnitude in a given year is on the order of 1/50 or 1/100.[1] Of course, the simultaneous flooding across the Colorado front range area was an extremely rare event (on the order of a 1000-year event). The assumptions for fitting extreme value distributions (EVD's) to data can, and should always, be checked, so it is not necessary to a priori decide on its appropriateness. A brief background on EVA is given in section 3, and the reader is referred to Coles (2001) for more detailed information (see also Beirlant et al., 2004; de Haan and Ferreira, 2006; Reiss and Thomas, 2007; Resnick, 2007).

The R (R Core Team, 2013) package `ismev` (Heffernan and Stephenson, 2012) is an R port of the same-named S-plus package accompanying Coles (2001). The R package `extRemes` (versions $<$ 2.0 Gilleland and Katz, 2011) provided graphical user interfaces (GUI's) or windows to functions from the `ismev` package, along with some additional functionality of its own. Since version 2.0, however, `extRemes` contains only command-line functions drawing on functionality from various EVA packages in R (see Gilleland et al., 2013, for a review of many of the existing packages in R for performing EVA). The present package, `in2extRemes`, replaces the windows into `ismev` with windows into `extRemes` $\geq$ 2.0 (Gilleland and Katz, 2014). For the most part, users familiar with `extRemes` $<$ 2.0 will not find it difficult to switch to the new GUI's. Many of them are identical to those used previously. Output from some of the functions will look different; especially many

[1] Much uncertainty is associated with EVA. Initial estimates were that the Boulder Creek flooding was a 100-year event, and such an event cannot be ruled out (within the associated uncertainty). Later estimates put the flooding at a 50-year event. On the other hand, flooding in nearby Longmont, Colorado was at the 500-year level.

of the plots. Section 4 provides all of the information required to install and open `in2extRemes`.

3 Crash Course on Extreme Value Analysis

The fundamental results of EVT come from a basic property associated with the maximum value of a sequence of sample observations. Suppose, for example, that we have a sample $x_1, \ldots, x_{200}$. If we divide this sequence into two subsequences by splitting it in the middle (i.e., $x_1, \ldots, x_{100}$ and $x_{101}, \ldots, x_{200}$), then the maximum value of the original sequence can be obtained, indirectly, as the maximum of the maximum of the two series. That is,

$$\max\{x_1, \ldots, x_{200}\} = \max\{\max\{x_1, \ldots, x_{100}\}, \max\{x_{101}, \ldots, x_{200}\}\}.$$

Consequently, an approximate distribution for the maximum must have a distribution function (df) F that satisfies $F^2(x) = F(ax + b)$, where $a > 0$ and b are scaling and centering parameters, respectively, that arise because the maximum value of a sequence increases as the length of the sequence increases. Such a df is called *max stable*.

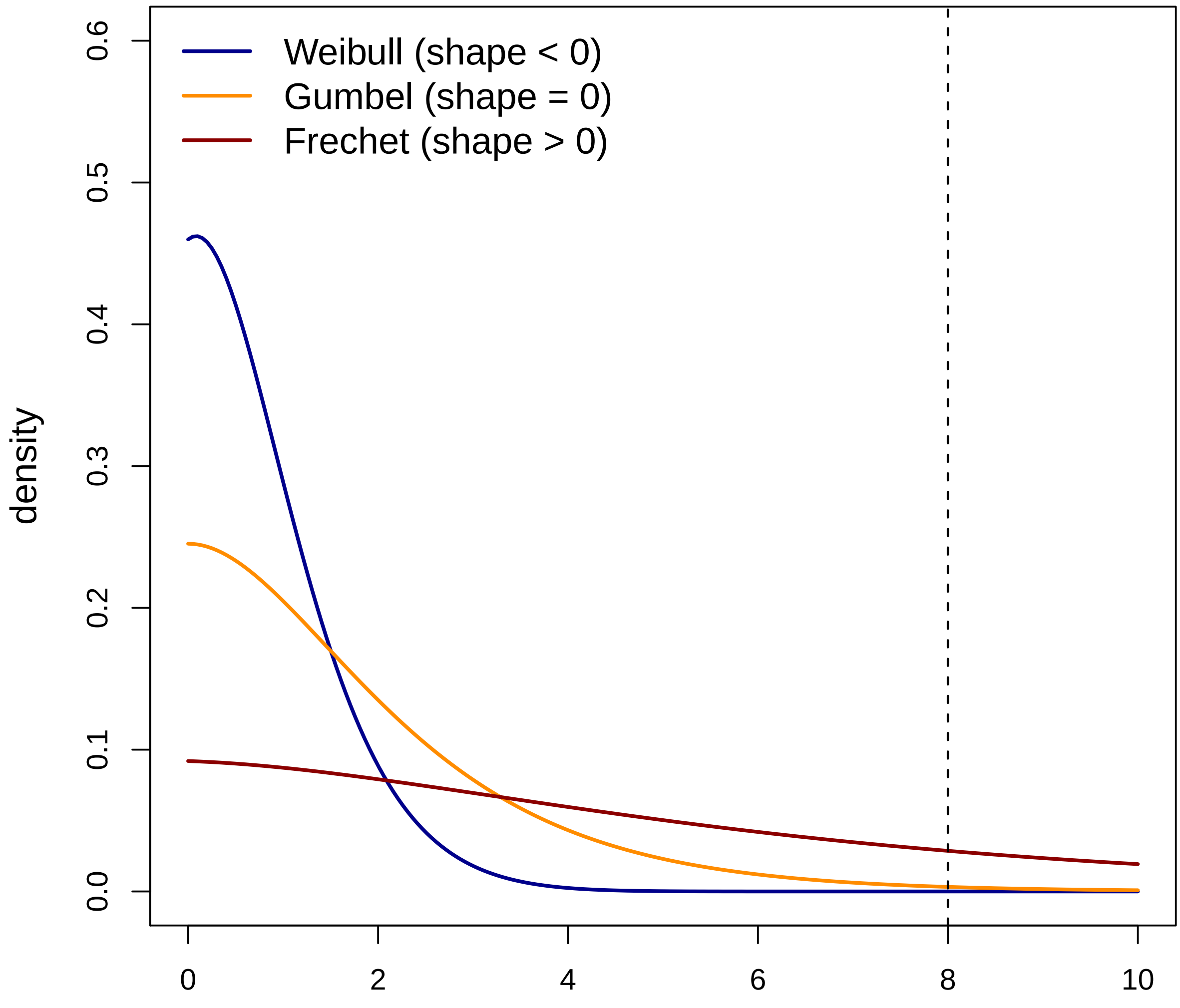
The only max-stable df's are in the form of the generalized extreme value (GEV) family. This family can be written in a simple form involving three parameters: location (denoted, here, as μ), scale ($\sigma > 0$) and shape (ξ). Different authors parametrize the GEV df differently. In this document, as in the `extRemes` package, we parametrize the GEV df so that a positive shape parameter implies a heavy tail df (i.e., the upper tail decays polynomially) and a negative shape parameter implies a bounded upper tail. A zero-valued shape parameter, defined by continuity, yields the light-tailed (tail decays exponentially) df. Figure 6 shows an example of how the sign of the shape parameter affects the GEV probability density function.

If interest is in the minima of sequences, the same techniques can be easily applied once we realize that $\min\{x_1, \ldots, x_n\} = -\max\{-x_1, \ldots, -x_n\}$. That is, we can simply take the negative transformation of our sample of interest, apply the methods for maxima, and remember to transform back in the final analysis.

Of course, simply looking at the maximum (or minimum) of a sequence is not the only approach for analyzing extremes. We may wish to analyze values that exceed some high threshold (typically in terms of the excesses over the threshold). For simplicity, we begin with the exponential df, which has an important property of being *memory-less*. That is, $\Pr\{X > x + u | X > u\} = \Pr\{X > x\}$ for any $x > 0$ and $u > 0$.

In order to approximate the upper tail of any df, the memory-less property must be weakened to peak over threshold (POT) stability. A POT stable df has

Figure 6: Example of generalized extreme value (GEV) probability density functions for each of the three types of tail behavior. Scale parameter varies to magnify differences in the tail behavior. Dashed vertical line shows upper bound of the (reverse) Weibull distribution. Location parameter is fixed at 0, and different values of this parameter would simply shift the densities shown left or right.



a shape that remains the same, but may otherwise be rescaled as the threshold is increased. The generalized Pareto (GP) df is POT stable meaning that the rescaled excess over the high threshold, u,

$$Y(u) = \frac{X - u}{\sigma(u)}$$

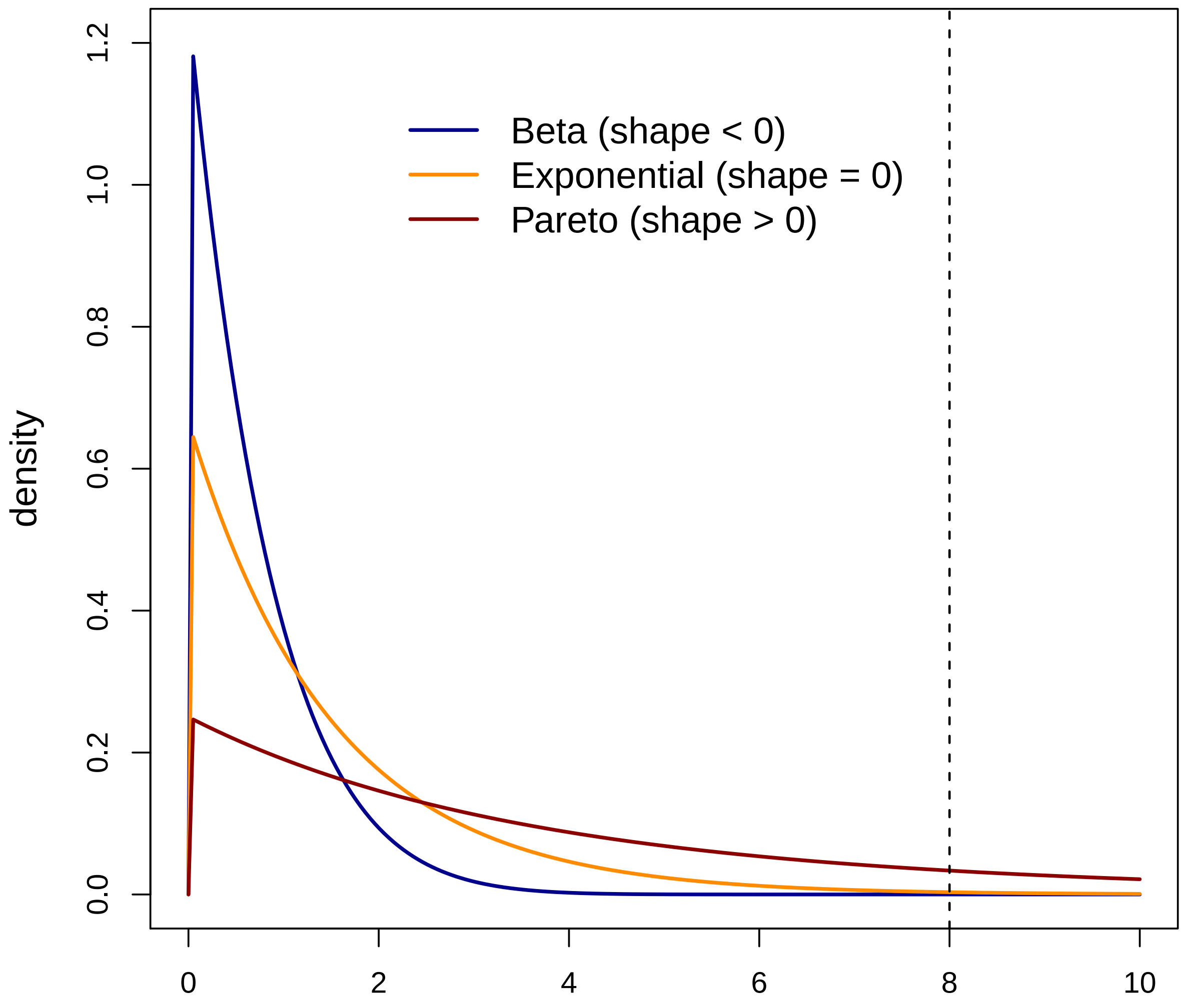
has the same df as u increases. The notation $\sigma(u) > 0$ emphasizes that the scale factor is a function of the threshold. In particular, $\sigma(u) = \sigma + \xi(u - \mu)$, where σ and μ correspond to the scale and location parameters of the equivalent GEV df characterizing the df for the maximum taken from the same original series (the shape parameter ξ is the same for both df's).

As one might anticipate, the POT-stability of the GP df is consistent with the max-stability of the GEV df. Moreover, the GP df also consists of three types of df's, and again, different authors parametrize the shape parameter differently. Analogously as for the GEV df, we parametrize the GP df so that a positive shape parameter implies a heavy upper tail and negative shape implies a bounded upper tail. Regardless of parametrization, a zero-valued shape parameter, defined by continuity, again yields a df with a light upper tail; which, in this case, is the aforementioned memory-less exponential df. The threshold for the GP df takes the place of a location parameter, so that there are effectively only two parameters for the GP df. If the GEV approximation (with a given shape parameter) holds for the maximum of a random variable, then the GP df provides an approximation for the upper tail and with precisely the same shape parameter. Figure 7 shows an example of how the GP probability density function varies according to the sign of the shape parameter.

When interest is in simply the frequency of occurrence of an event (e.g., a variable's exceeding a high threshold), and the probability of the event is low, then the Poisson distribution provides a good approximation. In fact, a two-dimensional point process (PP) can be employed whereby both the frequency and intensity are analyzed simultaneously. Such a model is sometimes referred to as a marked point process whereby, in this case, a one-dimensional Poisson process for the frequency component is employed and the marks (threshold excesses) are modeled by the GP df. Such a model can be easily parametrized as a GEV df, with the block length usually taken to be annual. This model is sometimes referred to as a Poisson-GP model, and the parameters are estimated orthogonally. However, it can also be modeled as a PP, which has the advantage of being able to fit both the frequency and GP parameters simultaneously so that account of the uncertainty in the parameter estimates is properly taken.

A natural question to consider before analyzing extreme values of a data set concerns which of the above general approaches (block maxima vs POT) to take. Figure 8 is an example demonstrating the pros and cons of each approach. The

Figure 7: Example of generalized Pareto (GP) probability density functions for each of the three types of tail behavior. Scale parameter varies to magnify differences in the tail behavior. Dashed vertical line shows upper bound of beta distribution.



raw data are shown in the upper left panel. In the upper right panel, blocks are marked by vertical dashed lines and the maximum in each block is indicated by a large orange number. Finally, values that exceed a high threshold are shown as black squares in the lower left panel of the figure.

The POT approach typically utilizes more of the available data than the block maxima approach, which can be vital given that extremes are rare so that they are associated with a paucity of data. However, it can be common for threshold excesses to *cluster* above a high threshold; especially with atmospheric data. While not important in terms of fitting EVD's to data, subsequent uncertainty analysis will be questionable (e.g., return level confidence intervals will be too narrow because effectively fewer data points are available than were used to fit the model).

Other idiosyncrasies to note from Figure 8 are that the block maxima approach may include points that are not very extreme (e.g., maxima number 1 in the figure), while in some cases it might miss extreme values simply because a larger value occurred somewhere else in the block (e.g., the second, or third, point that exceeds the threshold). On the other hand, the block maxima approach typically satisfies the independence assumption to a good approximation, and is easily interpretable in terms of return values.

When analyzing extremes of atmospheric phenomena, one often encounters non-stationarity in the data. That is, the df is not constant over time so that the df for the extremes may have a gradual trend or shift over time; even abrupt changes have been known to occur (e.g. Gilleland and Katz, 2011). The usual method for analyzing such data is to fit an EVD with parameters that vary as a function of a covariate (e.g., time is often used).

For reference, the exact form of the EVD's are given in the appendix along with tables showing some of their properties. Different methods for parameter estimation exist, and the most popular one is arguably maximum likelihood estimation (MLE), which easily allows for inclusion of covariate terms. MLE requires optimizing the likelihood function, which for EVD's does not have a closed form analytic solution. Therefore, numerical optimization is required.

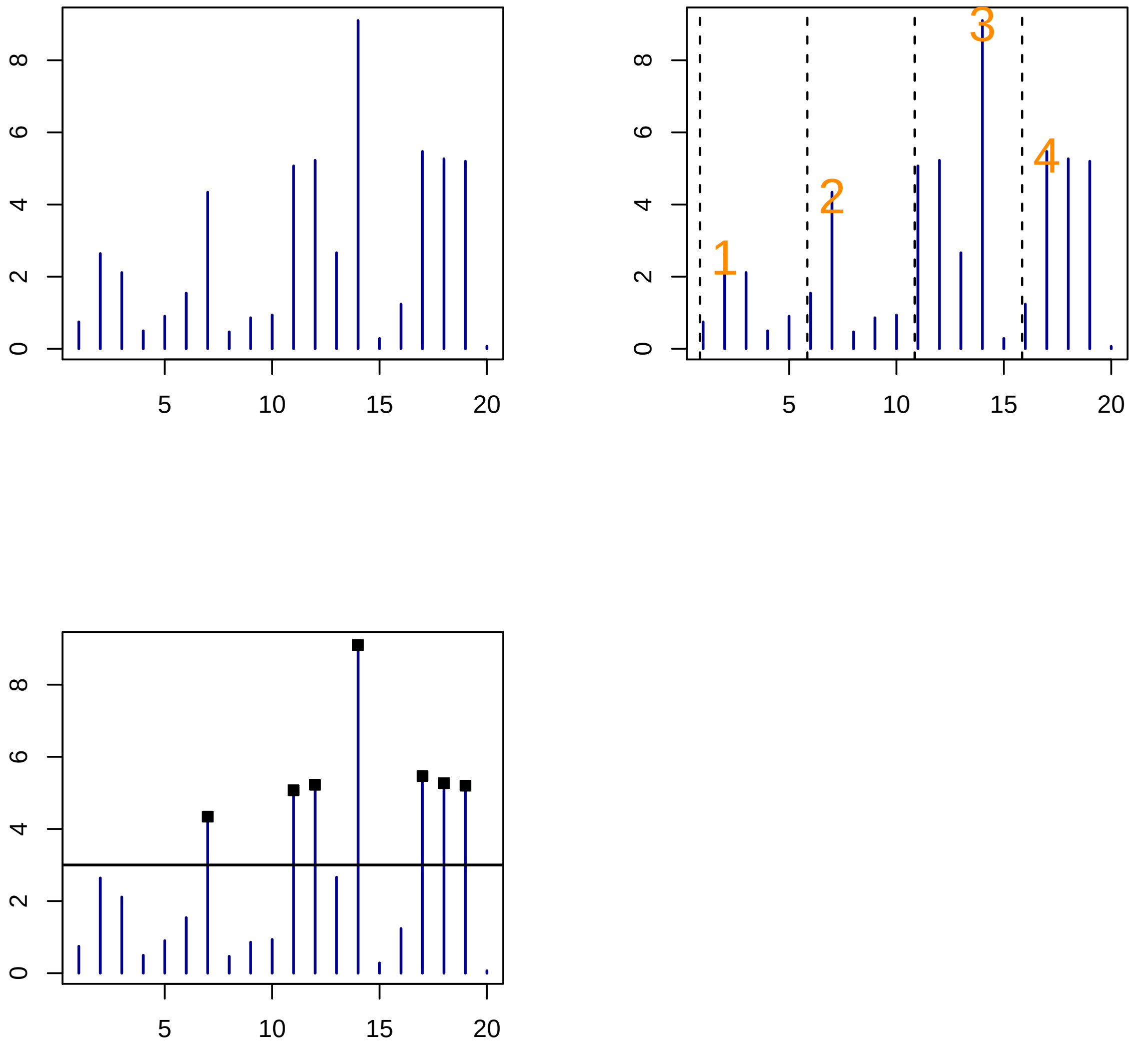
The package `extRemes` has a few other choices for parameter estimation, including Bayesian. However, `in2extRemes` currently only supports the MLE option.

4 Getting Started

4.1 R Basics

Installing R Installing R depends on your specific system, and may occasionally change with new versions of R. Therefore, the reader is referred to the R web page (http://www.r-project.org) for up-to-date instructions. Be sure to include the

Figure 8: Example data (upper left panel) showing block maxima (numbered in upper right panel where vertical dashed lines represent the start of a new block) and where points exceed a threshold (lower left panel; horizontal line shows the threshold and values exceeding the threshold are marked with a black square).



Tcl/Tk tools, which generally are not included with the base R distribution. It may be necessary to load these tools separately, so read the instructions provided on the R web site carefully. One of the most frequently asked questions concerns having installed R, and getting an error message about the `tcltk` package not available when trying to run `in2extRemes`.

Once R has been installed (with the Tcl/Tk tools, and subsequently also the `tcltk` package, which is separate from the Tcl/Tk tools), then you will need to install the `in2extRemes` package (all other required packages, such as `extRemes`, will be automatically installed). The installation can easily be done from within your R session.

Starting an R session To begin an R session, simply double click on the R icon (Mac GUI and Windows) or type R from the command prompt (linux, unix and Mac Xterm or Terminal users). Note that when an R session is opened, a file called `.RData` is created. This file will be located in the same directory as where R was opened (linux, unix, Mac Xterm/Terminal). Windows and Mac GUI users can type `getwd()` from the R command prompt to see where it exists on their systems. To see how to change this directory (primarily Windows and Mac GUI users, linux/unix users can just open R in a different folder), type `help(setwd)` from the R command prompt.

Installing an R package To install `in2extRemes`, type:

```
install.packages("in2extRemes")
```

You will be asked to select a CRAN mirror. Select one near to your location and click on the OK button. Once you have installed `in2extRemes` successfully, then you will not need to use the above command again. To be sure to have the most recent version of all your packages, you can use `update.packages()`, which will update all installed packages.

Loading a package Once you have installed `in2extRemes`, and every time you begin a new R session, you will need to load the package in order to use it. To do so, use the following command.

```
library(in2extRemes)
```

This command will also load `extRemes` and other required packages; namely, `Lmoments` (Karvanen, 2011), `distillery` (Gilleland, 2013), and `car` (Fox and

Weisberg, 2011). Note that when installing a package, quotation marks are used (e.g., `"in2extRemes"`), but when loading a package, they are not used.

R functions and help The R project web site has good introductory manuals for using R, so the reader is referred there. However, just to understand some basic syntax, a few brief remarks are made here. You may have noticed that functions are called by their name with arguments listed in parentheses, and when no arguments are used, the parentheses are still required. If the parentheses are not used, then the function's source code is listed out (unless the function is invisible). Every function has an associated help file, which can be seen using the `help` command, or using a `?` symbol before the function name. For special commands like matrix multiplication (i.e., `%*%`) quotation marks are used (e.g., `?"%*%"`). Sometimes packages have help files (e.g., type `?extRemes` after loading the package) to give an overview of the package. Many packages that have example data sets will also have help files for those data sets. The package `extRemes`, for example, contains several example data sets, which are named in the help file for the package. Often in this document, the reader will be referred to a help file in R. When this is the case, the notation used will be simply `?function.name`.

As you begin to get familiar with R and EVA, some potentially useful help functions from `extRemes` include:

```
?fevd
?devd
?pextRemes
?decluster
?extremalindex
?lr.test
?threshrange.plot
?mrlplot
?atdf
?taildep
?taildep.test
```

Starting `in2extRemes` To begin using the GUI windows, type the following command.

```
in2extRemes()
```

in2extRemes.log File Whenever a command is executed from an `in2extRemes` GUI window, the command is copied to a file called `in2extRemes.log`, which is a simple ASCII text file that shows the entire history of commands invoked by the windows. This file can be very useful for learning how to perform EVA from the command line in R, which will eventually allow you to accomplish many more tasks than just those available from the windows. This file is written to the same directory as your current R workspace (use `getwd()` to find the path where that is).

Main object types in R The main object types that you will likely deal with in R include: data frames, lists, vectors, matrices and arrays. A data frame is generally what you get after reading a data set into R. It is a matrix-like object in that it has the same number of rows for each column. Unlike a matrix, its columns are necessarily named (default names if none are present are `V1`, `V2`, etc.) and can be accessed by name. For example, suppose `x` is a data frame object in R with columns: `"height"` and `"weight"`. The `weight` column can be accessed by `x[["weight"]]` or by `x$weight`. The latter option requires less typing, but cannot be used by the `in2extremes` windows. Therefore, when looking in the `in2extRemes.log` file, you will only see the former, long-hand, approach.

Although a data frame object must have the same number of rows for each column, each column may have wildly different data types. For example, it is possible to have a column of date objects giving a date (and perhaps time) stamp for each data entry, another column giving a character or flag of sorts, while other columns may be numeric, integer, etc.

A list object is essentially a collection of components that may be very different from one another. For example, you might have a component that is a function, another that is a list, another that is a vector of length 10, another that is a vector of length 3, and another that is a matrix or data frame. A data frame object is a type of list object. Matrices, vectors and arrays are fairly self explanatory, but every component must have the same type (i.e., character, numeric, etc.). However, it is possible to change types on the fly (unlike, for example, in C).

Missing Values Missing values are denoted as `NA` in R. The logical function `is.na` can be useful for determining whether or not any missing values exist. Similarly, infinite values may be represented by `Inf` or `-Inf`. Division by zero results in `NaN` (not a number). Each may at times be handled similarly or differently depending on a specific function. Generally, missing values will result in `NA` when functions are applied (e.g., `1 + NA` results in `NA`).

Citing R and R Packages Because of the amount of work involved in writing software, and the need for funding, it is highly recommended to cite your use of software in research articles, etc. To see how to cite the R programming language in research articles, type the following command from the R prompt, which will show how to cite the software in both LaTeXand regular text formats.

```
citation()
```

To see how to cite R packages in research articles, the same command is used, but the package name is given in quotes. So, for example, when citing `in2extRemes`, you should also cite `extRemes`. The references to cite may change with new versions, so you should always use the following commands in order to cite the correct reference for the specific version of the software that you are using.

```
citation("in2extRemes")
citation("extRemes")
```

Saving the workspace and quitting R As objects are assigned in R, they exist in the `.RData` file and are not saved until explicitly saved either by the command `save.image()` or upon exiting the R session when selecting "yes" as prompted after entering the command `q()`. It is also possible to quit R and save the workspace with the command `q("yes")`, or quit R without saving the workspace with `q("no")`.

5 Data

Reading data into R can be performed in a number of ways (e.g., `read.data`). However, in order for the data to be "seen" by the `in2extRemes` windows, it must be of a particular type. One way to ensure that it is in the correct form is to load the data from a file using the `in2extRemes` windows (i.e., **File** > **Read Data**; cf. section 5.1). Alternatively, data frame objects already in your R workspace may be converted to the correct format using `as.in2extRemesDataObject` (cf. section 5.2).

Table 2: United States total economic damage (in billions of US dollars, USD) caused by floods (USDMG) by hydrologic year from 1932–1997. Also gives damage per capita (DMGPC) and damage per unit wealth (LOSSPW). See Pielke and Downton (2000), Katz et al. (2002) and `?Flood` for more information about these data. The full data set can be loaded, via the `extRemes` package, into the R workspace using `data(Flood)`.

OBS	HYEAR	USDMG	DMGPC	LOSSPW
1	1932	0.1212	0.9708	36.73
2	1933	0.4387	3.4934	143.26
3	1934	0.1168	0.9242	39.04
4	1935	1.4177	11.1411	461.27
⋮	⋮	⋮	⋮	⋮
64	1995	5.1108	19.4504	235.34
65	1996	5.9774	22.5410	269.62
66	1997	8.3576	31.2275	367.34

5.1 Loading Data into `in2extRemes`

There are two general types of datasets that can be read in using `in2extRemes`. One type is referred to, here, as *common* and the other, *R source*. Common data can take many forms as long as any headers do not exceed one line, and the rows represent the observations and the columns the variables. For example, Table 2 represents a typical common dataset; in this case data representing U.S. flood damage. See Pielke and Downton (2000) or Katz et al. (2002) for more information on these data.

R source data refer to data that have been dumped from R. These typically have a .R or .r extension. That is, it is written in R source code from within R itself. Normally, these are not the types of files that a user would need to load. As an R source file, the same dataset in Table 2 would look like the following.

```
"Flood"
structure(list(OBS = c(1, 2, 3, 4, ..., 64, 65, 66),
    HYEAR = c(1932, 1933, 1934, 1935, ..., 1995, 1996, 1997),
    USDMG = c(0.1212, 0.4387, 0.1168, 1.4177, ..., 5.1108, 5.9774, 8.3576),
    DMGPC = c(0.9708, 3.4934, 0.9242, 11.1411, ..., 19.4504, 22.541,
    31.2275),
    LOSSPW = c(36.73, 143.26, 39.04, 461.27, ..., 235.34, 269.62, 367.34)),
```

```
.Names = c("OBS", "HYEAR", "USDMG", "DMGPC", "LOSSPW"),
class = "data.frame", row.names = c("1", "2", "3", "4", ..., "64",
"65", "66"))
```

To read in a data set from a file anywhere on your computer for use with `in2extRemes`, do the following.

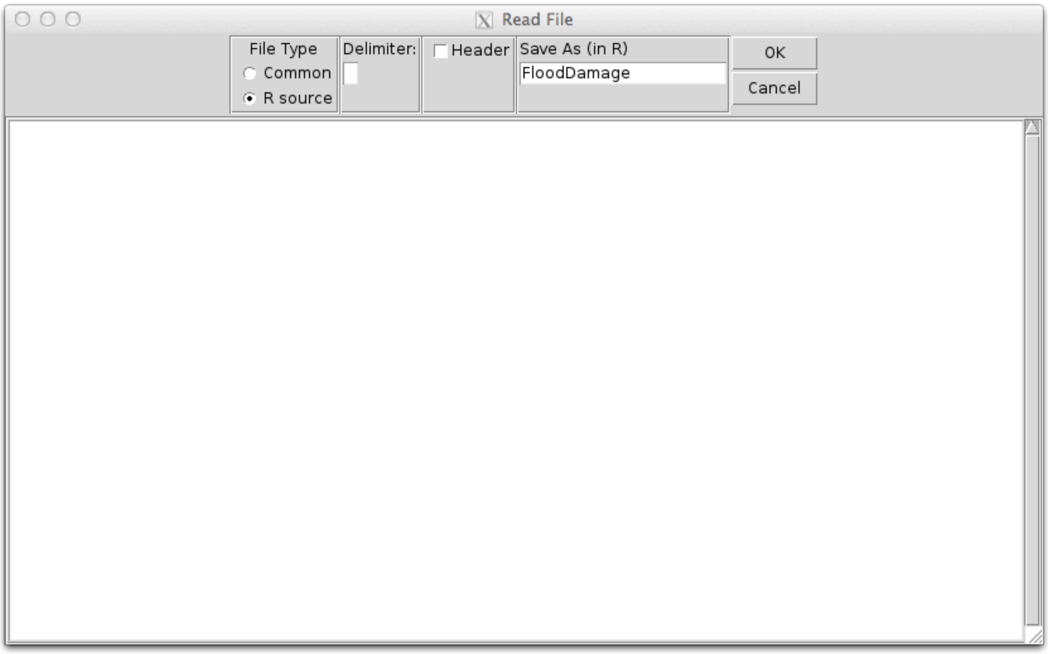
File > **Read Data**
Browse for your data using the window that opens, and double click on it (or single click > **Open**).
Select:
One of the **Common** or **R source** radio buttons depending on the file type.
Enter a delimiter (common type only) if other than spaces (e.g., enter a comma if it is comma delimited).
If there is a one-line header naming the columns of the data, then click the **Header** check button.
Enter a name in **Save As (in R)** > **OK**.

Note that any header information must contain only column names. Any other header text prior to the data should be removed before reading into `in2extRemes`. Alternatively, the data can be read into R and converted to the correct type (see section 5.2). Figure 9 shows an example of the window that appears after I browse and select the file you want to open. In this case, I have chosen an R source file that contains the flood damage data partially shown in Table 2, and I have chosen to call it `FloodDamage`. When I click **OK**, summary information is displayed in my R session along with a message about having saved the workspace.

5.2 Loading Data from the R Workspace

In some cases, it may be beneficial to first read data into R (or simulate data in a way not covered by the limited approaches availabel with `in2extRemes`). In order to make such data visible to the `in2extremes` windows, they must be converted into a certain format. This can be accomplished using the `as.in2extRemesDataObject` command. For example, we will make use of several example data sets available with the package, `extRemes`. The following commands will load and convert each data set into a format that is readable by `in2extRemes` windows. Note that the name to the left of the arrows (`<-`) is the name that will show up in the **Data Object** list boxes. See their help files for more information on any of them (e.g.,

Figure 9: Example of the dialogue window that opens after **File** > **Read Data** > selecting a dataset from the browsing window. In this case, an R source file containing the flood damage data has been selected, which is partially displayed in Table 2.



?Tphap).

```
# US flood damage data.
data(Flood)
FloodDamage <- as.in2extRemesDataObject(Flood)

# Fort Collins precipitation data.
data(Fort)
FortCollinsPrecip <- as.in2extRemesDataObject(Fort)

# Phoenix airport temperature data.
data(Tphap)
Phx <- as.in2extRemesDataObject(Tphap)

# Hurricane damage.
data(damage)
HurricaneDamage <- as.in2extRemesDataObject(damage)

# Port Jervis winter temperature data.
data(PORTw)
PORTwinter <- as.in2extRemesDataObject(PORTw)

# Number of Hurricanes per year from 1925 to 1995.
data(Rsum)
NumberOfHurricanes <- as.in2extRemesDataObject(Rsum)
```

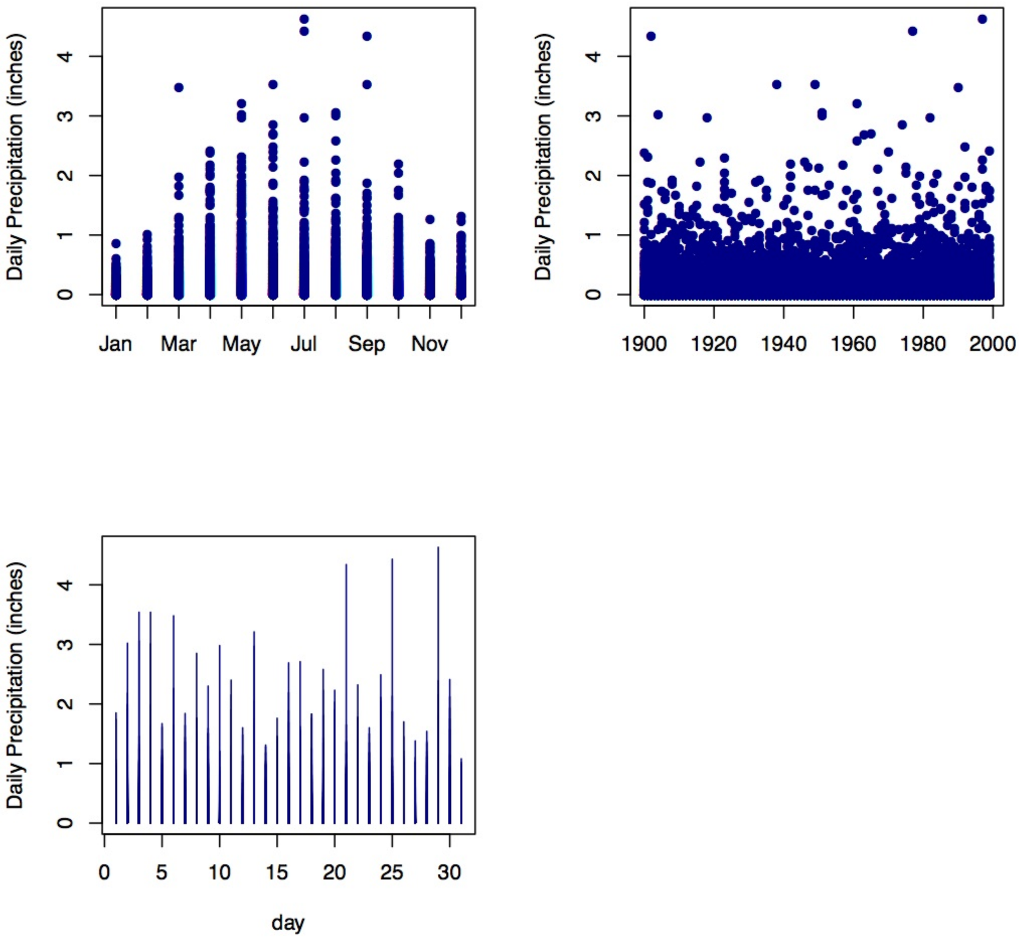
5.3 Example Data sets from `extRemes`

In section 5.2, several example data sets are loaded and set up so that they can be seen by `in2extRemes` windows. The data can be plotted as in the example in section 4, but appendix C shows the command-line R code for making the plots in this section, which are more sophisticated than what can be managed through the `in2extRemes` interface. For reference, we show plots for some of these datasets here.

Figure 10 plots Fort Collins precipitation data against month (top left panel), year (top right panel) and day (lower left panel). It is clear that the data vary seasonally, but there is no clear trend over time.

Phoenix minimum temperature (deg F) at Sky Harbor airport is shown against year in Figure 11 (top left panel). A clear upward trend over time is evident. In

Figure 10: Daily precipitation at a rain gauge in Fort Collins, Colorado, U.S.A. from 1900 to 1999. Panels show precipitation against month (top left), year (top right) and day (bottom left).



the upper right corner of the figure is estimated economic damage in billions of US dollars. No clear trend is present, but an extreme value, much larger than any other in this example, at the very beginning of the record is apparent.[2] Finally, total economic damage (billions USD) from floods is displayed in the lower left panel. The plot demonstrates increased variability with time, as well as larger extremes. Also shown are the number of hurricanes per year (lower right). Note that some years have no hurricanes, while the number otherwise varies from year-to-year.

Figure 12 shows the maximum winter temperature data example from the data object we assigned to `PORTwinter` in section 5.2. No obvious trend through time is apparent (left panel), but at least a weak positive association with the Atlantic Oscillation (AO) index is suggested (right panel).

5.4 Simulating Data

It is also possible to simulate data from a GEV or GP df from the `in2extRemes` windows (see also `?revd`). To simulate an iid sample from the GEV df, do the following.

File > **Simulate Data** > **Generalized Extreme Value (GEV)**
Enter parameter values as desired (see section 3 and Figure 6 for information about these parameters).
Enter a name for the simulated data > **Generate**

Figure 13 demonstrates simulating GEV distributed data using the default values and assigning the results to an object called `gevsim1`. Once data have been simulated, a plot of the simulated data will appear, and the assigned data will be available for use in the **Data Object** list boxes.

Simple linear trends may also be incorporated into the location (GEV df) or scale (GP df) parameters by entering a number other than zero (default) in the **Trend** entry box.

6 Analyzing Block Maxima

In section 5.2, daily precipitation (inches) data for Fort Collins, Colorado, U.S.A. was loaded and converted to an object readable by `in2extremes`, and given the name `FortCollinsPrecip`. See Katz et al. (2002) and `?Fort` for more information on these data.

In order to fit the GEV df to these data, we first need to find their annual maxima (we could choose blocks other than annual blocks, but we want them to

[2] This value is the (third) 1926 hurricane (*The blow that broke the boom*).

Figure 11: Phoenix summer (July–August) minimum temperature (deg F) 1948 to 1990 at Sky Harbor airport (top left). Economic damage from hurricanes (billions USD, top right). Total economic damage from floods (billions USD, lower left). Number of hurricanes per year (lower right).

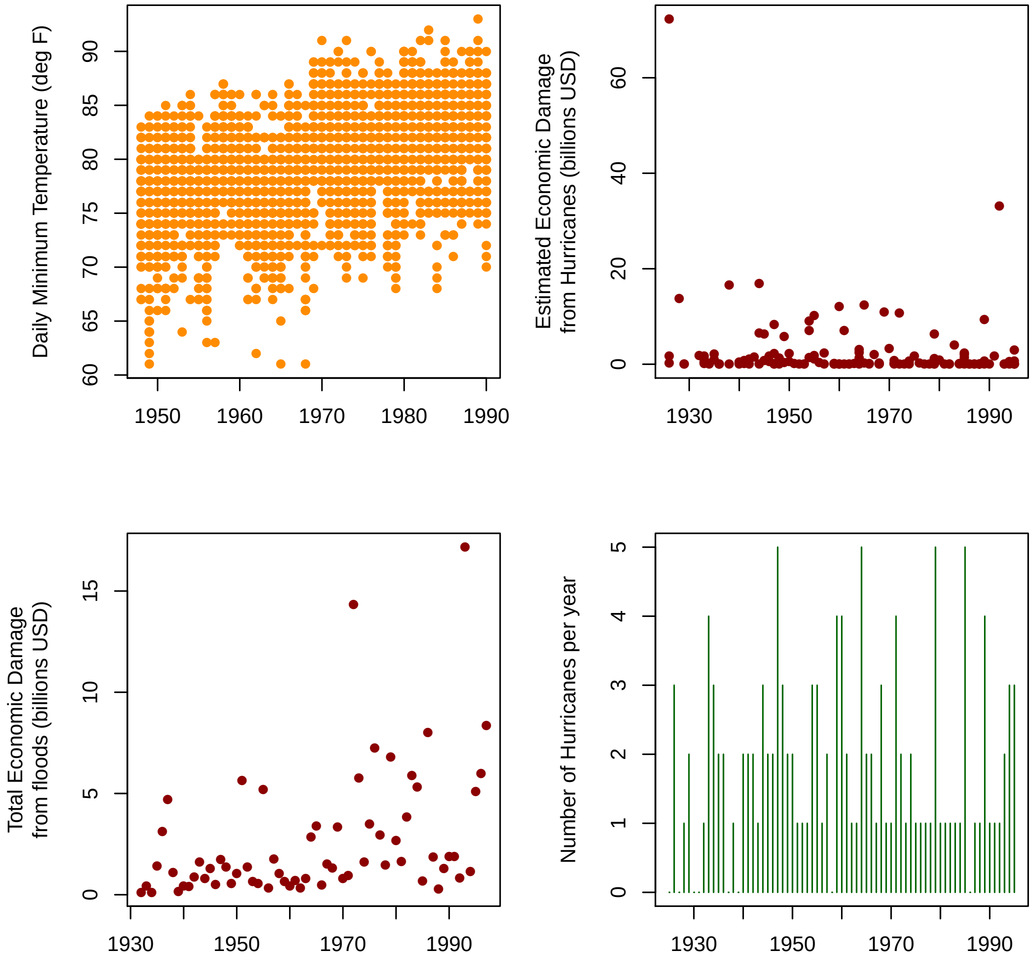


Figure 12: Winter maximum temperature (deg. C) at Port Jervis, New York 1927 to 1995. Plotted against year (left panel) and AO index (right panel).

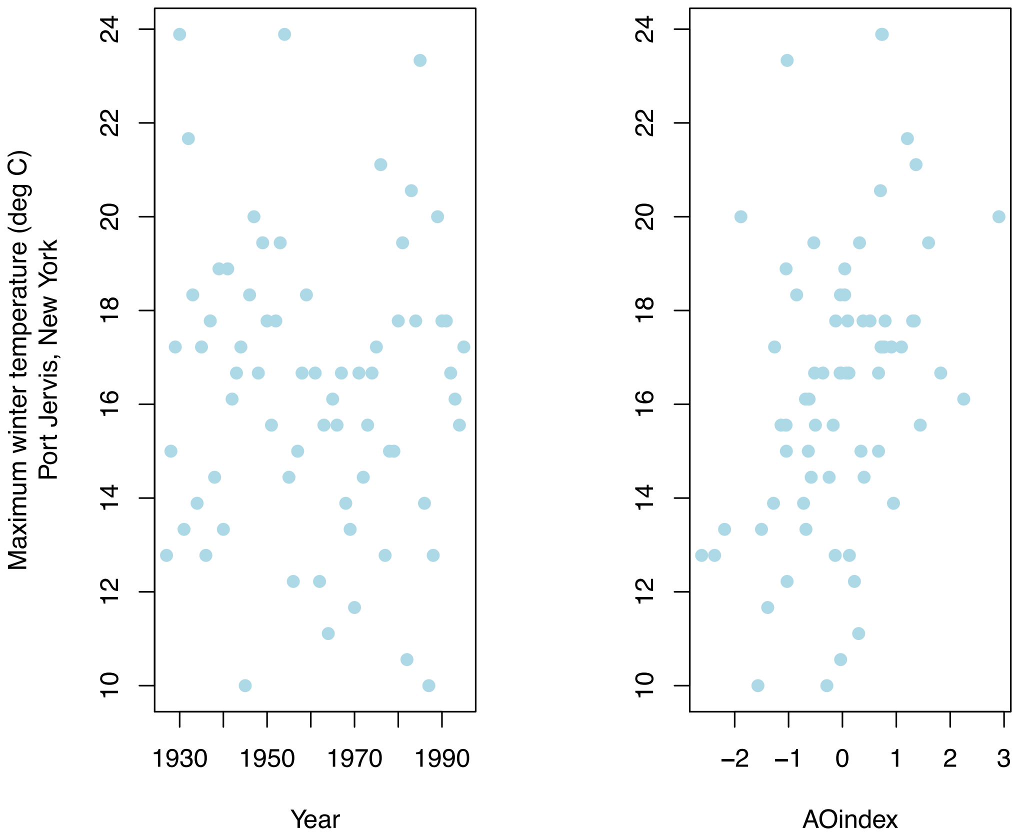


Figure 13: Example of the window that appears after **File** > **Simulate Data** > **Generalized Extreme Value (GEV)** and entering `gevsim1` in the **Save As** entry box.

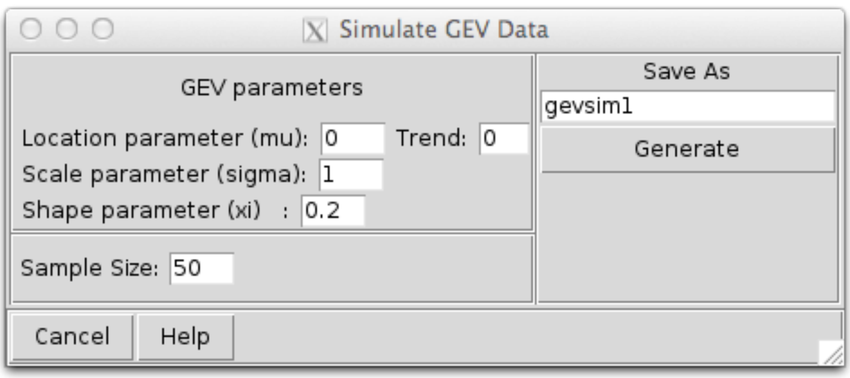
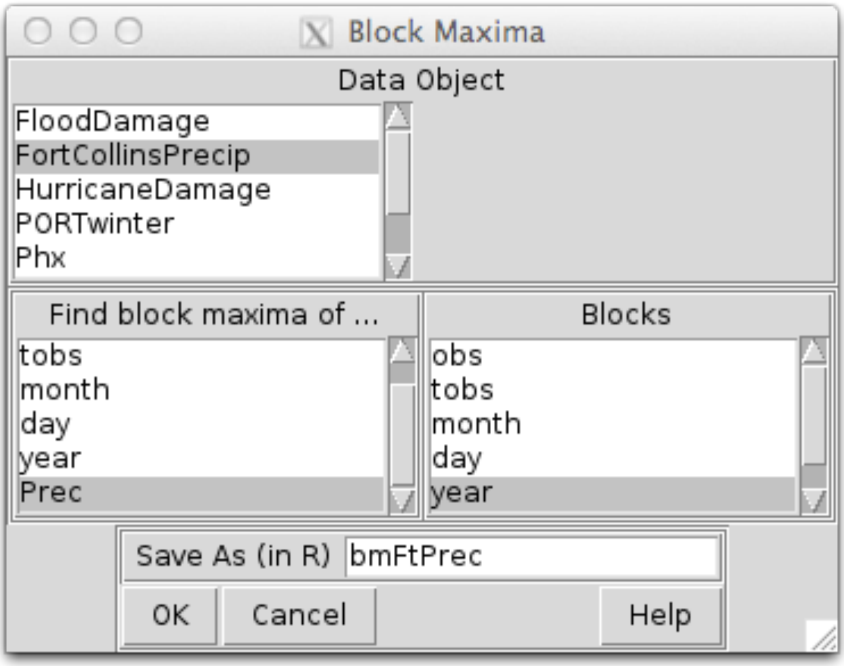


Figure 14: Example of finding block maxima using `in2extRemes`.



be long enough so that the assumptions for using the GEV will be sure to be met). The following instructions show how to find the block maxima for these data (see also Figure 14).

File > **Block Maxima**
Select:
Data Object > `FortCollinsPrecip`
Find block maxima of ... > `Prec`
Blocks > `year`
Save As (in R) > `bmFtPrec`
OK

We are now ready to begin analyzing these data using the GEV df.

6.1 Fitting the GEV df to Data

We fit the GEV df to block maxima data where the blocks are sufficiently long that the assumptions for using this distribution are met; typically block lengths are on a yearly time frame. In this section we will fit the GEV df to two data sets that were set up in section 5.2; one of which was converted to annual maximum data above.

We begin by fitting to annual maximum precipitation data from Fort Collins, called `bmFtPrec` in our **Data Object** list boxes.

Analyze > **Extreme Value Distributions**
Select:
Data Object > `bmFtPrec`
Response[3] > `Prec`
Model Type > `Generalized Extreme Value (GEV)`
Plot diagnostics > check
Response Units[4] > `inches`
OK

Figure 15 shows the window with the selections from above.

The resulting diagnostic plots are made and summary information about the fit is printed to the R session window. Figure 16 shows the plots that are produced. The upper left panel shows a qq-plot of the empirical data quantiles against those derived from the fitted GEV df. The plot is reasonably straight indicating that the assumptions for using the GEV df are met at least to a good approximation. The apparent departure from the straight line for the most extreme values is of a fairly typical magnitude when performing extreme value analysis, and it is important to remember that considerable uncertainty exists for these values so that such deviations at the extreme end are not necessarily indicative of ill-met assumptions.

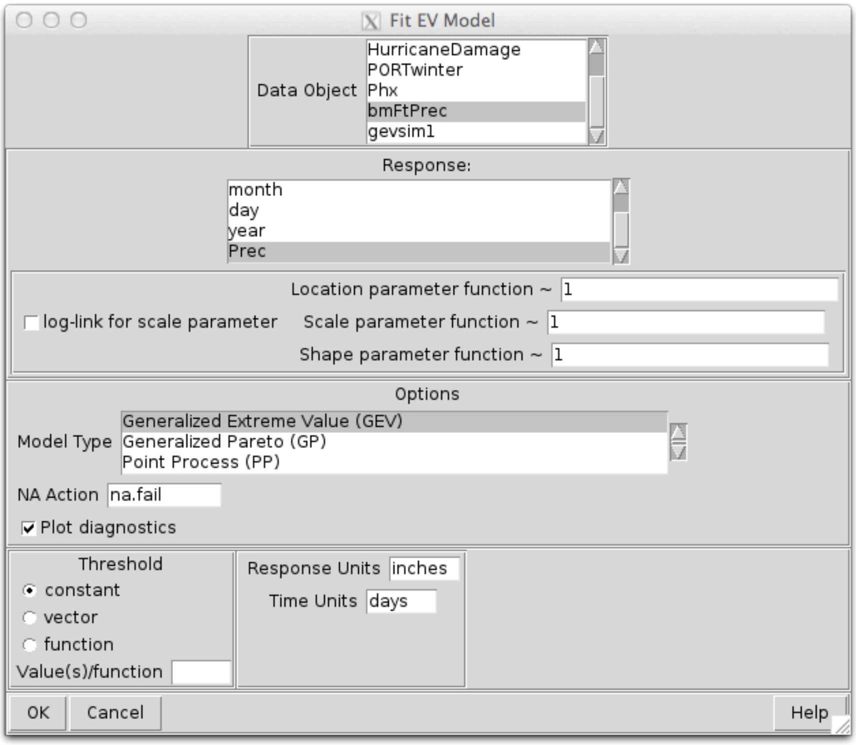
The qq-plot in the upper right panel is similar to the qq-plot in the upper left panel. It shows the qq-plot for randomly generated data from the fitted GEV df against the empirical data quantiles along with 95% confidence bands, a 1-1 line and a fitted regression line. Note that this plot may appear slightly different from the one you obtain, and indeed, from each subsequent time it is plotted because the data are randomly generated anew.

The density plot in the lower left panel shows good agreement between the empirical density (solid black line) and that of the fitted GEV df (dashed dark blue line). Finally, the return level plot (lower right panel) is on the log scale so that the heavy-tail case is concave, the bounded-tail case convex, and the light-tail case linear. The empirical return levels match well with those from the fitted df. For the GEV df, the return levels are the same as the quantiles so that this plot is

[3]The response is simply the data vector to which the GEV df will be fit. In this case, `Prec`, the annual maximum precipitation in Fort Collins, Colorado.

[4]The units entry box is optional, but entering the units will allow them to appear on certain plots.

Figure 15: Window showing the options selected for fitting the GEV df to the annual maximum Fort Collins precipitation as described in the text.



similar to the qq-plot in the upper left panel, but where the two sets of quantiles are plotted against the return period (on a log scale) instead of each other. The apparent concave shape shape of the return level plot is indicative of a heavy-tail GEV df.[5]

It is possible to plot the diagnostic plots in Figure 16 without re-fitting the GEV df to the data in the following way.

Plot > **Fit Diagnostics**
Select:
Data Object > `bmFtPrec`
Select a fit > `fit1`

From above, we see that the fitted object is available from `bmFitPrec` under the name `fit1`. Each time a new fit is made, a new item called `fitX` where `X` is the number of the fitted object. If you lose track of which object is which, you can always use: **Analyze** > **Fit Summary**, then select the data object and fit number, then **OK**. The summary of the fit will display in the R session window, which will include information about the fitted object.

Having verified that the diagnostic plots support the assumptions for using the GEV df, we can now look at the summary results. The summary is printed in the R session window automatically upon executing the fit, but you can always conjure it up again as in the paragraph above. The summary shows the actual function call used to implement the fit.[6]

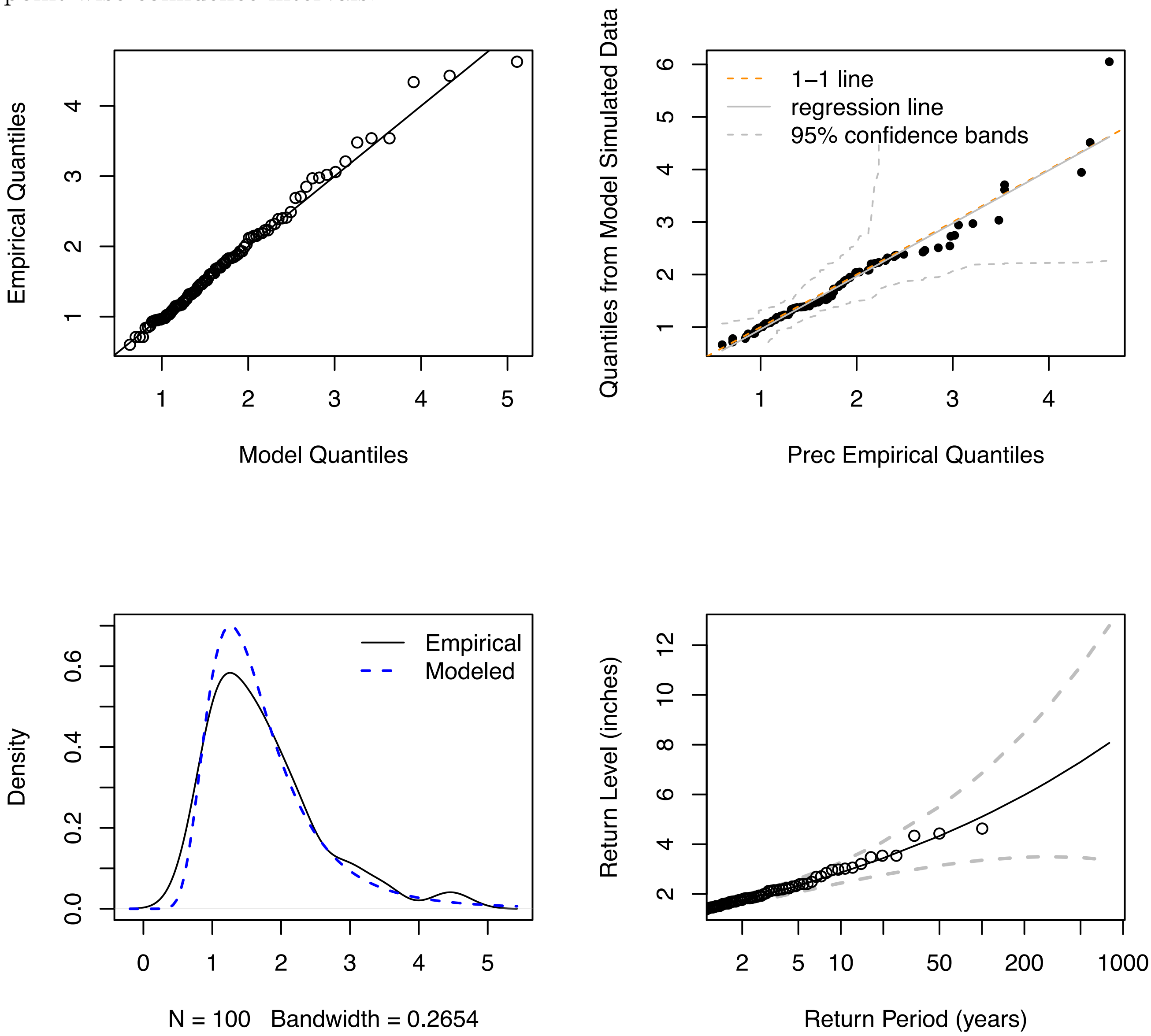
The summary shows the estimated parameters as $\mu \approx 1.35$, $\sigma \approx 0.53$ and $\xi \approx 0.17$. Also shown are the negative log-likelihood value (≈ 104.96), estimated parameter standard errors, parameter covariance matrix, as well as the AIC and BIC for the optimized fit. We will see how to obtain confidence intervals for the parameter estimates and return levels in section 6.2.

Another data example that was set up for use with `in2extRemes` in section 5.2 and assigned the name `PORTwinter` already gives annual maxima (see Thompson

[5] Although the return level plot's abscissa is labelled *Return Period*, it is actually an (asymptotic) approximation that is not very accurate at lower values; e.g., empirical return levels shown to be below one year are common. It is easy to plot the exact return periods, but at the expense of losing the exact linearity when the shape parameter is identically zero, which is arguably more important. It is common practice to label the axis as *Return Period* (personal communication, Alec G. Stephenson, 2013). Indeed, such short return periods are not generally of interest for EVA.

[6] It is possible to verify in the log file that the call listed at the beginning of the summary is the final function call. Note that the `data` argument is always `xdat`. In the log file, you will see the commands `dd <- get( "bmFtPrec")` and `xdat <- dd[["data"]]`, which verifies that `xdat` is the correct data frame.

Figure 16: Diagnostic plots from fitting the GEV df to annual maximum precipitation (inches) data in Fort Collins, Colorado. Plots are qq-plot of empirical data quantiles against fitted GEV df quantiles (top left panel), qq-plot of randomly generated data from the fitted GEV df against the empirical data quantiles with 95% confidence bands (top right panel), empirical density of observed annual maxima (solid black line) with fitted GEV df density (dashed dark blue line) superimposed (lower left panel), and return level plot (log scale) with 95% normal approximation point-wise confidence intervals.



and Wallace (1998), Wettstein and Mearns (2002) and `?PORTw` for more details about these data). Therefore, we do not need to obtain the annual maxima as we did for the precipitation data. To fit the GV df to these data:

Analyze > **Extreme Value Distributions**
Select:
Data Object > `PORTwinter`
Response > `TMX1`
Model Type > `Generalized Extreme Value (GEV)`
Plot diagnostics > check
Response Units > `deg C`
OK

Diagnostic plots from fitting the GEV df to these maximum winter temperature data are shown in Figure 17. The assumptions appear reasonable, although some curvature in the qq-plot (upper left panel) is evident. Notice that unlike the previous fit (Figure 16), the return level plot is convex because of the fitted model's being the upper bounded-tail case ($\xi \approx -0.22$).

6.2 Confidence Intervals for GEV Parameters and Return Levels

To obtain normal approximation confidence intervals for the parameter estimates and the 100-year return level[7] for the GEV df fit to the annual maximum Fort Collins precipitation data, do the following.

Analyze > **Parameter Confidence Intervals**
Select:
Data Object > `bmFtPrec`
Select a fit > `fit1`[8]
OK

Similar to the fit summary displayed in section 6.1, the original function call is displayed, followed by a matrix showing the parameter estimates and their 95% confidence intervals, as well as a message indicating that they were obtained via

[7] Any return level can be found by entering the desired return period into the **return period** entry box, but for brevity, here, we simply use the default value of 100.

[8] Here, it is assumed that the fitted object is called `fit1`, but select the appropriate name for your model object.

Figure 17: Diagnostic plots from fitting the GEV df to maximum winter temperature (deg C) at Port Jervis, New York (1927–1995).

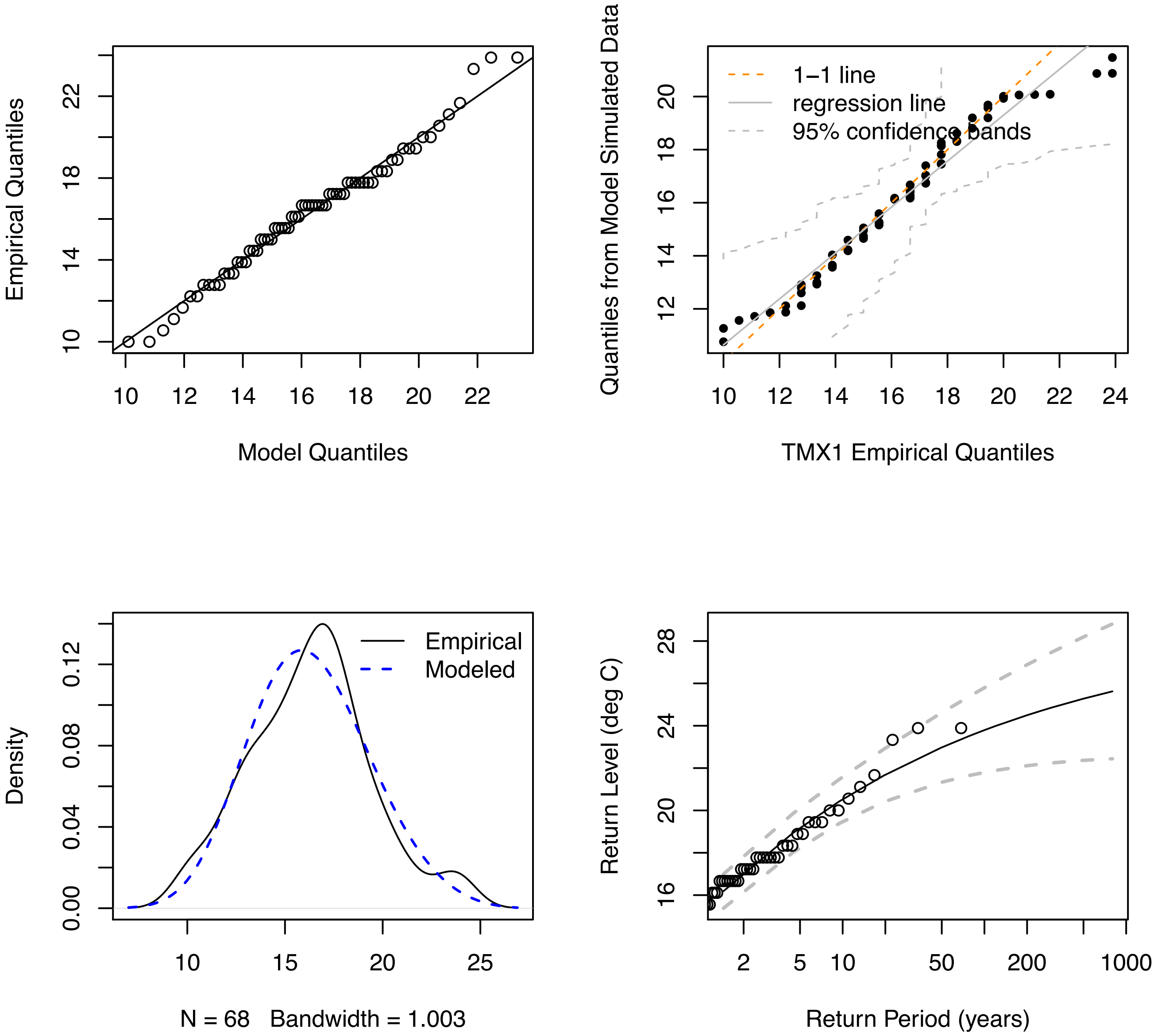


Table 3: 95% confidence intervals based on the normal approximation for the GEV df fit to annual maximum precipitation (inches) in Fort Collins, Colorado (values rounded to two decimal places).

	95% lower CI	Estimate	95% upper CI
location	1.23	1.35	1.47
scale	0.44	0.53	0.63
shape	-0.01	0.17	0.35
100-year return level	3.35	5.10	6.84

the normal approximation. Following these is the estimated 100-year return level along with its normal approximation 95% confidence interval. The printed output is reproduced in Table 3 as a check. Note that zero falls within the 95% confidence interval for the shape parameter. Therefore, the null hypothesis that the data follow a Gumbel df cannot be rejected. Because $\xi = 0$ is a single point in a continuous parameter space, there is zero probability of estimating a shape parameter that is exactly zero. It is possible to test the hypothesis that $\xi = 0$ against the alternatives (e.g., by checking if zero falls within the confidence limits), but it is arguably preferable to always allow the shape parameter to be nonzero even if a test supports retaining the null hypothesis that $\xi = 0$.

Although situated about 80 km south of Fort Collins, Boulder, Colorado, which has a very similar climate, received over 9 inches of rainfall in 24 hours in early September 2013, which is well above the 100-year return level upper confidence bound shown in Table 3. The event was part of a wide-spread heavy precipitation event that reached as far north as Fort Collins and as far south as Colorado Springs, Colorado (which lies approximately 80 km south of Boulder).

The confidence intervals obtained in Table 3 were calculated using the normal approximation method, which is fast and easy to compute. However, return level estimates with longer ranges (and sometimes those for the shape parameter) often have a more skewed distribution, making these intervals less accurate in general. Two other methods for finding confidence intervals are available with `extRemes` and `in2extremes` for MLE's: parametric bootstrap and profile likelihood.[9]

If `parametric bootstrap` is selected from the **Method** list box, then the

[9]These methods are not available for the models with covariates in the parameter estimates discussed in sections 6.3, 8.4 and 9.3, except for parameter confidence intervals in the case of the profile-likelihood method.

number of bootstrap replicates can be changed as desired. Both the bootstrap and profile-likelihood methods are iterative, and may take time to run. First, we demonstrate the parametric bootstrap method.

Analyze > **Parameter Confidence Intervals**
Select:
Data Object > `bmFtPrec`
Select a fit > `fit1`
Method > `parametric bootstrap`
OK

Results will vary some because the method is based on repeatedly drawing random samples from the fitted df, but should not vary too widely and for one instance, 95% bootstrap confidence interval for the 100-year return level was estimated to be about (3.77, 7.13). Notice that the MLE is not in the exact center of the interval and that both bounds are shifted towards higher values as compared to those based on the normal approximation. For details about the parametric bootstrap, see appendix B.0.4.

The following example demonstrates how to use the profile-likelihood approach for the 100-year return level. For this approach, it is generally advisable to plot the profile likelihood to make sure that the resulting confidence intervals are reasonable. The approach can be used for either return levels or parameters, but note that `in2extRemes` only allows for changing the search limits for return levels (for which this approach is arguably more necessary) and not individual parameter estimates. The software will attempt to find an appropriate search range, but will not always succeed; leading to intervals that are clearly incorrect (e.g., the estimated parameter does not fall inside the limits). In such a case, the search range can be changed in the command-line code of `extRemes`, but not from the `in2extRemes` window.

Analyze > **Parameter Confidence Intervals**
Select:
Data Object > `bmFtPrec`
Select a fit > `fit1`
Method > `profile likelihood`
Parameter > uncheck
OK

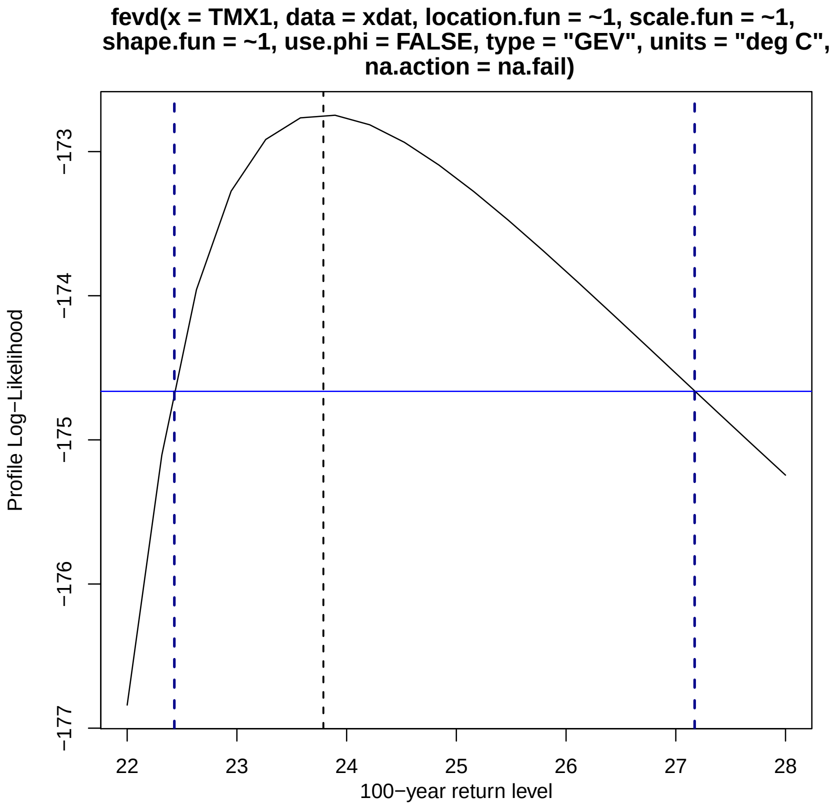
The resulting limits are estimated to be (3.94, 7.11). It is difficult to obtain a graph for the profile likelihood for this particular fit, but clearly the three vertical dashed lines are present, which is most important. Narrowing the plotting limits (**Lower Limit** and **Upper Limit** entry boxes) as well as changing the **Number of points at which to calculate the profile likelihood** value can refine the approximation, but for these data, it does not drastically change the limits. The limits obtained by the profile-likelihood method are in better agreement with those obtained from the parametric bootstrap than those from the normal approximation, which is less appropriate for such a long return period. In order to provide an example that yields a more legible graph for the profile likelihood, we turn to the Port Jervis maximum winter temperature example.

Figure 18 shows the profile likelihood for the 100-year return level from the GEV df fit to the Port Jervis maximum winter temperature example, which is much easier to visualize than that for the Fort Collins precipitation example. The horizontal line in the figure is at the maximum profile likelihood value minus one half of the 0.95 quantile of the χ^2_1 df. Per the interval given by Eq (13) in appendix B.0.3, all parameter values associated with likelihoods that differ from the maximum likelihood for the parameter in question by less than one half of the $\chi^2_{1-\alpha,k=1}$ quantile are within the $1-\alpha$ confidence region. Therefore, all return levels associated with profile likelihood values above the lower horizontal line in the figure are contained in the 95% confidence region, and all the return values with profile likelihood values below the line are outside the 95% region. Thus, the approximate confidence intervals are obtained by finding where the profile likelihood crosses this horizontal line, and the blue vertical dashed lines show where the estimated confidence intervals were found. The black dashed line in the middle shows the MLE, in this case ≈ 23.79 degrees centigrade for the 100-year return level of annual maximum winter temperature.

6.3 Fitting a Non-Stationary GEV df to Data

In section 6.1, the GEV df was fit to winter temperature maxima at Port Jervis, New York. It is believed, however, that this series should vary with the Atlantic

Figure 18: Example of profile likelihood graph for the 100-year return level for the GEV df fit to annual maximum winter temperatures at Port Jervis, New York, U.S.A.



Oscillation (e.g. Thompson and Wallace, 1998). We can analyze whether or not such non-stationarity exists for the extremes by fitting a GEV model with the Atlantic Oscillation (AO) index as a covariate in one or more of its parameters. We can then perform the likelihood-ratio test on this model against the model we fit in section 6.1 without covariates. First, we try to fit the GEV df with AO index (given in the example data set as `AOindex`) as a linear covariate in the location parameter.

Analyze > **Extreme Value Distributions**
Data Object > `PORTwinter`
Response > `TMX1`
Location parameter function ~ > Replace `1` with `AOindex`
Model Type > `Generalized Extreme Value (GEV)`
Plot diagnostics > check
Response Units > `deg C`
OK

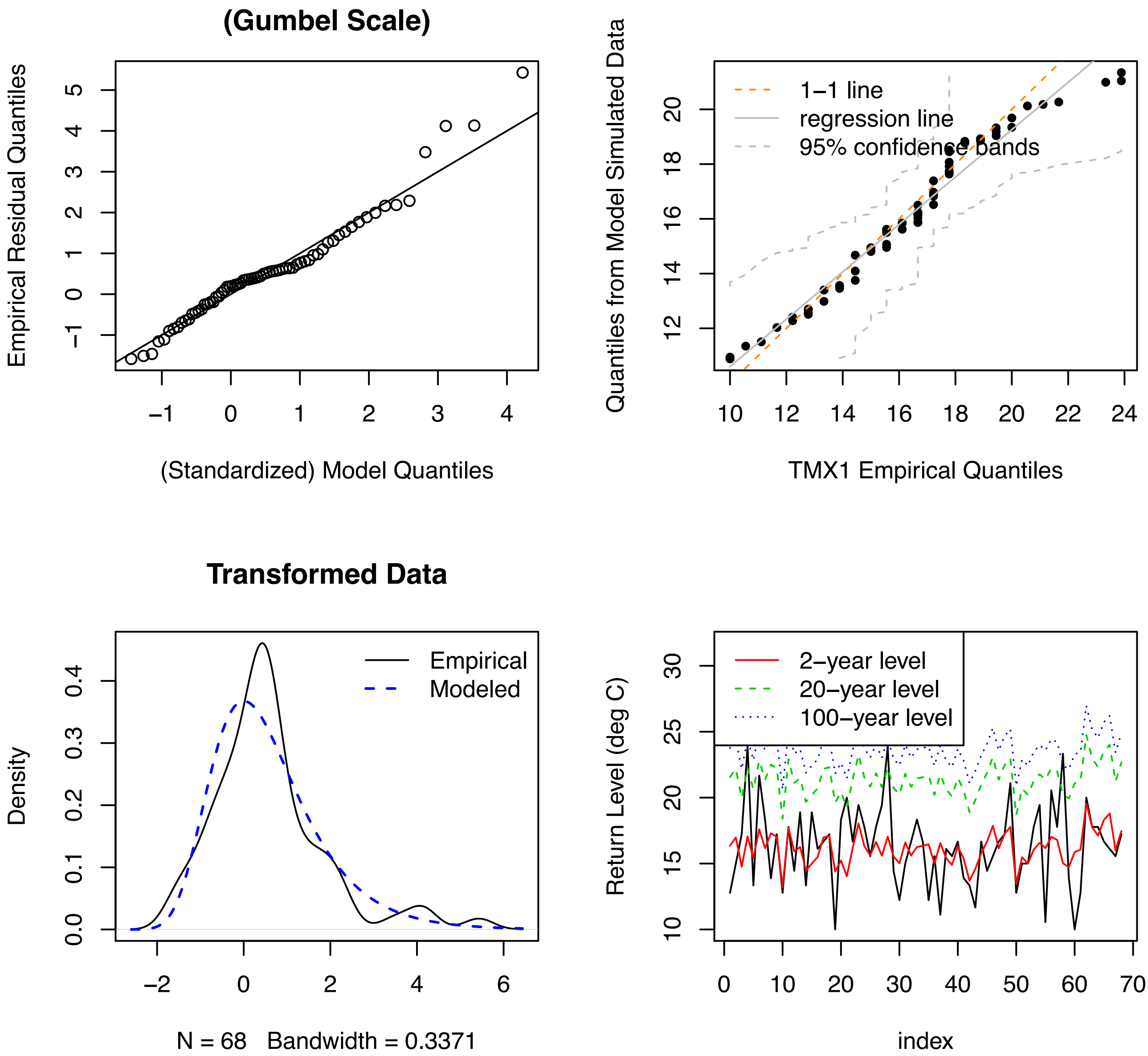
Figure 19 shows the resulting diagnostic plots. Notice that the plots shown are similar to those for the non-stationary GEV df fits from section 6.1. However, important differences exist. For example, the qq-plot in the upper left panel and the density plot in the lower left panel are for data that have been transformed to the Gumbel scale, which results in a stationary df. The second qq-plot (upper right panel) is effectively the same as before except now the data have been simulated from a non-stationary GEV df. The most striking difference comes from the return level plot, which is now a plot of the block maxima with effective return levels that show how the GEV df varies with the AO index. Note that the 2-year return level corresponds to the median of the GEV df. Generally, these diagnostic plots suggest that the assumptions for the model are reasonable, however, some curvature is still apparent in the qq-plots.

To perform the likelihood-ratio test to determine if inclusion of the AO index in the location parameter is statistically significant, we use the following.

Analyze > **Likelihood-ratio test**
Data Object > `PORTwinter`
Select base fit (M0) > `fit1`
Select comparison fit (M1) > `fit2`

The test gives a deviance statistic (likelihood-ratio statistic) of 11.89, which is larger than the associated χ_1^2 0.95 quantile of 3.84, and the associated p-value is

Figure 19: Diagnostic plots for non-stationary GEV df fit to Port Jervis, New York maximum winter temperature (deg C) data example.



less than 0.001. Therefore, these results provide strong support for including the AO index as a covariate in the location parameter.

One may want to also check if the scale parameter should also vary with AO index.

Analyze > **Extreme Value Distributions**
Data Object > `PORTwinter`
Response > `TMX1`
Location parameter function ~ > Replace `1` with `AOindex`
Scale parameter function ~ > Replace `1` with `AOindex`
log-link for scale parameter > check
Model Type > `Generalized Extreme Value (GEV)`
Plot diagnostics > check
Response Units > `deg C`
OK

The above fits the model with a location parameter that varies linearly with AO index and a scale parameter that varies exponentially with it. Performing the likelihood-ratio test of this model against the model without covariates gives a statistically significant result, but when comparing the model to that with a linear covariate in the location parameter only is not significant. Therefore, one can conclude that there is not strong support for varying the parameters as a function of the AO index.

As another example of the use of covariates, it is checked if the location parameter should depend not only on the AO index, but on the year as well.

Analyze > **Extreme Value Distributions**
Data Object > `PORTwinter`
Response > `TMX1`
Location parameter function ~ > Replace `1` with `AOindex + Year`
Model Type > `Generalized Extreme Value (GEV)`
Plot diagnostics > check
Response Units > `deg C`
OK

Note that various R formulas can be utilized in the **parameter function ~** entry boxes. See `?formula` for details about how R formulas work.

Applying the likelihood-ratio test to compare the fit with both AO index and year in the model against that with only AO index shows that the p-value is about

0.25, so we fail to reject the null hypothesis that only AO index should be included as a covariate for the location parameter. The AIC and BIC (appendix B.0.2) criterion are also both larger for the model with both AO index and year; supporting the conclusion of the likelihood-ratio test.

7 Frequency of Extremes

Often it is of interest to look at the frequency of extreme event occurrences. As the event becomes more rare, the occurrence of events should approach a Poisson process, so that the relative frequency of event occurrence over a given time interval approaches a Poisson distribution. For fitting the Poisson distribution to extreme event occurrences with covariates, `in2extRemes` calls upon the `glm` function from the `stats` package, which is automatically available with R (R Core Team, 2013) (without any need to separately install or load the package). Otherwise, without covariates, there is only one parameter to estimate (the rate parameter or mean), the MLE of which is simply the average number of events over the entire record.

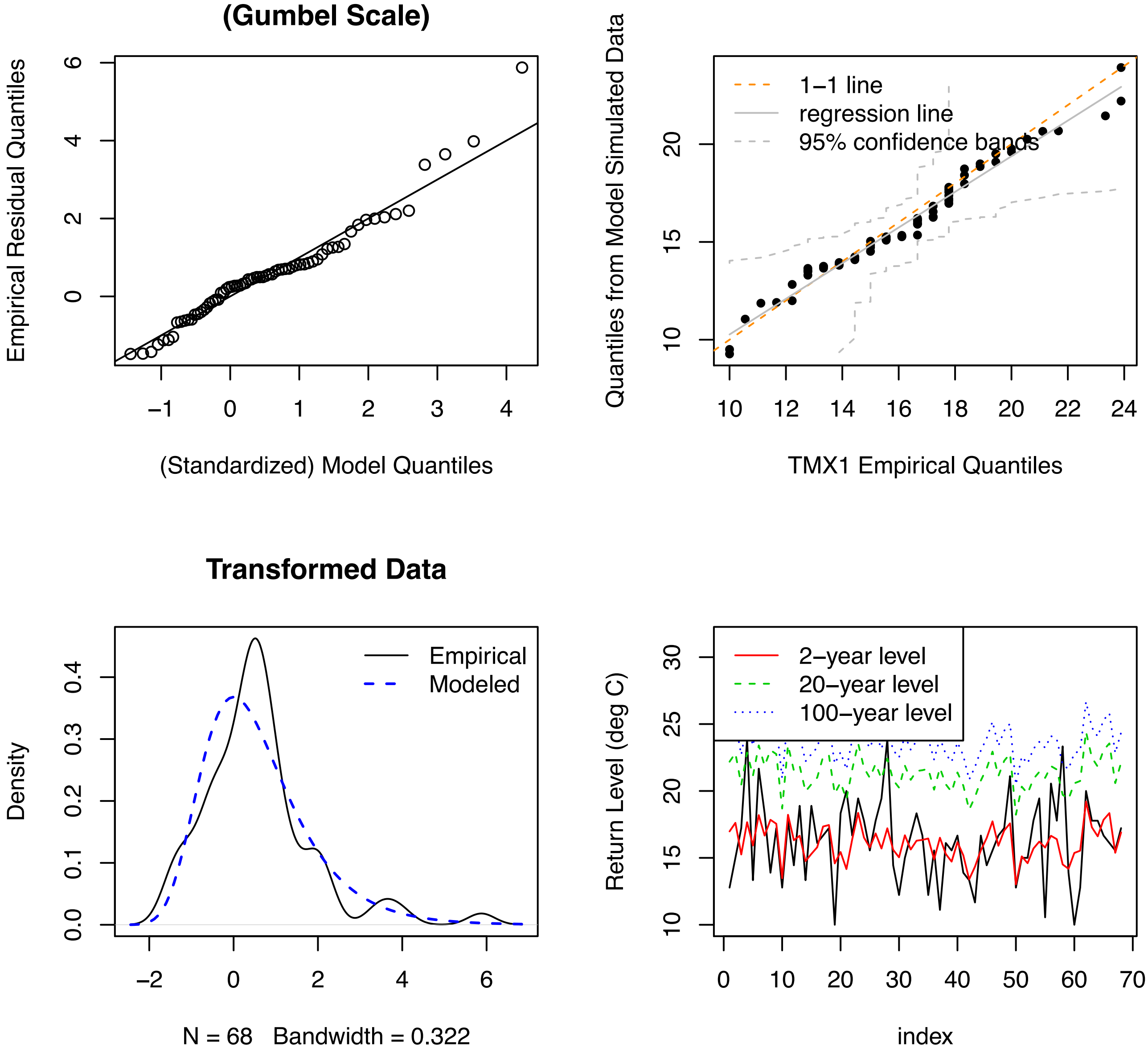
In section 5.2 an example data set was loaded that gives the numbers of hurricanes per year and called `NumberOfHurricanes` (Figure 11 lower right panel). The simple, no covariate, fit is performed by the following.

Analyze > **Poisson Distribution**
Select:
Data Object > `NumberOfHurricanes`
Response > `Ct`
OK

The main result from the above is that the estimated rate parameter, $\hat{\lambda}$, is approximately 1.82, indicating that, on average, nearly two hurricanes caused damage per year from 1925 to 1995. A property of the Poisson distribution is that the mean and variance are the same and are equal to the rate parameter. As per Katz (2002), the estimated variance is shown to be 1.752, which is only slightly less than that of the estimated mean (1.817). The chi-square statistic (with 70 degrees of freedom) is shown to be 67.49 with associated p-value of 0.563 indicating that there is no significant difference in the mean and variance. Similar to the GEV distribution, it is often of interest to incorporate a covariate into the Poisson distribution. For example, it is of interest with these data to incorporate ENSO state as a covariate.

Analyze > **Poisson Distribution**

Figure 20: Diagnostics for GEV df fit to Port Jervis, New York winter maximum temperatures with AO index and year as covariates in the location parameter.



Select:
Data Object > `NumberOfHurricanes`
Response > `Ct`
Covariate (log link) > `EN`
Plot diagnostics > check
OK

The column `EN` for this data frame represents the ENSO state (-1 for La Niña events, 1 for for El Niño events, and 0 otherwise). A plot of the residuals is created if the **plot diagnostics** check button is checked, and can also be performed later using **Plot** > **Fit Diagnostics**. The fitted model is found to be $\ln \hat{\lambda} \approx 0.56 - 0.25 \cdot \texttt{EN}$. For fitting a Poisson regression model to data, a likelihood-ratio statistic is given in the main toolkit dialog, where the ratio is the null model (of no trend in the data) to the model with a trend (in this case, ENSO). Here the addition of ENSO as a covariate is significant at the 5% level (p-value $\approx$ 0.03) indicating that the inclusion of the ENSO term as a covariate is reasonable.

8 Analyzing Threshold Excesses

In this section, the extent by which a value x exceeds a high threshold, u, is modeled. That is, interest lies in the random variable $Y = X - u$, given that $X > u$. We call Y a threshold excess, or simply excess. EVT provides justification for fitting the GP df to excesses.

8.1 Threshold Selection

When fitting the GEV df to block maxima, it is important to choose blocks sufficiently long that the GEV provides a good approximation. However, they must be short enough that enough data remains to be modeled in order to avoid excesses variability in the parameter estimates. In the case of block maxima, it is common to take blocks to be a year or a season in length. Provided a sufficient number of years worth of data are available, determining an appropriate block length is not generally an issue.

For POT models, an analogous bias-variance trade-off is required, but the choice is usually of greater concern than for block maxima. In this case, a threshold, u should be chosen high enough for the EVD to provide a reasonable approximation. On the other hand, it should be low enough so that the estimated parameter variances are sufficiently small.

Many approaches to threshold selection have been proposed with two methods for diagnosing an appropriate threshold available with `in2extremes`. The first is to fit the GP df to the excesses for a range of thresholds, and then choose the lowest value of u such that the parameter estimates for fits with adjacent higher values of u being reasonably stable.[10] The second, often called the mean residual life plot (or mean excess plot), uses a result that the mean of the excesses over a high threshold should vary linearly with the threshold once a sufficiently high threshold is attained. We will demonstrate this approach for the GP df using the Fort Collins, Colorado daily precipitation example.

Plot > **Fit POT model to a range of thresholds**
Select:
Data Object > `FortCollinsPrecip`
Select Variable > `Prec`
Minimum Threshold > `0.2`
Maximum Threshold > `2.5`
Number of thresholds > `30`
OK

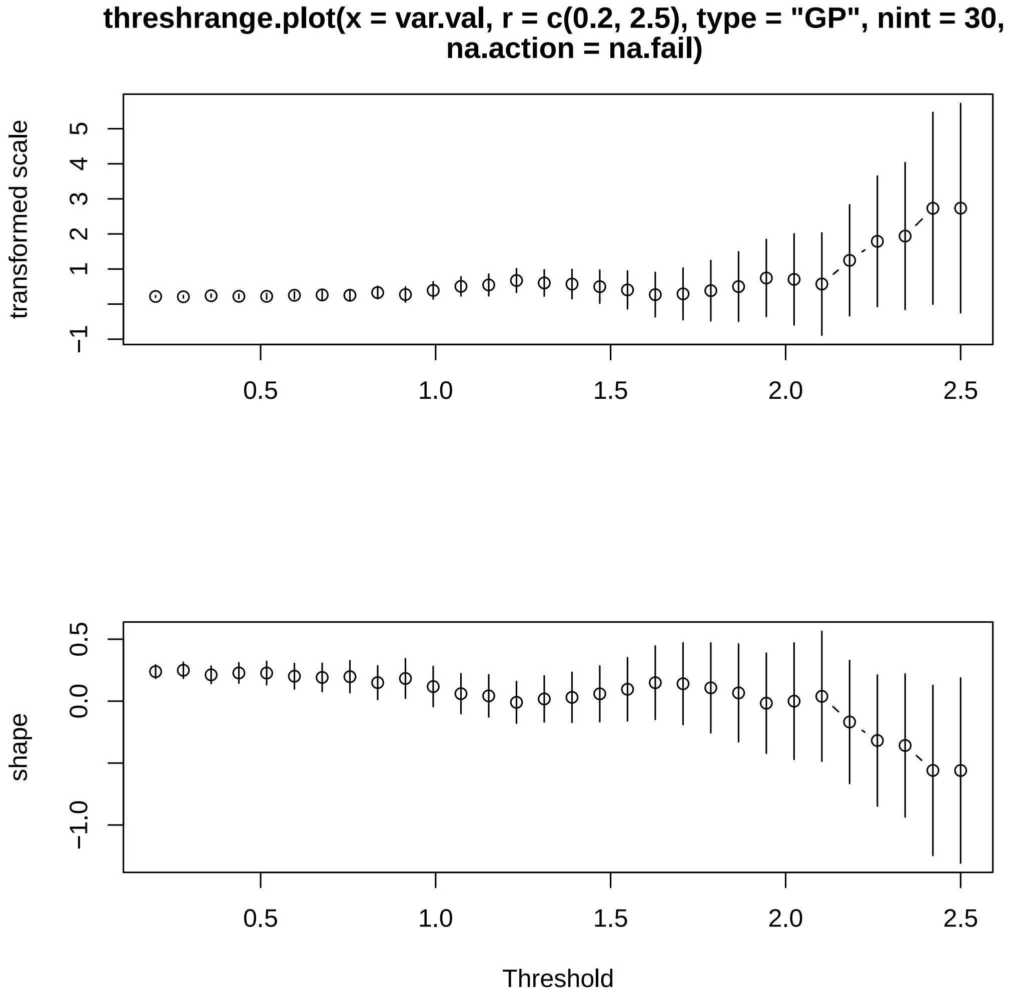
The above may take several moments as many fits, in this case 30, must be performed. The minimum and maximum threshold values, as well as the number of thresholds, are somewhat arbitrary decisions that should be made with the data in mind. If very little data were above 2.5 inches, then it would not be useful to try anything larger. Figure 21 shows the results for the above example. Because the parameter estimates do not appear to vary considerably for even the smallest values, we might choose a relatively low value (e.g., 0.395 inches) for now, although we will revisit this choice later when we fit the PP model to these data. The estimates begin to vary for the much higher thresholds, but the estimated values at lower thresholds easily fall within the 95% confidence intervals (vertical line segments).[11]

Next, we make the mean residual life plot for these same data. **Plot** > **Mean Residual Life Plot**
Select:
Data Object > `FortCollinsPrecip`
Select Variable > `Prec`

[10]Recall that the scale parameter for the GP df is a function of the threshold. Therefore, a transformed scale parameter is plotted so that it should be independent of the threshold.

[11]Similar plots created by `ismev` and used by `extRemes` versions < 2.0 show uncertainty using 1.6 times the estimated parameter standard error rather than 95% confidence intervals. Therefore, results may differ slightly.

Figure 21: GP df fit using a range of thresholds to the daily precipitation in Fort Collins data. Vertical line segments display the 95% normal approximation confidence intervals for each parameter estimate.



OK

The resulting plot from the above instructions is shown in Figure 22. It is not always easy to diagnose a good threshold choice from these plots, but the idea is to find the lowest threshold whereby a straight line could be drawn from that point to higher values and be within the uncertainty bounds (dashed gray lines).

Next, we will fit the PP model to a range of thresholds.

Plot > **Fit POT model to a range of thresholds**
Select:
Data Object > `FortCollinsPrecip`
Select Variable > `Prec`
Minimum Threshold > `0.3`
Maximum Threshold > `2`
Type of POT model > `PP`
Number of thresholds > `20`
OK

The resulting plots are shown in Figure 23. Once again, it appears that 0.395 inches will suffice as the threshold.

8.2 Fitting the Generalized Pareto df to Data

Figure 24 shows the GP df fit over a range of thresholds to the hurricane damage data (Figure 11, upper right panel). From these plots, it appears that a threshold of 6 billion USD will be a good choice. One issue for fitting the GP df to these data is that there are different numbers of hurricanes each year, and some years may not have a hurricane. Therefore, we need to estimate the number of hurricanes per year. It is not important as far estimating the parameters, but it will be important in terms of estimating return levels appropriately. Because there are 144 hurricanes that occurred over the 70-year span, one reasonable choice is to use 144/70, which is about 2.06 per year. This information is entered into the **Time Units** entry box in the form `2.06/year`.[12]

Analyze > **Extreme Value Distributions**
Select:
Data Object > `HurricaneDamage`

[12]The original data ranged from 1925 to 1995 with no events in 1925. Therefore, Katz (2002) treats the data as having a 71-year span.

Figure 22: Mean residual life plot for daily Fort Collins precipitation data. Dashed gray lines show the 95% confidence intervals.

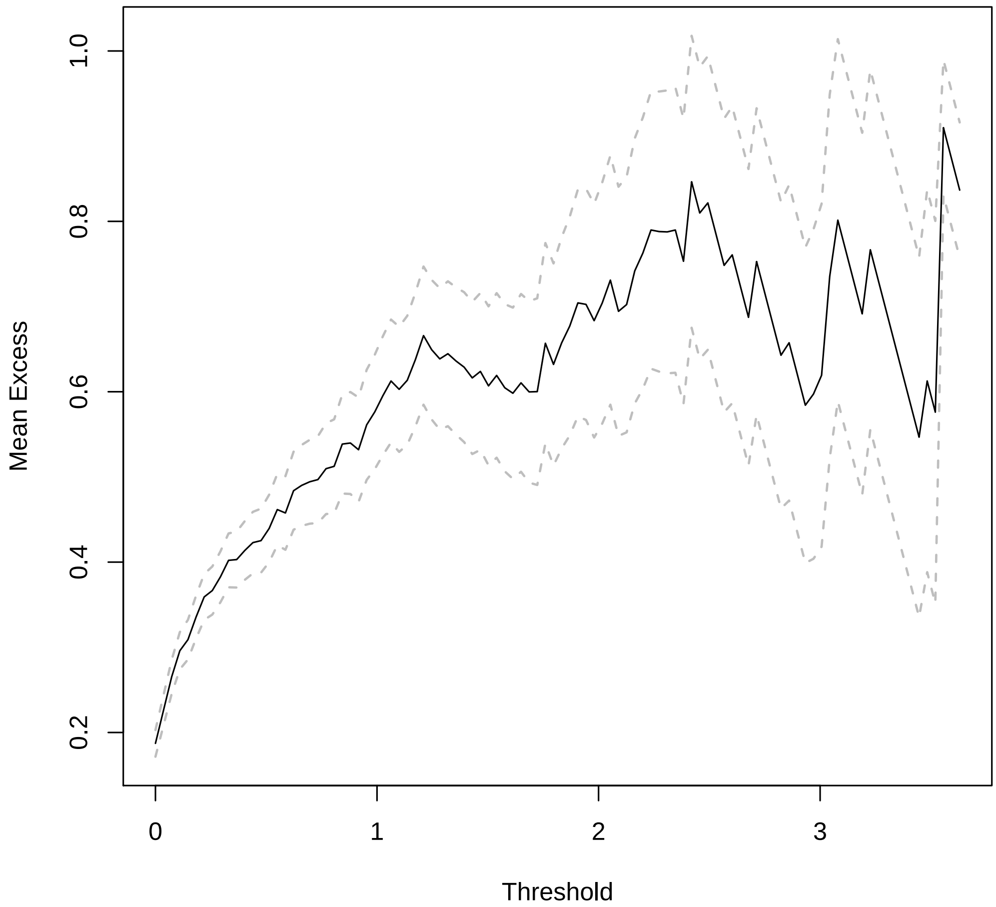


Figure 23: Parameter values from fitting the PP model using a range of thresholds to the daily Fort Collins precipitation data.

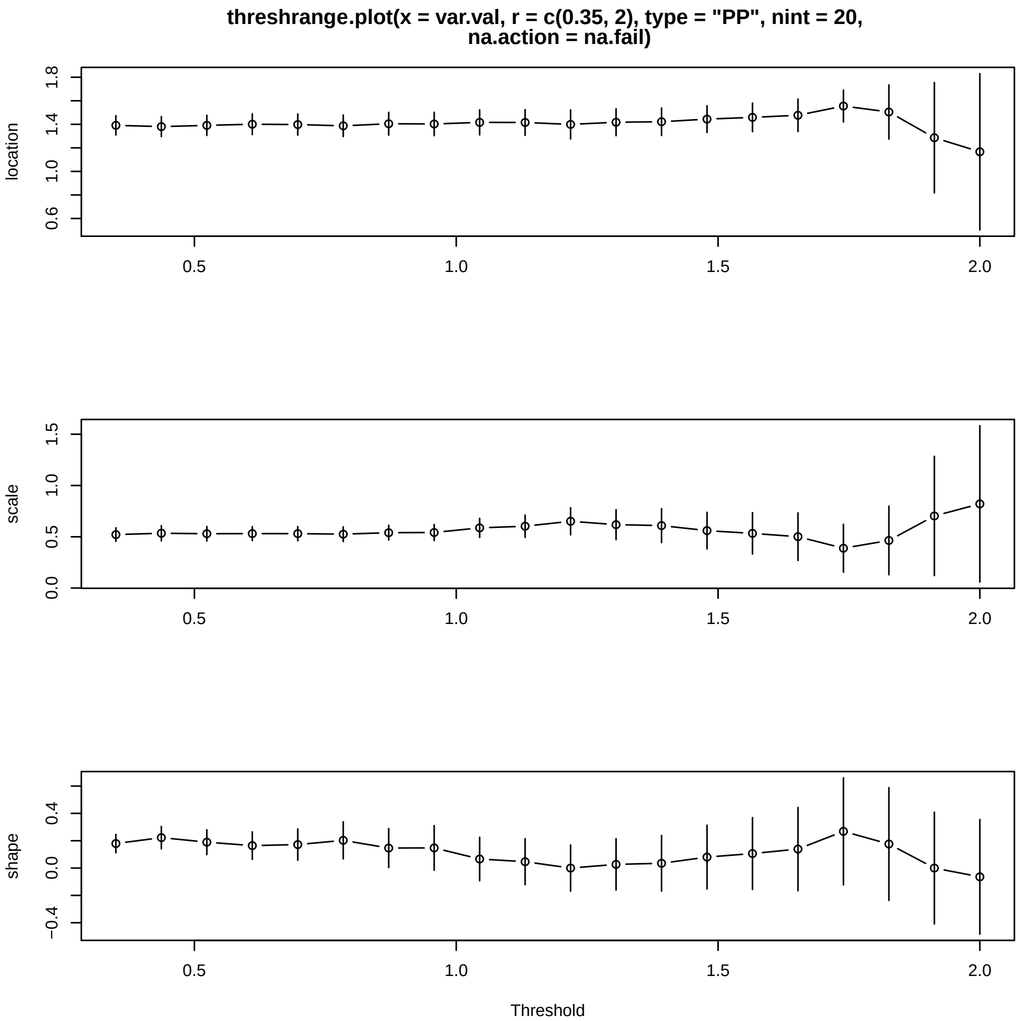


Figure 24: GP df fit for a range of thresholds to the hurricane damage (billions USD) data example.

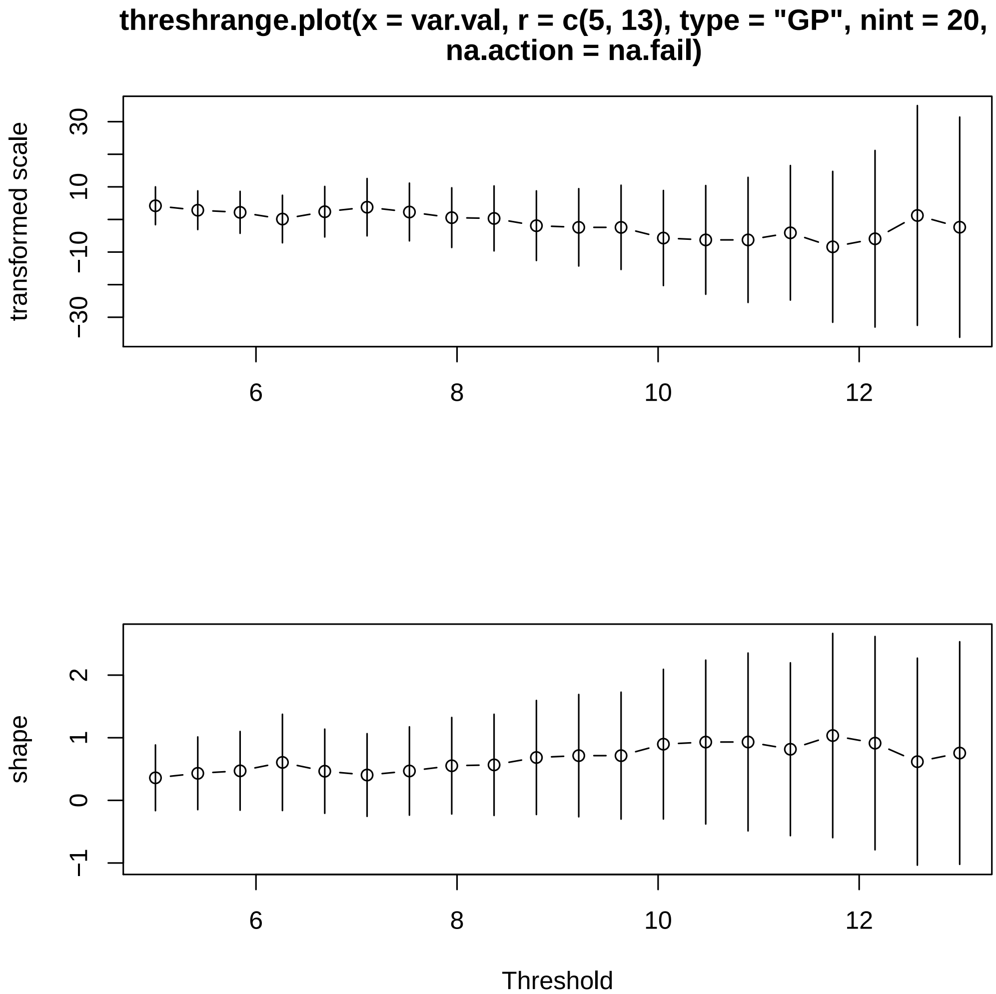
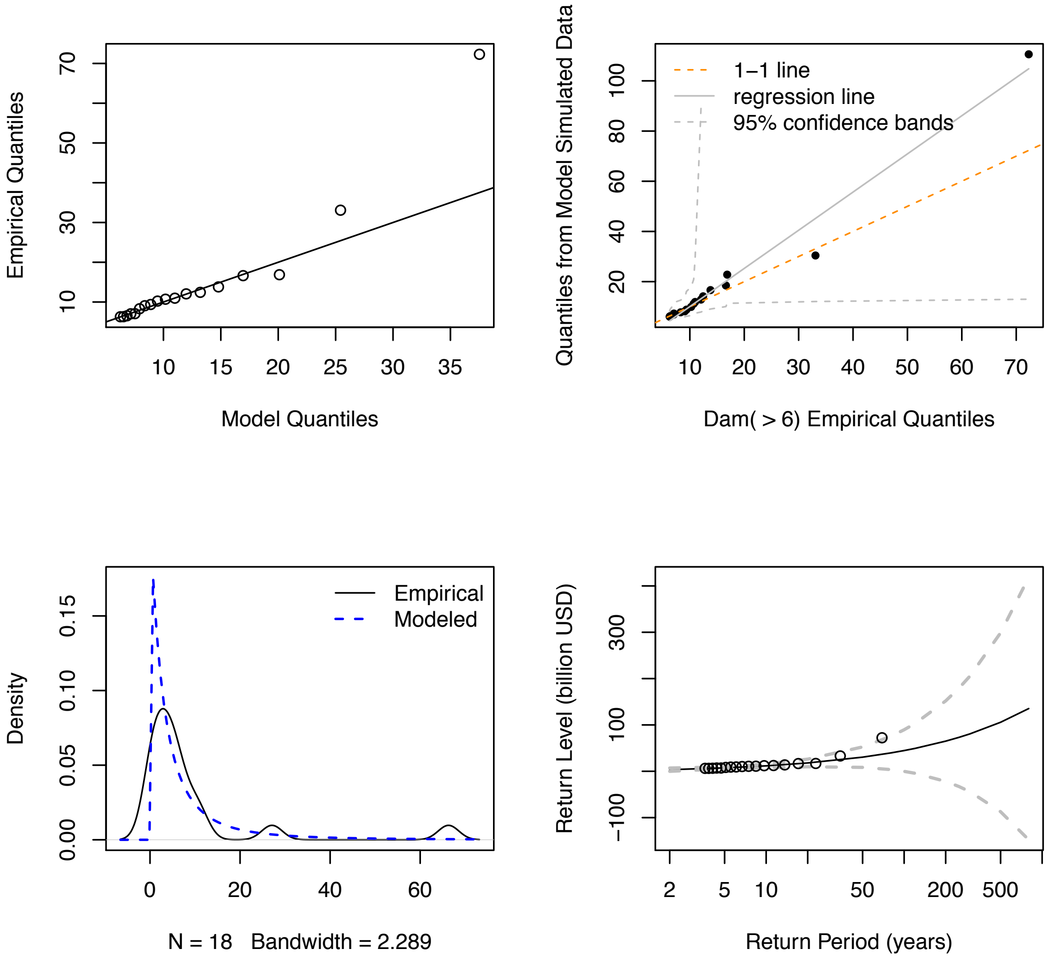


Figure 25: Diagnostic plots for the GP df fit to the hurricane damage (billions USD) data example with a threshold of 6 billion USD.



Response > `Dam`
Model Type > `Generalized Pareto (GP)`
Plot diagnostics > check
Threshold > **Value(s)/function** > `6`
Response Units > `billion USD`
Time Units > `2.06/year`
OK

Resulting diagnostic plots from the above fit are shown in Figure 25, and the assumptions appear to be reasonable.

In section 8, it was decided that a threshold of 0.395 inches is appropriate for the precipitation data from Fort Collins. Because these values occur daily for the entire year, we can use the default time units of 365.25 days per year (**Time Units** > `days`). Therefore, we are ready to fit the GP df to these data.

Analyze > **Extreme Value Distributions**
Select:
Data Object > `FortCollinsPrecip`
Response > `Prec`
Model Type > `Generalized Pareto (GP)`
Plot diagnostics > check
Threshold > **Value(s)/function** > `0.395`
Response Units > `inches`
OK

Diagnostic plots from the above fit are shown in Figure 26, and the assumptions for fitting the GP df to the data appear to be reasonable.

Figure 27 shows diagnostic plots from fitting the GP df to the flood damage data; in particular, the variable named `USDMG` from the `Flood` data example available from `extRemes` and loaded into `in2extRemes` in section 5.2 as `FloodDamage`. We leave it for the reader to: (i) plot the data, (ii) verify that the 5 billion USD threshold used to fit the model is appropriate and (iii) reproduce Figure 27.[13]

8.3 Confidence Intervals for GP Parameters and Return Levels

Confidence intervals for GP df parameters and return levels can be obtained analogously as for the GEV df. For example, the following obtains normal approximation 95% confidence intervals for the parameters of the GP df fit to the hurricane damage data.

Analyze > **Parameter Confidence Intervals**
Select:
Data Object > **HurricaneDamage**
Select a fit > `fit1`
Method > `normal approximation`

Results from the above instructions are displayed in Table 4. Notice that the lower return level bound for the 100-year return level is negative, implying

[13] Hint: how many floods are there each year in these data?

Figure 26: Diagnostics for GP df fit to daily Fort Collins precipitation (inches) using a threshold of 0.395 inches.

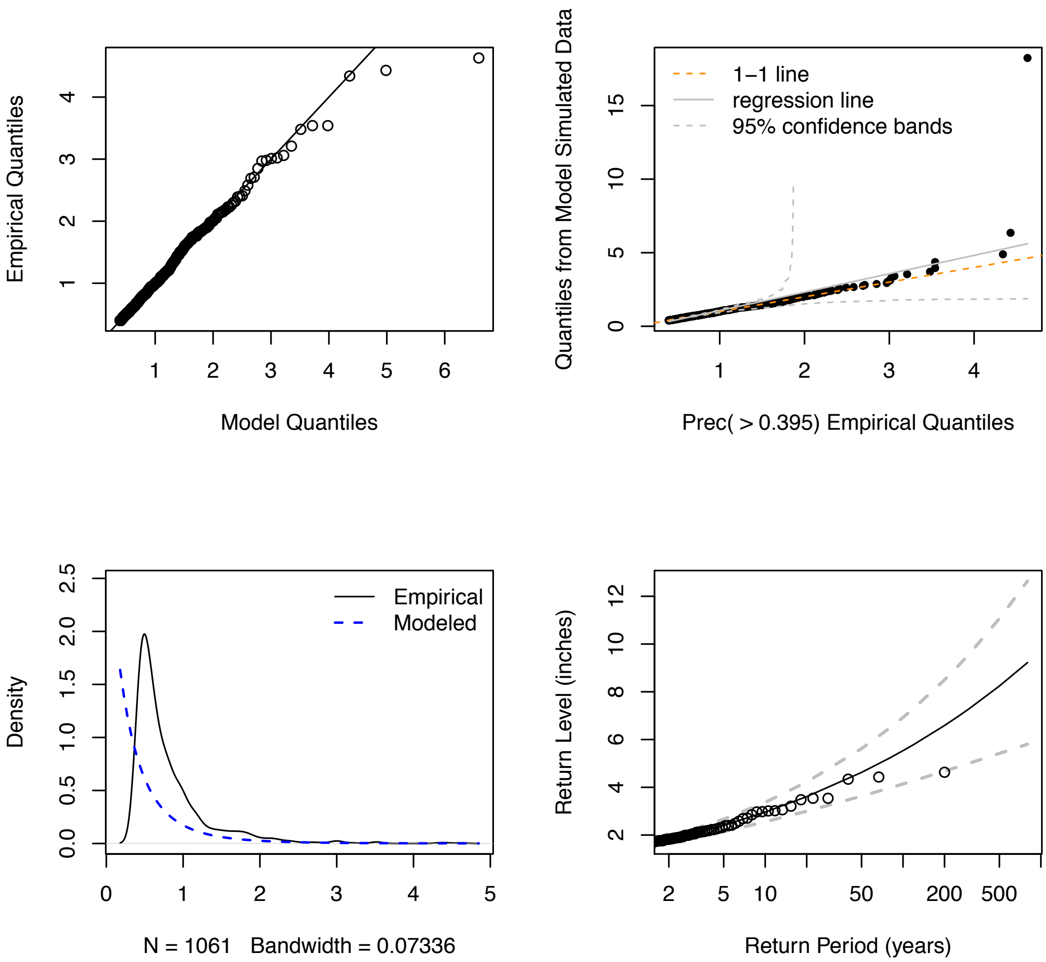


Figure 27: Diagnostic plots from fitting the GP df to US flood damage (billions USD) data with a threshold of 5 billion USD.

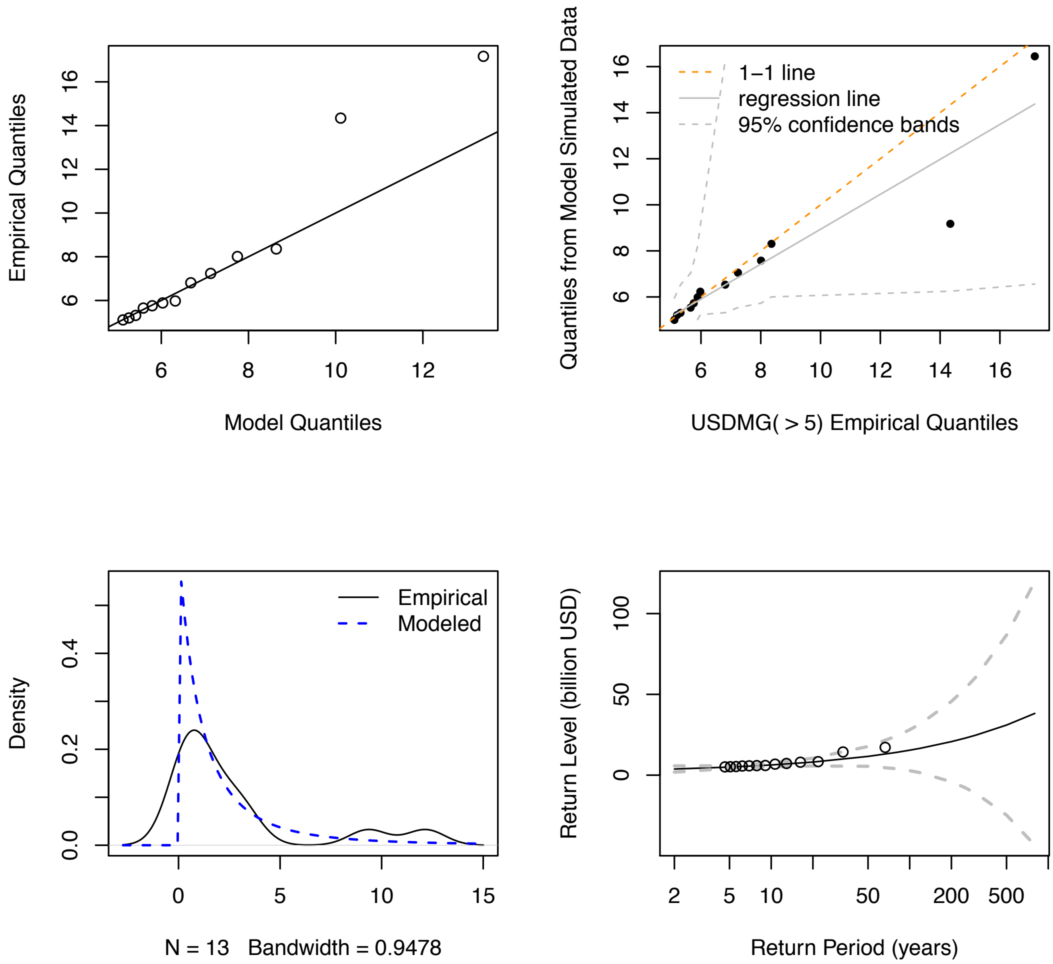


Table 4: 95% normal approximation confidence intervals for the parameters and 100-year return level from the GP df fit to hurricane damage data (billions USD). Values rounded to two decimal places.

	95% lower CI	Estimate	95% upper CI
scale	1.03	4.59	8.15
shape	-0.15	0.51	1.18
100-year return level	-0.21	44.71	89.62

that the 100-year event could give money back. Naturally, this is not the correct interpretation. In fact, there is nothing to prevent the normal approximation interval from taking on irrelevant values. However, recall from section 6.2 that the normal approximation is generally not appropriate for return levels associated with such a long period. The profile-likelihood and parametric bootstrap methods are better suited to the task. Indeed, intervals obtained from the parametric bootstrap have the property that they will preserve the natural range of the data giving more realistic bounds; in this case, using the default replicate sample size of 502, the interval is about (17.72, 120.85) billion USD.[14] The parametric bootstrap should be run again with larger replicate sample sizes to make sure it is large enough; a second run with replicate sample sizes of 1000 gave a very similar interval.

Analyze > **Parameter Confidence Intervals**
Select:
Data Object > **HurricaneDamage**
Select a fit > `fit1`
Method > `profile likelihood`
Parameter > uncheck
Lower Limit > `16`
Upper Limit > `400`
Number of points at which to calculate profile likelihood > `75`
OK

The values for the lower and upper limits, as well as the number of points at which to calculate the profile likelihood, were all determined by trial-and-error. The final result is shown in Figure 28, and the estimated confidence interval is

[14]Again, results for the parametric bootstrap will vary, but should not vary substantially.

found to be about (25.00, 363.83). Note the large skewness in the profile likelihood for these data leading to a very wide interval with the maximum likelihood estimate close to the lower bound. This last interval is perhaps the most accurate of three, and it emphasizes the difficulty in providing risk guidance for long return period events. We can say that our best guess for the 100-year return period of hurricane damage is about 44 billion USD, but we cannot rule out much larger amounts.[15]

8.4 Fitting a non-stationary GP df to Data

Figure 11 (top left) clearly shows an apparent increasing trend in the Phoenix Sky Harbor summer minimum temperature (deg F) series. In order to properly fit the GP df to these data, therefore, it is necessary to account for this trend. In doing so, we will allow for both a non-constant threshold, as well as a varying scale parameter. We vary the threshold linearly by (recall that we are interested in extremes of minimum temperature, so we will be fitting to the negative summer minimum temperature)

$$u(t) = -68 - 7 \cdot t,\ t \in [0, 1] \tag{1}$$

In Eq (1), t represents (re-scaled) time in years.[16] In order to vary the threshold, we will need to create a numeric vector, which we will assign the name `u`, using commands on the R prompt as follows.

```
u <- -68 - 7 * (Tphap$Year - 48)/42
```

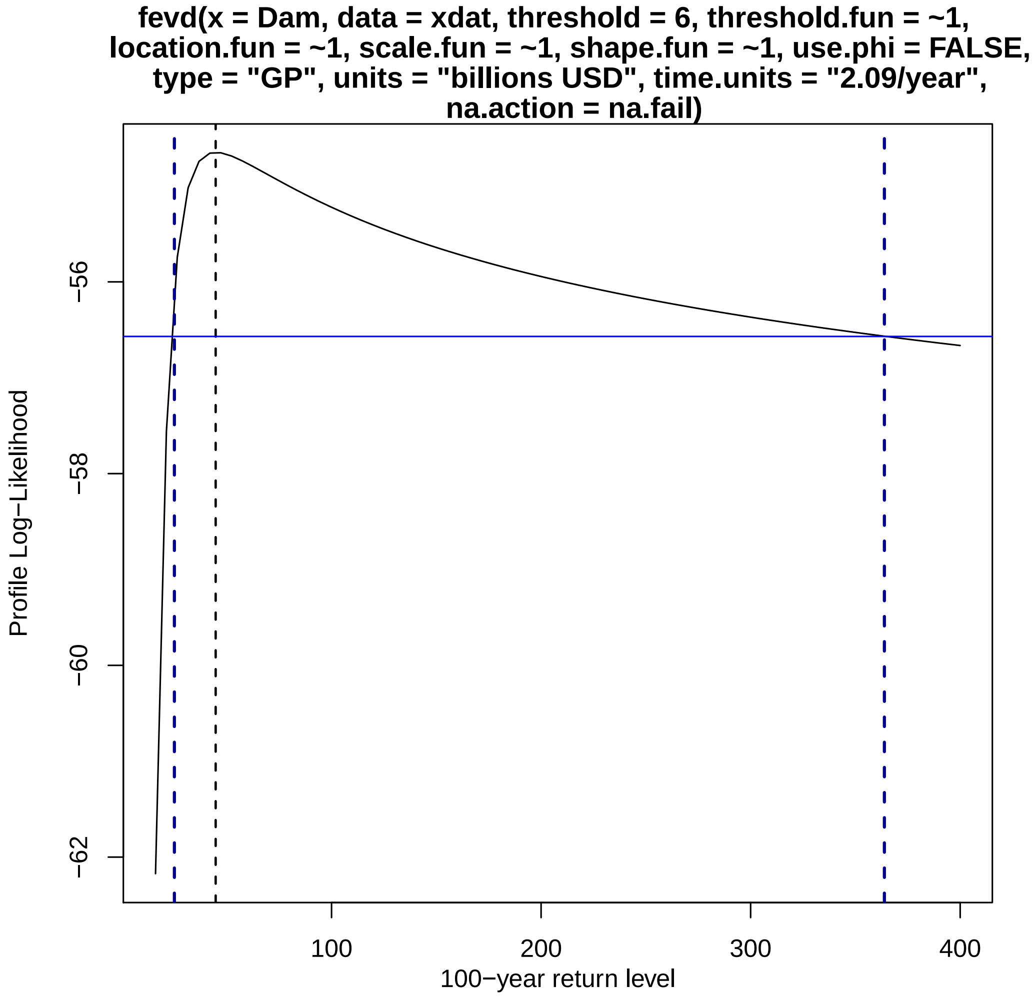
Next, we will need to transform the `Year` column of the `in2extRemes` data object `Phx`.

File > **Transform Data** > **Affine Transformation**
Select:
Data Object > `Phx`
Variables to Transform > `Year`
(X - c > `48`
)/b > `42`

[15]The flood damage data have been adjusted for inflation and changes in vulnerability (i.e., population and wealth) over time.

[16]It is recommended that the covariates within the parameter models are (at least approximately) centered and scaled. For this particular example, it makes a big difference in terms of being able to fit the model.

Figure 28: Profile likelihood for the 100-year return level from the GP df fit to the hurricane damage data with a threshold of 6 billion USD.



OK

A message is printed to the R session window that a new column has been added to our `Phx` data called `Year.c48b42`.[17] Next, we need to take the negative transformation of the `MinT` column and assign it to a new column.

File > **Transform Data** > **Negative**
Select:
Data Object > `Phx`
Variables to Transform > `MinT`
OK

A message is printed to the R session window that says the negative transformation has been taken and assigned the name `MinT.neg`. Now, we are ready to fit the model; recalling that we have 62 data points per year (summer only).

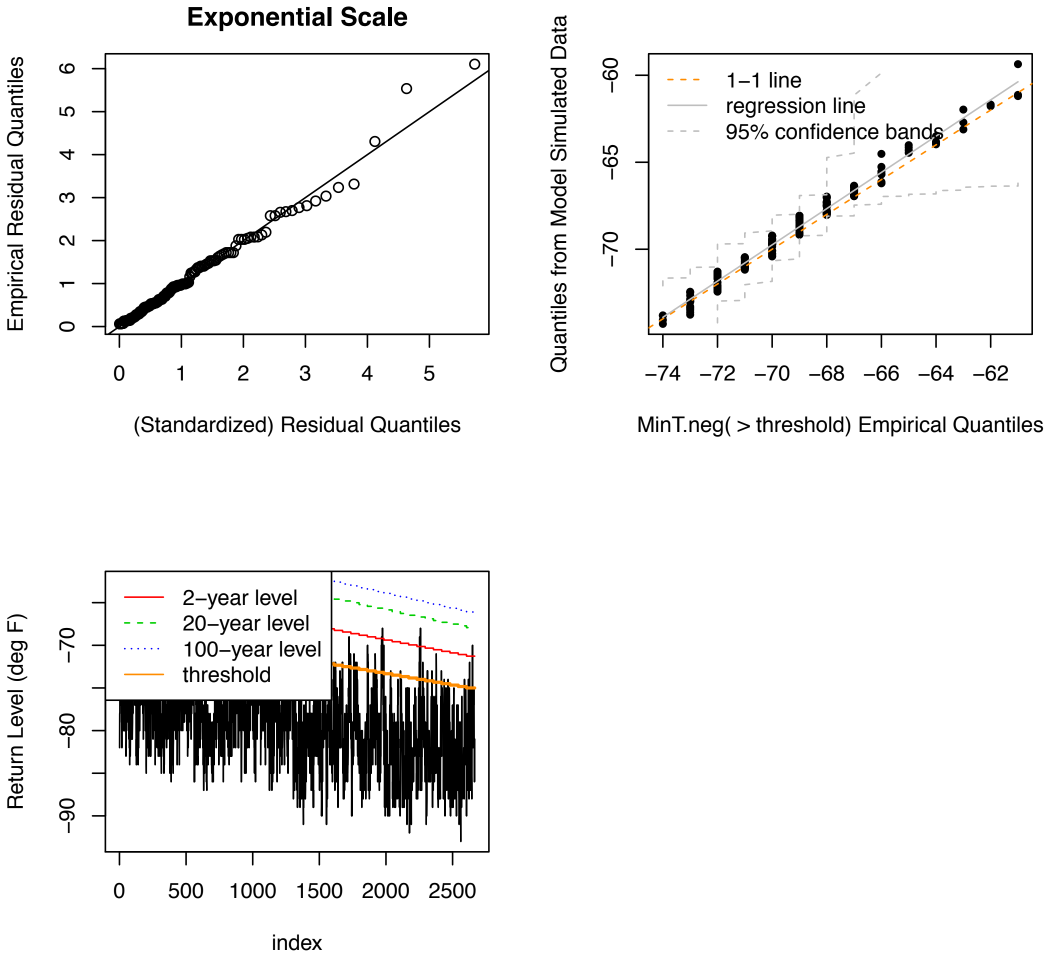
Analyze > **Extreme Value Distributions**
Select:
Data Object > `Phx`
Response > `MinT.neg`
log-link for scale parameter > check[18]
Scale parameter function~ > `Year.c48b42`
Model Type > `Generalized Pareto (GP)`
Plot diagnostics > check
Threshold > **vector** > select
Threshold > **Value(s)/function** > `u`
Response Units > `deg F`
Time Units > `62/year`
OK

Fit diagnostics from the above procedure are displayed in Figure 29. The assumptions for fitting the GP df appear to be reasonable for these data. The effective return level plot shows the data plotted (as lines) by observation entry, which is not necessarily appropriate for these data. Moreover, because the GP

[17]The new column will always be named by the name of the old column followed by `.cXbY` where `X` is the number subtracted from the previous column, and `Y` the number by which it is divided.

[18]To ensure that the scale parameter will be positive everywhere, we use the log-link, which also forces it to be a non-linear function of year.

Figure 29: Fit diagnostics for GP df fit to Phoenix Shy Harbor Airport (negative) summer minimum temperature (deg F) data with varying threshold according to Eq (1) and scale parameter varying non-linearly by (re-scaled) year.



does not directly model the frequency (only the intensity) of threshold excesses, it can be more useful to analyze these data using the PP model, where it is possible to simultaneously model non-stationarity in all of these parameters.

9 Point Process Model

Continuing with the examples from section 8.2 to which the GP was fitted to these data, we will now fit the PP model, which allows for modeling both the frequency and intensity of values that exceed a high threshold simultaneously. Because of the nature of the hurricane damage data set, whereby the dates of occurrence of hurricanes is not included, it is possible for multiple hurricanes to occur in a given year, thus violating the Poisson process assumption. Therefore, this data set is not further analyzed here.

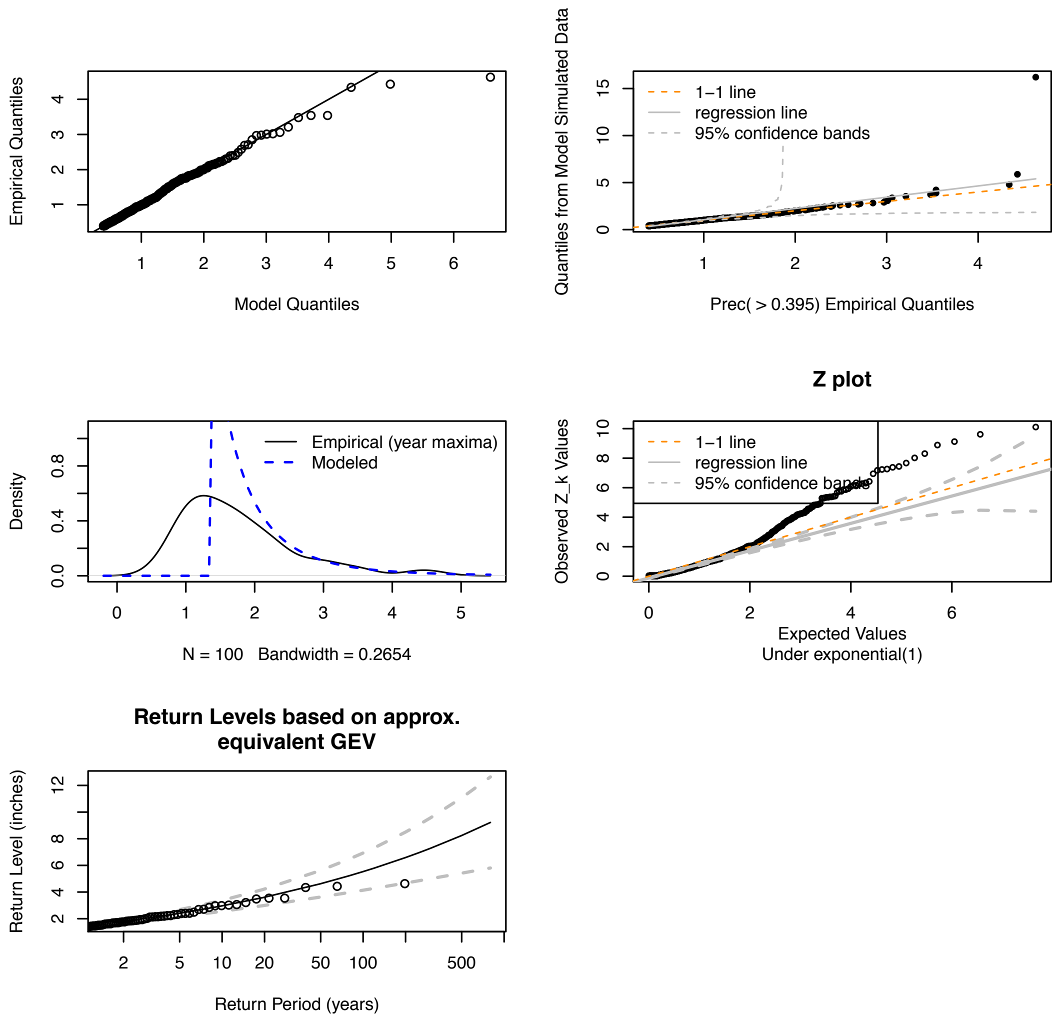
9.1 Fitting the PP model to Data

We continue analyzing the Fort Collins, Colorado precipitation data, using the threshold of 0.395 inches found to be a reasonable choice in section 8.1 (Figure 23).

Analyze > **Extreme Value Distributions**
Select:
Data Object > `FortCollinsPrecip`
Response > `Prec`
Model Type > `Point Process (PP)`
Plot diagnostics > check
Threshold > **Value(s)/function** > `0.395`
Response Units > `inches`
OK

Figure 30 displays the resulting diagnostic plots from the above fit. Once again, the plots are nearly identical to those from the GP df fit to the same data (Figure 26). However, the Z plot appears problematic. Clear curvature exists that extends well beyond the 95% confidence bands. Although the assumptions for fitting the GP df to the excesses appear to be met, those for fitting the Poisson to the frequency of exceeding the threshold apparently are not met. Figure 31 shows the diagnostic plots from fitting the PP model to these data, but with a higher threshold of 0.75 inches. Now, all of the qq-plots appear to show that the assumptions are reasonable, although the Z plot still shows some curvature, which may be a result of not incorporating seasonality into the model. Note that

Figure 30: Diagnostic plots from fitting a PP model to the Fort Collins, Colorado precipitation (inches) data using a threshold of 0.395 inches.



a substantially higher threshold of 2 inches was also applied, and while all of the diagnostics appeared to show a reasonable model, the estimated shape parameter (≈ -0.6) differs considerably from those obtained from lower thresholds and with the block maxima approach. Therefore, such a high threshold does not appear to be reasonable for fitting the PP to these data.

9.2 Confidence Intervals for PP Parameters and Return Levels

Confidence intervals for PP parameters and return levels can be attained analogously as in sections 6.2 and 8.3. For example, to obtain normal approximation limits for the PP model fit to the hurricane damage data in section 9.1, we can use the following.

Analyze > **Parameter Confidence Intervals**
Select:
Data Object > **HurricaneDamage**
Select a fit > `fit2`
Method > `normal approximation`

Table 5 shows the results of the above instructions. It is useful to compare this table with Table 4 where the 95% normal approximation limits are provided for the parameter estimates and 100-year return level from fitting the GP df to these same data. Indeed, results are nearly identical for the 100-year return level and shape parameter. In addition to having a lower bound for the 100-year return level that is below zero (i.e., outside the natural range of the variable), that for the scale parameter is also negative. Again, this issue is an artifact of the normal approximation method, and perhaps better return level confidence intervals can be found from another method, such as the profile likelihood method.

9.3 Fitting a non-stationary PP model to Data

In section 8.4, a non-stationary GP df is fit to Phoenix summer (negative) minimum temperature (deg F) data with a linearly varying threshold by year and an annually varying scale parameter. A difficulty in interpreting the results concerns the inability to coherently model the frequency of exceeding the event. In this section, this obstacle is removed by fitting a non-stationary PP model to the data. In this case, we will allow the location parameter, as well as the threshold, to vary. Recall that we assigned the R object `u` in section 8.4 in order to allow the

Figure 31: Diagnostic plots from fitting a PP model to the Fort Collins, Colorado precipitation (inches) data using a threshold of 0.75 inches.

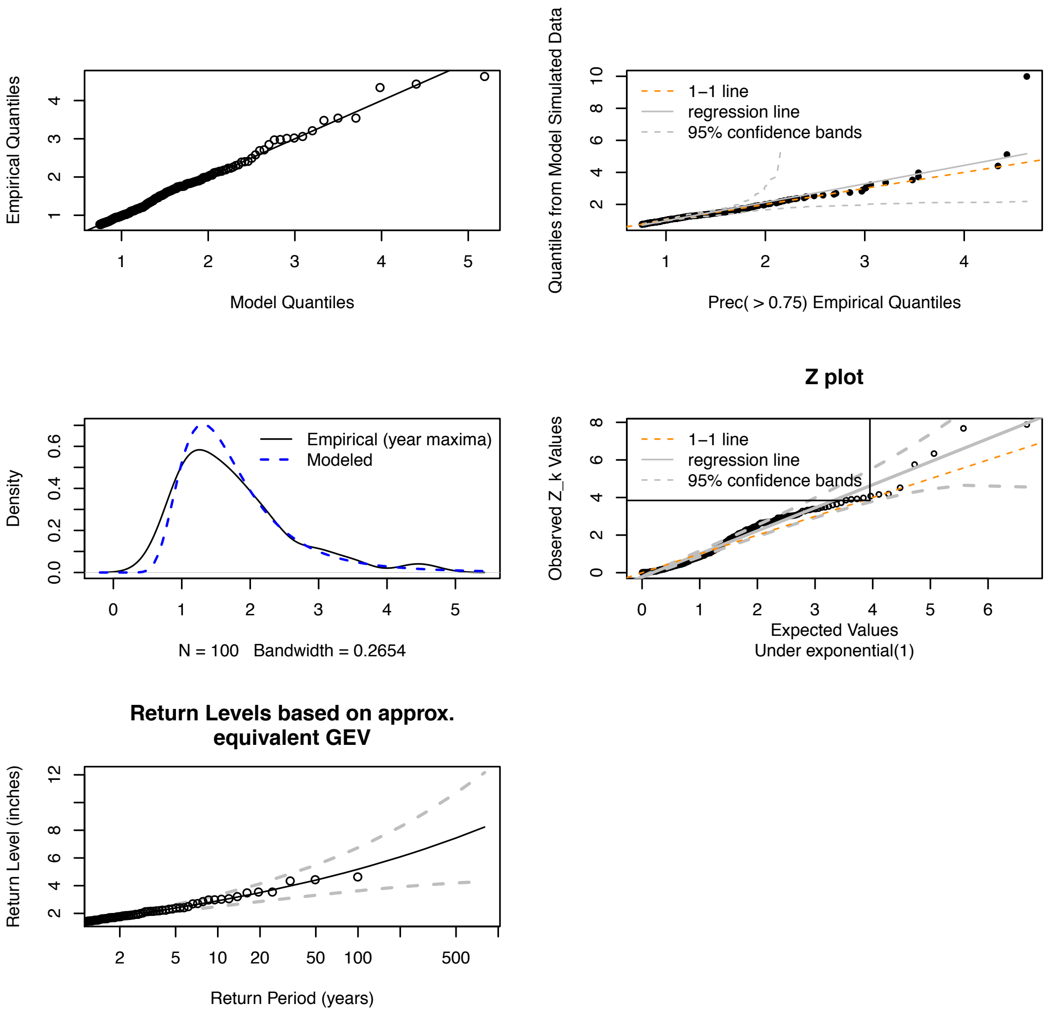


Table 5: Parameter estimates and 95% normal approximation confidence intervals (rounded to two decimal places) for the PP model fit to hurricane damage data in section 9.1 (cf. Table 4).

	95% lower CI	Estimate	95% upper CI
location	-3.23	1.54	6.32
scale	-1.13	2.31	5.74
shape	-0.16	0.51	1.18
100-year return level	-1.49	44.58	90.64

threshold to vary. We also transformed two columns of our data to obtain the two new columns: `Year.c48b42` and `MinT.neg`.

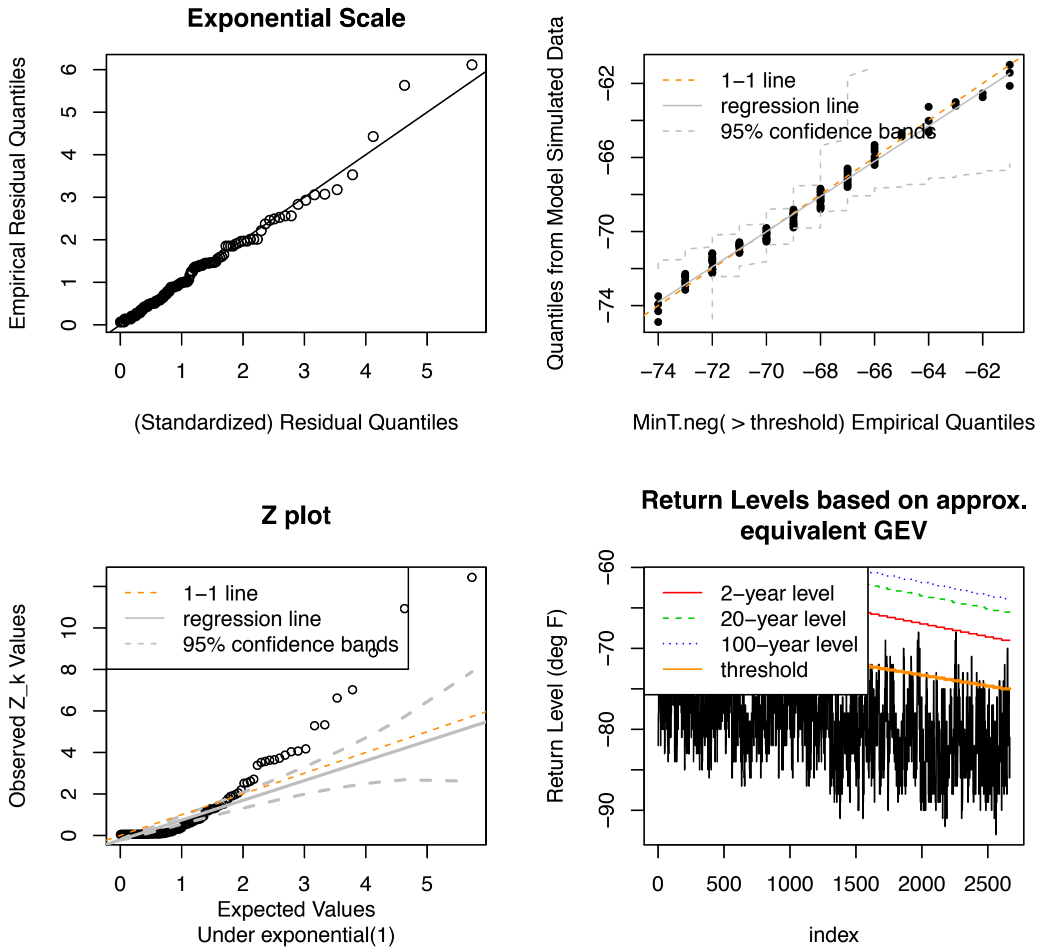
Analyze > **Extreme Value Distributions**
Select:
Data Object > `Phx`
Response > `MinT.neg`
Location parameter function~ > `Year.c48b42`
Model Type > `Point Process (PP)`
Plot diagnostics > check
Threshold > **vector** > select
Threshold > **Value(s)/function** > `u`
Response Units > `deg F`
Time Units > `62/year`
OK

The diagnostic plots in Figure 32 indicate that the assumptions for fitting the GP df to the excesses are met, but the Z plot once again shows that the frequency of occurrence is far too rapid. We will revisit fitting the PP model to these data in section 10.

To make plots that are perhaps more more appropriate than those shown, the command-line functions need to be used. See `?return.level` from `extRemes` to learn how to obtain the return levels for plotting (see also Gilleland and Katz, 2014).

In section 9 it was determined that although a threshold of 0.395 inches was sufficient as far as the excesses was concerned, the mean time between the events of exceeding the threshold was not modeled fast enough. However, increasing

Figure 32: Diagnostic plots from fitting a non-stationary PP model to (negative) summer maximum temperature (deg F) at Phoenix Sky Harbor airport.



the threshold to 0.75 inches resulted in a much improved Z plot. Therefore, we continue this example using a threshold of 0.75 inches. We might also consider allowing the threshold to vary, but we will not follow this path here. Recall that from Figure 10, no obvious trend through time is apparent, but a clear seasonal variation exists. The following may take a few moments to run.

Analyze > **Extreme Value Distributions**
Select:
Data Object > `FortCollinsPrecip`
Response > `Prec`
Location parameter function~ > `cos(2 * pi * day / 365.25)`
Model Type > `Point Process (PP)`
Plot diagnostics > check
Threshold > **Value(s)/function** > `0.75`
Response Units > `inches`
OK

Diagnostic plots (Figure 33) suggest that the assumptions for fitting this model to the data are met, although some curvature remains in the Z plot.

9.4 Relating the PP model with the Poisson-GP model

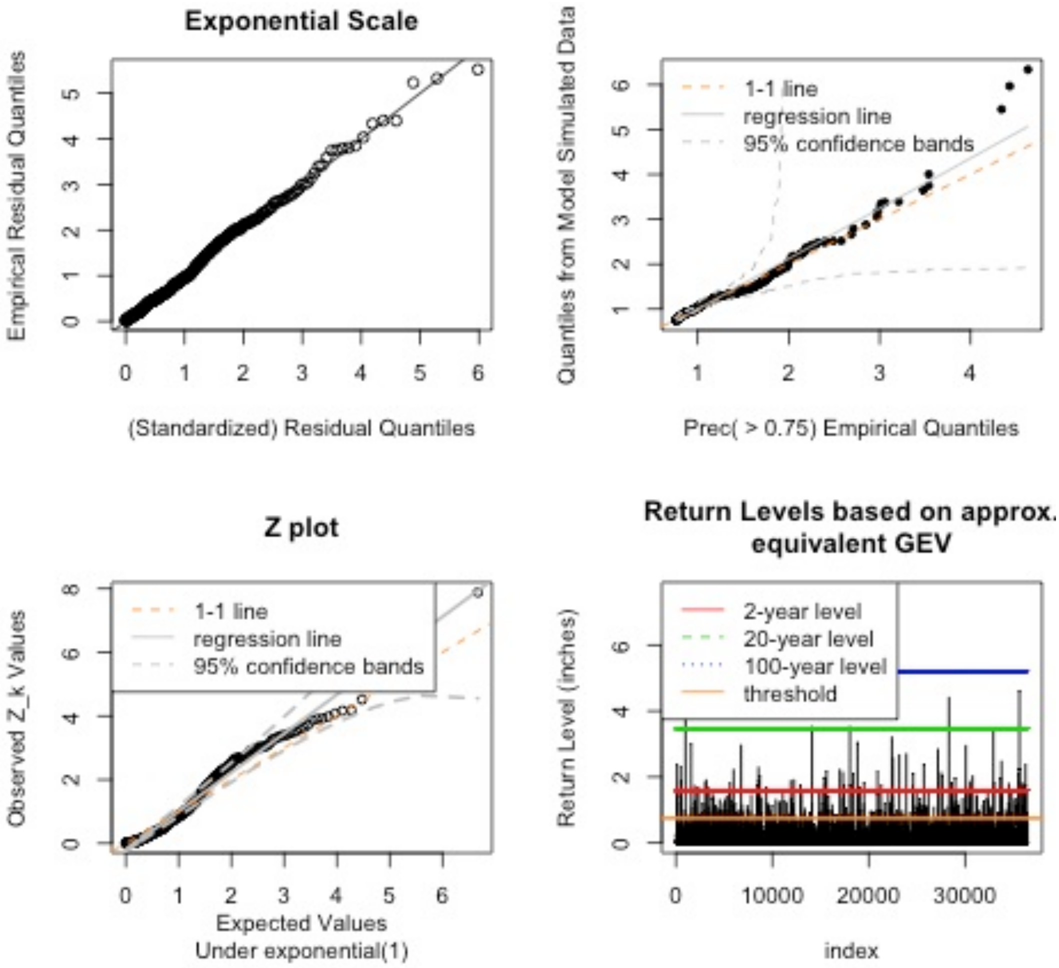
The parameters of the PP model can be expressed in terms of the parameters of the GEV df, or, equivalently, through transformations specified in appendix B.2, in terms of the parameters of a Poisson process and of the GPD (i.e., a Poisson-GP model).

For example, when fitting the PP model to the Fort Collins precipitation data with a threshold of 0.75 inches and 365.25 observations per year, the following parameter estimates $\hat{\mu} \approx 1.395$, $\hat{\sigma} \approx 0.530$ and $\hat{\xi} \approx 0.179$ are obtained. If we fit the GP df to these data with a threshold of 0.75 inches, we get $\hat{\sigma}^*(0.75 \text{ inches}) \approx 0.414$ and $\hat{\xi} \approx 0.179$. The shape parameter should be identical for both models, but variability in the optimization procedures will result in small differences. In this case, they are the same within the uncertainty, indeed, they are the same within five decimal places.

Using `sum(Fort$Prec > 0.75)` from the R session window prompt, we see that 395 values exceed the threshold of 0.75 inches out of a total of 36,524 observations. Using Eq (15), we have that the (log) MLE for the Poisson rate parameter is

$$\ln \hat{\lambda} \approx \ln\left(365.25 \cdot \frac{395}{36,524}\right) \approx 1.373743.$$

Figure 33: Diagnostic plots for the non-stationary PP model fit to Fort Collins, Colorado precipitation (inches) data with a constant threshold of 0.75 inches and seasonally varying location parameter.



Plugging these values into Eq (17) and (18) gives

$$\ln \hat{\sigma} \approx \ln \hat{\sigma}^*(0.75 \text{ inches}) + \hat{\xi} \ln \hat{\lambda} = \ln(0.4140984) + 0.1792041 \cdot (1.373743) \approx -0.635$$

so that $\hat{\sigma} \approx \exp(-0.635) \approx 0.530$ as was estimated from fitting the PP model directly to the data.

10 Extremes of Dependent Sequences

As mentioned in section 3, uncertainty information will be affected by dependence, if any exists, over a high threshold (e.g., confidence intervals will be too narrow), and `in2extRemes` allows for using the runs declustering method to try to remove any clustering above the threshold. In order to help diagnose whether or not dependence above the threshold exists, the auto-tail dependence function can be plotted (Reiss and Thomas, 2007).

Plot > **Auto tail-dependence function**
Select:
Data Object > `FortCollinsPrecip`
Select Variable > `Prec`
OK

The resulting plot from the above selections is shown in Figure 34. In the top panel (labeled `rho`), the sample auto tail-dependence function based on ρ is produced (Reiss and Thomas, 2007, Eq (2.65) p. 76), which takes on values between zero and one (inclusive). If the values over a high threshold are stochastically independent, then the values of $\hat{\rho}$ should be close to $1 - u$ at each lag, where u is the quantile threshold, which in the example above $u = 0.8$. Inspection of the top panel in Figure 34 shows that all lags greater than one are fairly close to $1 - 0.8 = 0.2$, but that the lag-one term is about 0.4, so the assumption of independence may or may not be reasonable for these data. Perfect dependence will yield $\rho = 1$, which is why lag-zero has a value of 1.

The bottom panel in Figure 34 shows the sample auto tail-dependence function based on $\bar{\rho}$ (Reiss and Thomas, 2007, Eq (13.28) p. 325), which takes values between -1 and 1 (inclusive). Again, perfect dependence will have $\bar{\rho} = 1$, which is why lag-zero attains this value. If $\rho = 0$, then the variable is tail independent and $\bar{\rho}$ gives the degree of *dependence*. For independent variables, $\bar{\rho} = 0$ at each lag. The sample auto tail-dependence function based on $\bar{\rho}$ for this example is close to zero at all lags greater than one, but once again, the lag-one term appears to be greater than zero, if small. Therefore, we have further evidence that the assumption of independence over a high threshold may not be valid, but the degree of dependence is not very strong.

An additional entry field not used above is **Quantile Threshold**. This value can be any number between zero and one, and should be high enough (close to one) to reduce bias, but low enough to include enough data (lower variance).

Another tool available for diagnosing dependence above the threshold is the extremal index. For independent data, the extremal index is one, although the converse does not hold so that a value of one does not necessarily mean that the data are independent. If it is less than one, however, then some dependence (clustering) exists in the limit. Two estimators are available with `extRemes` and `in2extRemes`: *runs* (Coles, 2001, sec. 5.3.2) and *intervals* (Ferro and Segers, 2003); both based on declustering algorithms..

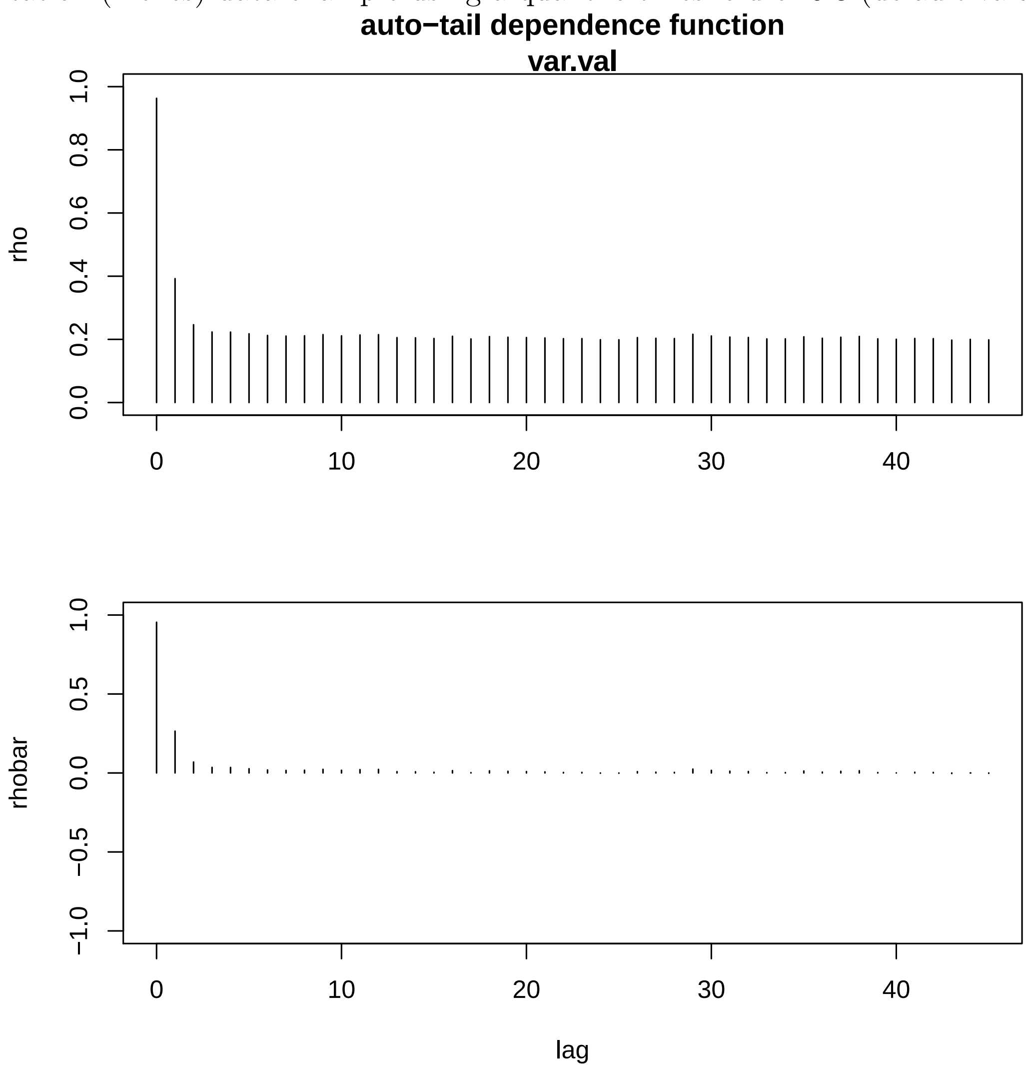
Analyze > **Extremal Index**
Select:
Data Object > `FortCollinsPrecip`
Select Variable > `Prec`
Threshold > `0.75`
OK

The intervals estimate performed from the above instructions can be greater than one, which is not a valid parameter value. Therefore, if the estimate is larger than one, then the function sets it to 1, and subsequently the value of exactly 1 can be estimated despite its otherwise being a single point in a continuous parameter space. The estimated extremal index using the intervals method is about 0.788 for the daily Fort Collins precipitation data set with 95% confidence interval based on the bootstrap procedure suggested by Ferro and Segers (2003) of (0.701, 0.893), which concurs with our findings from the auto tail-dependence function plots (Figure 34) that some (statistically significant) extremal dependence exists in the tails.

The output from the above commands also yields an estimate for the number of clusters and the run length. A cluster is a group of threshold exceeding values that occur near one another in time, and a run length is the (average) length between clusters. Again, the bootstrap procedure of Ferro and Segers (2003) provides confidence intervals for both of these estimates.

Perhaps the simplest method for handling dependence in the tails is to `decluster` the series. That is, first identify clusters of threshold excesses, and then use a single value (usually the maximum of the values within a cluster). In doing so, it is generally best not to completely remove the other cluster members from the series in order to preserve the frequency information. Therefore, in `extRemes` and sub-

Figure 34: Auto tail-dependence function for daily Fort Collins, Colorado precipitation (inches) data example using a quantile threshold of 0.8 (default value).



sequently `in2extremes`, the default is to set those values to the threshold. The simplest method for declustering a series is the runs declustering method. In this method, a cluster is identified as beginning with the first value that exceeds the threshold, and ending once r values fall below the threshold, where r is the chosen (or possibly estimated) run length.

We now return to the Phoenix (negative) minimum summer temperature example. Recall that a distinct increasing trend in the summer minimum temperatures is evident (decreasing for the negative). Therefore, we use our varying threshold created in section 8.4 and named `u`.[19] The following will take longer to run than for the hurricane damage example.

Analyze > **Extremal Index**
Select:
Data Object > `Phx`
Select Variable > `MinT.neg`
Threshold > `u`
OK

Table 6 displays the results from above. Clearly, dependence is an issue as the extremal index is considerably less than one.

Because we have tail dependence in the Phoenix minimum temperature data, we will decluster these data. From Table 6, the estimated run length we should use is 5, but first, we will decluster with a run length of 1 for comparison (leaving it to the reader to decluster with a run length of 5). Recall that for the Phoenix temperature data, we have only 62 days per year (i.e., summer only). Therefore, we do not want clusters to form across years because we have natural breaks. Recall also that a distinct trend exists for the data and that in section 8.4 we performed the negative transformation of our observed variable of interest, `MinT` (minimum temperature), which is called `MinT.neg`.

File > **Decluster**
Select:
Data Object > `Phx`
Variable to Decluster > `MinT.neg`
Decluster by > `Year`
Threshold(s) > `u`
Plot data > `check`

[19] Currently, the auto tail-dependence function in `extRemes` does not allow for a varying threshold, so it is not clear that it would make sense to use it for the Phoenix temperature data.

Table 6: Extremal index (values rounded to two decimal places), number of clusters (rounded to nearest whole number) and run length for Phoenix (negative) minimum summer temperature data, as estimated by the intervals method.

	95% lower CI	Estimate	95% upper CI
extremal index	0.24	0.40	0.57
number of clusters	28	60	54
run length	1	5	18
		Declustered (run length = 1)	
extremal index	0.55	0.78	1.00
number of clusters	31	65	58
run length	1	5	16
		Declustered (run length = 5)	
extremal index	0.64	0.91	1.00
number of clusters	31	61	55
run length	1	6	16

The upper 95% CI for numbers of clusters is clearly incorrect as it is below the estimated value, and should be ignored.

OK

Figure 35 shows the plot that is produced. Both the original data (gray circles) and the *declustered* data (black circles) are graphed on the same plot. Values that used to be above the threshold, but were not the maximum of their cluster, are set to the threshold value, which is the reason for the clustering along the threshold line (dashed line in the figure).

A new column is added to the data set called `MinT.neg.uur1dcbyYear`, where the naming convention is to append `.uXrYdc[by]Z` to the original column's name, where `X` is the threshold (either a number or the name of a vector), `Y` is the run length used, and if the data are grouped into natural clusters, then `by` is also appended with `Z` the name of the column by which the data were grouped.[20]

Information is also printed to the R session window, which includes the estimated extremal index for the original data based on both the *runs* and *intervals* estimates. Table 6 shows the estimated extremal index for the declustered series using run lengths of both 1 and 5. Using a run length of 1, the data appear to have less of a tail dependence issue, and for those declustered with a run length of 5, the estimated extremal index is very nearly 1 indicating independence in the limit. Figure 36 shows the diagnostic plots for the declustered series having used a run length of 5. Comparing with Figure 32, we see that the Z plot is now very nearly straight suggesting that the assumptions for fitting the PP model to the declustered series are met.

Appendix

The information provided in this appendix is intended as a quick reference only, not a complete guide to EVT/EVA. Brief, relevant information is given for some of the fundamental components of `in2extRemes`.

A Extreme Value Distributions

The forms of the EVD's and return levels are provided here along with some of their basic properties.

[20]Because of this naming convention, `in2extRemes` will not allow the same declustering procedure to be performed more than once without first removing the new column from the data set (e.g., by **File** > **Scrubber**).

Figure 35: Plot showing the year breaks and declustered (run length = 1) negative Phoenix minimum summer temperature data. Light gray circles are the original data points, and black circles are the resulting data points after declustering. Values above the threshold that have been “removed" by the declustering algorithm are still included in the data set, but have been re-set to the threshold value.

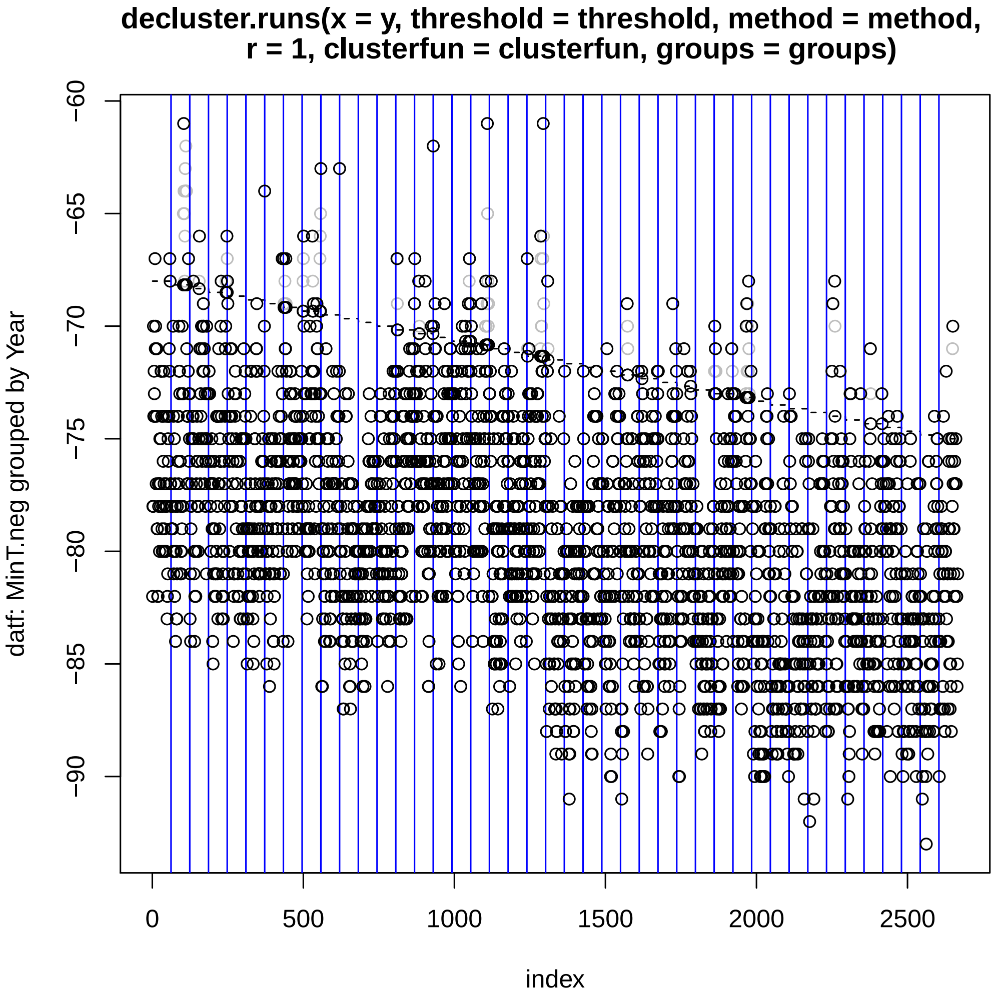
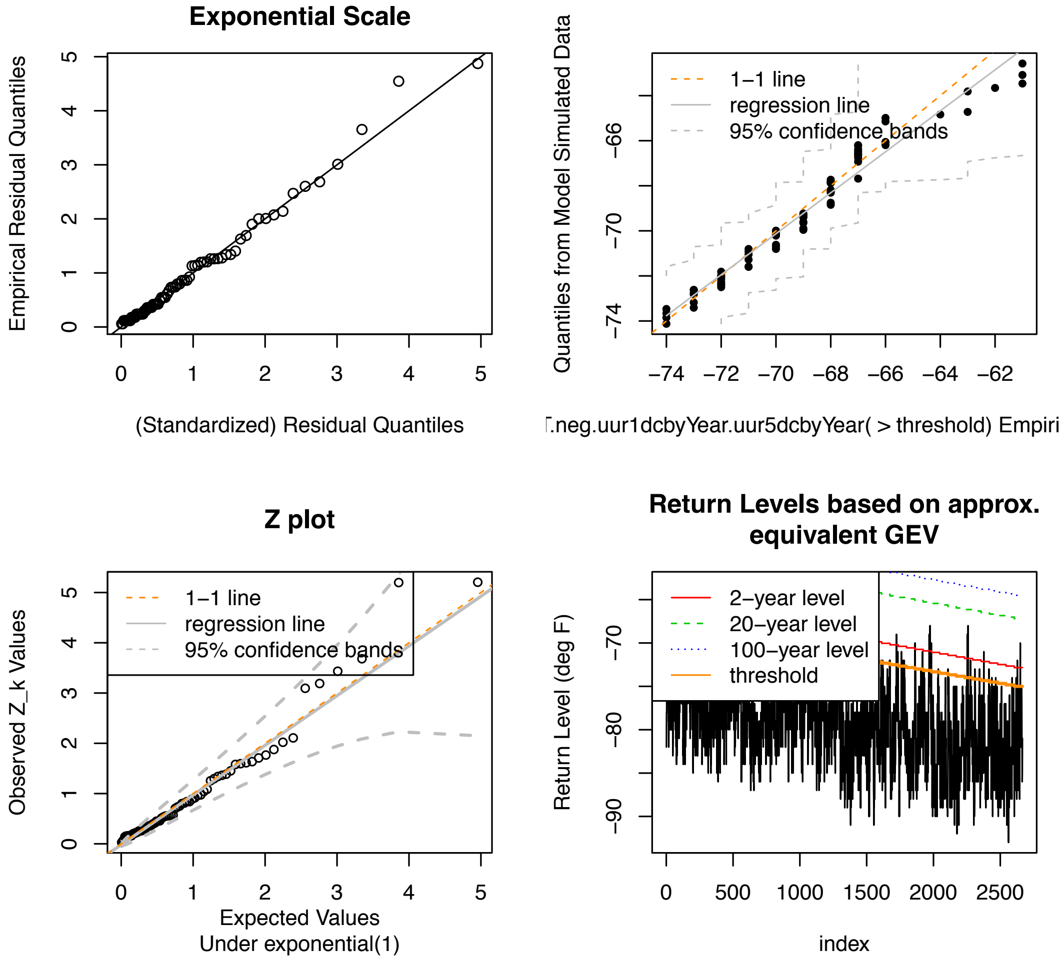


Figure 36: Diagnostic plots for the PP model fit to (negative) declustered (run length = 5) Phoenix summer minimum temperature data.



A.1 The Generalized Extreme Value Distribution

The three types of GEV df's are better known as the Gumbel, Fréchet and (reverse) Weibull, respectively. Each has a location and scale parameter, and the latter two have a nonzero shape parameter, which we denote as ξ. The Gumbel case corresponds to $\xi = 0$, which is defined by continuity. Jenkinson (1955), a meteorologist, noted that the three types can be combined to be expressed as a single parametric family, namely the GEV family (the result was originally found by von Mises (1936)). The GEV df is defined as

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

where $y_+ = \max\{y, 0\}$, $\sigma > 0$ and $-\infty < \mu, \xi < \infty$. From this representation, the Gumbel type is obtained by taking the limit as $\xi \to 0$ giving

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

The Weibull distribution has a bounded upper tail at $z_u = \mu - \sigma/\xi$ (a function of the parameters), so that the probability of observing a value larger than z_u is zero. Note that the more common form of the Weibull df has a lower bound, and the version that belongs to the GEV family is often referred to as the reverse (or reflected) Weibull, and sometimes as the stretched exponential.

Some properties of the GEV df are shown in Table 7. Note that the k-th order moment exists only when $\xi < 1/k$, and consequently heavy tailed distributions have rapidly vanishing moments. In particular, the variance is not finite when $\xi \geq 1/2$, and, in such a case, inference based on the first two moments will be invalid.

Often, return levels are of interest when analyzing extremes. To that end, let z_p be the return level for the $T = 1/p$ return period, $0 < p < 1$. One interpretation of z_p is that it is the level expected to be exceeded on average once every T periods, where one period is the block length over which the maxima are taken.

A return level is equivalent to the corresponding quantile of the distribution. That is, we seek the value, z_p, such that $G(z_p) = 1-p$. For the GEV df, we simply invert Eq (2) and solve for z_p to get

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

The return level is linear in μ and σ, but nonlinear in ξ.

Table 7: Some properties of the GEV df (2). Here, $g_k = \Gamma(1 - k\xi)$, where $\Gamma(x)$ is the gamma function.† The parameters are defined on: location $-\infty < \mu < \infty$, scale $\sigma > 0$, and shape $-\infty < \xi < \infty$.

(cumulative) df	$F(x) = \exp\left[-\left\{1 + \xi\left(\frac{x-\mu}{\sigma}\right)\right\}_+^{-1/\xi}\right]$
probability density function	$f(x) = \frac{1}{\sigma}(1 + \xi(x-\mu)/\sigma)_+^{-1/\xi-1} F(x)$
Mean	$E(X) = \mu - \sigma(1 - g_1)/\xi$ for $\xi < 1$
Variance	$\mathrm{Var}(X) = \sigma^2(g_2 - g_1^2)/\xi^2$ for $\xi < 1/2$
Median	$\mu - \sigma(1 - \ln 2)^{-\xi}/\xi$
Mode	$\mu + \sigma((1+\xi)^{-\xi} - 1)/\xi$ for $\xi \geq -1$
r^{th} Probability Weighted Moment (PWM)	$\beta_r = \mu - \sigma[1 - (r+1)^{\xi}\Gamma(1-\xi)]/(\xi(r+1))$
L-moments	$\lambda_1 = \beta_0$, $\lambda_2 = 2\beta_1 - \beta_0$, $\lambda_3 = 6\beta_2 - 6\beta_1 + \beta_0$
Support	$-\infty < x < \infty$ for $\xi = 0$ (type I Gumbel, cf. Table 8), $x > \mu - \sigma/\xi$ for $\xi > 0$ (type II Fréchet) and $x < \mu - \sigma/\xi$ for $\xi < 0$ (type III Weibull).

† The Gamma function, $\Gamma(x)$, $x > 0$ is defined to be $\int_0^\infty t^{x-1}e^{-t}dt$, which reduces to $(x-1)!$ when x is a positive integer.

Table 8: Some properties of the Gumbel df (3). The parameters are defined on: location $-\infty < \mu < \infty$ and scale $\sigma > 0$.

(cumulative) df	$F(x) = \exp\left[-\exp\left\{-(x-\mu)/\sigma\right\}\right], \; -\infty < x < \infty$
probability density function	$f(x) = \frac{1}{\sigma}\exp(\frac{x-\mu}{\sigma})F(x)$
Mean	$E(X) = \mu + \gamma\sigma$, where γ is the Euler-Mascheroni constant†
Variance	$\mathrm{Var}(X) = (\pi\sigma)^2/6$
Median	$\mu - \sigma\ln(\ln 2)$
Mode	μ

†The Euler-Mascheroni constant is defined as the limiting difference between the harmonic series and the natural logarithm. Specifically, $\gamma = \lim_{n\longrightarrow\infty}\left[\sum_{k=1}^{n} 1/k - \ln(n)\right] = \int_1^\infty (1/\lfloor x\rfloor - 1/x)dx$. Its numerical value is ≈ 0.5772.

A useful simplification of (4) is obtained by letting $y_p = -1/\ln(1-p)$, giving

$$z_p = \begin{cases} \mu + \frac{\sigma}{\xi}\left[y_p^{\xi} - 1\right], & \text{for } \xi \neq 0, \\ \mu + \sigma \ln y_p, & \text{for } \xi = 0. \end{cases} \quad (5)$$

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

A.2 The Generalized Pareto Distribution

The GP df is given by

$$F(x) = 1 - \left[1 + \xi\left(\frac{x-u}{\sigma(u)}\right)\right]_+^{-1/\xi}, \quad (6)$$

where $x > u$, scale parameter $\sigma(u) > 0$, and shape parameter $-\infty < \xi < \infty$. To obtain the exponential case (i.e., when $\xi = 0$), the limit as $\xi \to 0$ is taken from the above expression to obtain

$$F(x) = 1 - e^{-(x-u)/\sigma}. \quad (7)$$

Note that for the exponential case $\sigma(u) = \sigma + 0\mu = \sigma$. Properties for the POT df's are given in Tables 9 and 10

The p-th quantile for the GP df is found by solving $F(y_p) = 1 - p$ for y_p with F as in Eq (6). Namely,

$$y_p = \begin{cases} u - \frac{\sigma(u)}{\xi}\left[1 - (1-p)^{-\xi}\right] & \xi \neq 0, \\ u - \sigma \ln(1-p) & \xi = 0 \end{cases} \quad (8)$$

For reporting return levels, however, it is convenient to convert the conditional probability of excesses over a high threshold u exceeding a value, say x, into an unconditional probability based on the original random variable, X (i.e., instead of the excesses $X - u$). To do this, use the definition of conditional probability to get

$$\begin{aligned} \Pr\{X > x\} &= \Pr\{X > x | X > u\} \Pr\{X > u\} \\ &= \left[1 + \xi\left(\frac{x-u}{\sigma(u)}\right)\right]^{-1/\xi} \zeta_u, \end{aligned} \quad (9)$$

Table 9: Some properties of the GP df (6). The parameters are defined on $\sigma(u) > 0$ and $-\infty < \xi < \infty$.

(cumulative) df	$F(x) = 1 - \left[1 + \xi\left(\frac{x-u}{\sigma(u)}\right)\right]_+^{-1/\xi}$
probability density function	$f(x) = \frac{1}{\sigma(u)}\left[1 + \xi\left(\frac{x-u}{\sigma(u)}\right)\right]^{-1/\xi-1}$
Mean	$E(X) = u + \frac{\sigma(u)}{1-\xi}$ for $\xi < 1$
Variance	$\mathrm{Var}(X) = \frac{\sigma(u)^2}{(1-\xi)^2(1-2\xi)}$, for $\xi < 1/2$
Median	$u + \frac{\sigma(u)}{\xi}(2^\xi - 1)$
r^{th} Probability Weighted Moment (PWM)	$\beta_r = \sigma(u)/\left[(r+1)(r+1-\xi)\right]$
L-moments	$\lambda_1 = \beta_0$ $\lambda_2 = 2\beta_1 - \beta_0$ $\lambda_3 = 6\beta_2 - 6\beta_1 + \beta_0$
Support	$x \geq u$ for $\xi \geq 0$ (Exponential (cf. Table 10) and heavy-tailed Pareto) $u \leq x \leq u - \sigma(u)/\xi$ for $\xi < 0$ (Beta)

Table 10: Some properties of the exponential df (7). The scale parameter is defined on $\sigma > 0$.

(cumulative) df	$F(x) = 1 - e^{-(x-u)/\sigma}$
probability density function	$f(x) = \frac{1}{\sigma} e^{-(x-u)/\sigma}$
Mean	$E(X) = u + \sigma$
Variance	$\text{Var}(X) = \sigma^2$
Median	$u + \sigma \ln 2$
Support	$x > u$

where $\zeta_u = \Pr\{X > u\}$. Then, to find the value, $x_m > u$, that is exceeded on average once every m observations, we solve $\Pr\{X > x_m\} = 1/m$ in Eq. (9) for x_m to get

$$x_m = \begin{cases} u + \frac{\sigma}{\xi}\left[(m\zeta_u)^{\xi} - 1\right] & \xi \neq 0, \\ u + \sigma \ln(m\zeta_u) & \xi = 0. \end{cases} \quad (10)$$

Note that Eq (10) differs from (8) only in that the form of $1 - p$ is changed to reflect the frequency of exceeding u, and to be interpreted in terms of exceeding x_m once every m observations (e.g., where m is usually on a time scale much shorter than a year so that it is customarily converted into years).

As before, plotting x_m against $\ln(m)$ results in the same qualitative features as with the GEV return levels. That is, when $\xi = 0$ the plot is a straight line, $\xi > 0$ results in a concave curve, and $\xi < 0$ a convex curve.

A.3 The Point Process Characterization for Extremes

Recall that a discrete random variable N follows a Poisson distribution if

$$\Pr(N = m) = \frac{\lambda^m \exp(-\lambda)}{m!}, \text{ for } m = 0, 1, 2, \ldots$$

Here, λ is the intensity or rate parameter, which is also the mean and variance of the distribution. For extremes, define the random variable $N_n = \sum_{i=1}^{n} I_i$ where I_i is 1 if $X_i > u_n$ for a high threshold, u_n, (the subscript n indicates that the threshold will increase as the sample size increases). Because the X_i's are assumed to be iid, N_n is a Binomial random variable with mean $n\Pr\{X > u_n\} = n(1 - F(u_n))$.

That is, the average number of times that the threshold is exceeded depends on both the sample size and the probability of the event's occurring. As the sample size n increases, it is desired that the mean be stable. That is, the threshold must be chosen so that $\lim_{n\to\infty} n(1-F(u_n)) = \lambda > 0$. Under this condition, it turns out that the counting variable N_n can be approximated by the Poisson variable N with parameter λ.

Now, consider the event $N_n = 0$, or having no event occur in n trials. This event is tantamount to saying that the maximum of the n sample values is less than or equal to the threshold u_n, so that $\Pr\{\max\{X_1, \ldots, X_n\} \leq u_n\} = \Pr\{N_n = 0\} \approx e^{-\lambda}$ by the Poisson approximation for the binomial distribution.

The Poisson-GP model combines the Poisson process with a GP distribution for excesses over the threshold. This model is a special case of a *marked* Poisson process (e.g. Guttorp and Minin, 1995) with a mark (or excess) being associated with each event of exceeding the threshold.

A process $\{N(t), Y_1, \ldots, Y_{N(t)}, t \geq 0\}$ is *Poisson-GP* if

1. $\{N(t),\, t \geq 0\}$ is a one-dimensional, homogeneous Poisson process with rate parameter λ,

2. Conditional on $N(t) = k \geq 1$, the excesses (or marks) $Y_1, \ldots, Y_k$ are iid GP with parameters $\sigma(u)$ and ξ.

The two-dimensional Poisson process developed for EVA by Smith (1989), is a further extension of these ideas (see also Leadbetter et al., 1983; Resnick, 1987). In this context, the times when the values of a random sequence X exceeds a high threshold *and* the associated excess values are treated as a two-dimensional point process, so that if the process is stationary, then the limiting form is a *non-homogeneous* Poisson process with intensity measure, Λ, on a two-dimensional set of the form $A = (t_1, t_2) \times (x, \infty)$ given by

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

where the parameters μ, $\sigma > 0$ and ξ are the same parameters as the GEV df.

A common way to deal with non-stationarity for extreme values is to allow the parameters to vary. For example, one might consider a linear trend through time in the location parameter of the GEV df, giving

$$\mu(t) = \mu_0 + \mu_1 \cdot t,\ t = 1, 2, \ldots.$$

For the scale parameter, care should be taken to ensure that it is positive everywhere. Therefore, when incorporating non-stationarity into this parameter estimate, the log link function is often used. For example, a linear trend through time could be modeled in the scale parameter as

$$\sigma(t) = \sigma_0 + \sigma_1 \cdot t,\, t = 1, 2, \ldots.$$

The above parametrization may be useful provided it is positive for all values of t under consideration. Using the log-link, the model is no longer linear, rather

$$\ln \sigma(t) = \phi(t) = \phi_0 + \phi_1(t),\, t = 1, 2, \ldots.$$

The above model yields $\sigma(t) = \exp(\phi_0 + \phi_1 \cdot t)$. It is also possible to model non-stationarity in the shape parameter in a similar way. For POT models, it can be useful to allow the threshold to vary as well, but generally this procedure should only be necessary when one or more of the parameters varies as well.

The interpretation of return levels for such non-stationary models becomes difficult. One method for describing the models is to look at *effective* return levels (e.g. Gilleland and Katz, 2011). An effective return level is simply the return level you would have for a specific value(s) of the covariate(s) (i.e., a time varying quantile). For example, if a linear trend is modeled in the location parameter, the effective return level for time $t = 5$ would be found by first finding $\mu(5) = \mu_0 + \mu_1 \cdot 5$, and then estimating the return level using $\mu(5)$ as the location parameter. Such an approach can be applied for numerous values of the covariate(s) so that one can observe how the df changes with the covariate(s).

B Confidence Intervals and Hypothesis Testing

A result from statistical theory says that MLE's follow a normal df (e.g., Coles, 2001, Chapter 2), which provides a way of making inferences about EVD parameters, as well as their associated return levels. For a parameter estimate, say $\hat{\theta}$ of θ, the $(1 - \alpha) \cdot 100\%$ confidence intervals based on a normal approximation are simply

$$\hat{\theta} \pm z_{\alpha/2}\text{se}(\hat{\theta}),$$

where $\text{se}(\hat{\theta})$ is the standard error of $\hat{\theta}$, which generally needs to be estimated in practice.

Another well known result is that the MLE for any scalar function of a parameter, say $g(\theta)$ can be obtained by substituting the MLE for the parameter, θ, into the function, g. That is, the MLE for $\eta = g(\theta)$ is given by $\hat{\eta} = g(\hat{\theta})$. Suppose θ is a d-dimensional parameter (e.g., $\theta = (\mu, \sigma, \xi)$, the parameters of the GEV df), then a further result says that if $\eta = g(\theta)$ is a scalar function, the MLE for η is approximately normally distributed with its variance given by a simple quadratic matrix multiplication of the gradient of η and the inverse Hessian of the likelihood for θ. Specifically, the variance for η is given by $\nabla_\eta^T H_\theta^{-1} \nabla_\eta$. This last result is known as the *delta method*, and provides a simple method for obtaining confidence intervals for return levels from the extreme value df's.

The gradient for the return levels of a GEV df is given by (cf Coles, 2001, Chapter 3)

$$\begin{aligned} \nabla x_p^T &= \left[\frac{\partial z_p}{\partial \mu}, \frac{\partial z_p}{\partial \sigma}, \frac{\partial z_p}{\partial \xi}\right] \\ &= \left[1,\ -\xi^{-1}(1 - y_p^{-\xi}),\ \sigma\xi^{-2}(1 - y_p^{-\xi}) - \sigma\xi^{-1} y_p^{-\xi} \ln y_p\right] \end{aligned} \quad (11)$$

where $y_p = -\log(1-p)$, and can be evaluated by substituting in the parameter estimates. The gradient for the GP df return levels is similarly computed. An approximate $(1-\alpha)\cdot 100\%$ confidence interval for the return level, x_p, is given by

$$x_p \pm z_{\alpha/2}\sqrt{\nabla x_p^T H^{-1} \nabla x_p}.$$

The delta method confidence intervals are based on the assumption that the MLE of interest is approximately normally distributed. However, for longer return levels that are most likely to be of interest, this approximation is likely not very accurate. The profile-likelihood can be used to find more accurate intervals, which also produces confidence intervals that are more realistic in shape (i.e., not symmetric like the delta method intervals, but longer above the point estimate than below).

B.0.1 Likelihood-Ratio Test

To test whether incorporation of a covariate in the parameter is statistically significant, the likelihood-ratio test can be used. Including more parameters in the model will necessarily increase the maximized likelihood function (cf. Coles, 2001, Chapter 2), and this method tests whether or not the improvement is statistically significant. The test compares two *nested* models so that one model, the base model, must be contained within the model with more parameters (e.g., setting one or more parameters in the more complicated model to zero would result in the base model). Suppose ℓ_0 is the log-likelihood value for the base model and ℓ_1 that for the model with more parameters. Then the test statistic, called the deviance statistic,

$$D = 2(\ell_1 - \ell_0)$$

follows an approximate χ^2_ν df with degrees of freedom, ν, equal to the difference in the number of parameters between the two models. The null hypothesis that $D = 0$ is rejected if D exceeds the $1-\alpha$ quantile of the χ^2_ν df.

B.0.2 Information Criteria

Alternatives to the likelihood-ratio test for comparing the relative quality of a statistical model include the Akaike Information Criterion (AIC) and the Bayesian Information Criterion (BIC). Neither requires a model to be nested like the likelihood-ratio test, but nor do they correspond to a formal statistical test. The AIC can be defined as

$$\mathrm{AIC}(p) = 2n_p - 2\ell,$$

where n_p is the number of parameters in the p-th model, and ℓ its maximized log-likelihood value. Similarly, for the p-th model fit to data with a sample size n, the BIC is

$$\mathrm{BIC}(p) = n_p \ln n - 2\ell.$$

Both the AIC and BIC attempt to counteract the problem of over fitting a model from adding more parameters by incorporating a penalty based on the number of parameters. The BIC is more parsimonious than the AIC. Among the candidate models, the one with a lower AIC/BIC is preferred.

B.0.3 Profile likelihood

The profile log-likelihood method can be used to obtain confidence intervals that are usually more accurate than those based on the normality assumption (cf Coles, 2001, Chapter 2). Letting θ be a d-dimensional parameter vector (e.g., $\theta = (\mu, \sigma, \xi)$ for the GEV df), the log-likelihood for θ can be written as $\ell(\theta_i; \theta_{-i})$, where θ_{-i} indicates the parameter vector θ excluding the i-th parameter. The profile log-likelihood for θ_i is defined as

$$\ell_p(\theta_i) = \max_{\theta_{-i}} \ell(\theta_i, \theta_{-i}). \tag{12}$$

In words, for each value of the parameter θ_i, the profile log-likelihood for θ_i is the maximized value of the log-likelihood with respect to all the other components of θ.

In general, if θ is partitioned into two components, (θ_1, θ_2), with θ_1 a k-dimensional parameter vector, and θ_2 a $d-k$-dimensional parameter vector, then Eq (12) generalizes to

$$\ell_p(\theta_1) = \max_{\theta_2} \ell(\theta_1, \theta_2).$$

Approximate confidence regions can then be obtained using the deviance function, $D_p(\theta_1) = 2\{\ell(\hat{\theta}) - \ell_p(\theta_1)\}$, as described in appendix B.0.1, which approximately follows a χ^2_k distribution, leading to the $1-\alpha$ confidence region given by

$$C_\alpha = \{\theta_1 : D_p(\theta_1) \leq \chi^2_{1-\alpha,k}\}. \tag{13}$$

This methodology can be extended for inference on combinations of parameters, such as return level estimates (cf Coles, 2001, Chapter 3). This requires a re-parameterization of the GEV df so that the return level is one of the parameters. For example, for any specified return level z_p, the following expression can be used

$$\mu = z_p + \frac{\sigma}{\xi}[1 - \{-\ln(1-p)\}^{-\xi}].$$

That is, μ in the likelihood function is replaced with the above expression and the parameter vector (z_p, σ, ξ) is estimated instead of (μ, σ, ξ). A complication in using the profile likelihood approach for confidence intervals for extreme value df's is that the likelihood must be optimized with numerical techniques. Therefore, the process can be computationally expensive and difficult to automate.

B.0.4 Parametric Bootstrap

Another method for finding confidence intervals for MLE's is the bootstrap method (see e.g., Efron and Tibshirani, 1993; Lahiri, 2003; Gilleland, 2010). There are various different types of bootstrap methods, and because `extRemes` has functions only for the parametric bootstrap, we give some details for this approach here.[21] The approach can be summed up with the following steps. Suppose the data to which we have fit an EVD has size n.

1. Draw a random sample of size n from the fitted parametric distribution (e.g., GV).

2. Fit the EVD of interest to the random sample and record the parameter and/or return level(s) estimate(s).

3. Repeat steps 1 and 2 R times to obtain a *replicate sample* of the parameter(s) estimate(s) of interest.

4. From the replicate sample of parameter(s)/return level(s) in step 4, estimate $(1-\alpha)\cdot 100\%$ confidence intervals by finding the $1-\alpha$ quantiles for each parameter/return level from the replicate sample.

[21] Although the bootstrap procedure generally requires fewer assumptions about the distribution of the statistic(s) of interest for inference than most other confidence interval methods, assumptions do remain. These assumptions are generally not met for the usual iid resampling procedure that is most well known. Because distributions for the extremes are already being assumed, and the appropriateness of their assumptions checked, the parametric approach seems to be a reasonable choice.

The size, R, of the replicate samples can be determined by starting with a relatively small value (faster computation) and increased for a second trial. If the results are similar to those done previously, then the first value for R should suffice.[22] If not, increase R until the results are stable.

B.1 The Z plot

The *Z plot* is a diagnostic plot for checking an assumption for fitting the PP model, with possibly time-varying parameters, to data. It was introduced by Smith and Shively (1995). The one-dimensional PP for the event of exceeding a high threshold u is a non-homogeneous Poisson process with rate parameter $\lambda(t) = [1 + \xi(t)(u(t) - \mu(t))/\sigma(t)]^{-1/\xi(t)}$. Denote T_0 as the starting time of the record, and let $T_1, T_2, \ldots$ be the successive times that u is exceeded, and consider the variable

$$Z_k = \int_{T_{k-1}}^{T_k} \lambda(t)dt, \; k \geq 1. \tag{14}$$

For a non-homogeneous Poisson process, the variables $Z_1, Z_2, \ldots$ should be independent and exponentially distributed with mean one. One way to diagnose whether or not this is the case is to make a qq-plot of Z_k against the quantiles from an exponential df with mean one; we call such a plot, the Z plot. In the above notation, the parameters all written as functions of time t to emphasize that this plot is appropriate for models that depend on a covariate; however, it also applies in the special case of a homogeneous Poisson process (i.e., parameters independent of t).

B.2 Poisson-GP Model

The parameters of the point process model can be expressed in terms of those of the GEV df or, equivalently, through transformations specified below, in terms of the parameters of a one-dimensional Poisson process and the GP df (i.e., a Poisson-GP model). Specifically, given a high threshold u, μ, σ and ξ from the PP model, we have the following equations (the shape parameter is unchanged):

$$\ln \lambda = -\frac{1}{\xi} \ln \left[1 + \xi \frac{u - \mu}{\sigma} \right], \tag{15}$$

where λ is the Poisson rate parameter and for GP scale parameter $\sigma^*(u)$,

$$\sigma^*(u) = \sigma + \xi(u - \mu). \tag{16}$$

[22]Technically, one should run several trials, but practice dictates that if similar results are obtained, then there is no need to continue.

Eq (15) and (16) can be used to solve for σ and μ simultaneously to obtain the parameters of the associated PP model (Katz et al., 2002). Doing so yields the following equations.

$$\sigma = \lambda^{\xi}\sigma^{*}(u) \tag{17}$$

$$\mu = u - \frac{\sigma}{\xi}\left(\lambda^{-\xi} - 1\right). \tag{18}$$

The block maxima and POT approaches can involve a difference in time scales, h. For example, if observations are daily ($h \approx 1/365$) and annual maxima are modeled, then it is possible to convert the parameters of the GEV df for time scale h to the corresponding GEV parameters for time scale h' (see Katz et al., 2005) by converting the rate parameter, λ, to reflect the new time scale; namely,

$$\lambda' = \frac{h}{h'}\lambda.$$

C Examples of Command-Line Scatter Plots in R

In this section, the R code used to make Figures 10 to 12 is shown. For more information about plotting in R, see `?plot` and `?par`. The data frames `Fort`, were loaded from `extRemes` in section 5.2.

Figure 10 was created with the following commands.

```
par(mfrow = c(2,2))
plot(Prec~month, data = Fort, pch = 16, col="darkblue",
    xlab="", ylab = "Daily Precipitation (inches)",
    xaxt = "n")
axis(1, at = 1:12, labels = c("Jan", "Feb", "Mar", "Apr",
    "May", "Jun", "Jul", "Aug", "Sep", "Oct", "Nov", "Dec"))
plot(Prec~year, data = Fort, pch = 16, col="darkblue",
    xlab="", ylab = "Daily Precipitation (inches)")
plot(Prec~year, data = Fort, type = "l", col="darkblue",
    xlab="day", ylab = "Daily Precipitation (inches)")
```

Figure 11 was created using the following commands.

```
par(mfrow = c(2,2), mar = c(4.1, 5.1, 2.1, 1.1))
plot(MinT~Year, data = Tphap, pch = 16, col = "darkorange", xaxt = "n",
```

```
    xlab = "", ylab = "Daily Minimum Temperature (deg F)")
axis(1, at = pretty(Tphap$Year),
    labels = paste("19", pretty(Tphap$Year), sep = ""))
plot(Dam~Year, data = damage, pch = 16, col = "darkred",
    xlab = "",
    ylab = "Estimated Economic Damage\nfrom Hurricanes (billions USD)")

plot(USDMG~HYEAR, data = Flood, pch = 16, col = "darkred",
    xlab = "", ylab = "Total Economic Damage\nfrom floods (billions USD)")

plot(Ct~Year, data = Rsum, type = "h", col = "darkgreen",
    xlab = "", ylab = "Number of Hurricanes per year")
```

Finally, Figure 12 was made as follows.

```
par(mfrow = c(1, 2), mar = c(5, 5, 7, 2))
plot(TMX1~Year, data = PORTw, pch = 16, col = "lightblue",
    ylab = "Maximum winter temperature (deg C)\nPort Jervis, New York")

plot(TMX1~AOindex, data = PORTw, pch = 16, col = "lightblue",
    ylab = "")
```

Acknowledgments

Support for the production of this document and for the development and maintenance of the `extRemes` and `in2extRemes` software packages was provided by the Weather and Climate Impacts Assessment Science Program (http://www.assessment.ucar.edu) led by Linda O. Mearns and funded by the National Science Foundation. Any opinions, findings, and conclusions or recommendations expressed in this publication are those of the authors and do not necessarily reflect the views of the National Science Foundation.

References

Beirlant, J., Y. Goegebeur, J. Teugels, and J. Segers, 2004: *Statistics of Extremes: Theory and Applications*. Wiley Series in Probability and Statistics, Wiley, Chichester, West Sussex, England, U.K. 522 pp.

Coles, S., 2001: *An Introduction to Statistical Modeling of Extreme Values*. Springer-Verlag, London, UK, 208 pp.

de Haan, L. and A. Ferreira, 2006: *Extreme Value Theory: An Introduction*. Springer-Verlag, New York, N.Y., 288 pp.

Efron, B. and R. Tibshirani, 1993: *An introduction to the bootstrap*, volume 57. Chapman & Hall/CRC, Boca Raton, FL, 384pp., New York.
URL `http://www.loc.gov/catdir/enhancements/fy0730/93004489-d.html`

Ferro, C. A. T. and J. Segers, 2003: Inference for clusters of extreme values. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, **65**, 545–556.

Fox, J. and S. Weisberg, 2011: *An R companion to applied regression*. SAGE Publications, Thousand Oaks, California, U.S.A., 2nd ed. edition.

Gilleland, E., 2010: *Confidence intervals for forecast verification*. NCAR/TN-479+STR, 71pp.

— 2013: *distillery: Method Functions for Confidence Intervals and to Distill Information from an Object*. R package version 1.0-1.
URL `http://CRAN.R-project.org/package=distillery`

Gilleland, E. and R. W. Katz, 2011: New software to analyze how extremes change over time. *Eos*, **92**, 13–14.

— 2014: extRemes 2.0: An Extreme Value Analysis Package in R. *Submitted to the Journal of Statistical Software*.

Gilleland, E., M. Ribatet, and A. G. Stephenson, 2013: A software review for extreme value analysis. *Extremes*, **16**, 103–119.

Guttorp, P. and V. N. Minin, 1995: *Stochastic Modeling of Scientific Data*. Chapman & Hall/CRC, Boca Raton, FL, 384pp.

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

Jenkinson, A. F., 1955: The frequency distribution of the annual maximum (or minimum) of meteorological elements. *Quant. J. R. Met. Soc.*, **81**, 158–171.

Karvanen, J., 2011: *Lmoments: L-moments and quantile mixtures. R*. Package version 1.1-3.
URL http://CRAN.R-project.org/package=Lmoments

Katz, R. W., 2002: Stochastic modeling of hurricane damage. *Journal of Applied Meteorology*, **41**, 754–762.

Katz, R. W., G. S. Brush, and M. B. Parlange, 2005: Statistics of extremes: modeling ecological disturbances. *Ecology*, **86**, 1124–1134.

Katz, R. W., M. B. Parlange, and P. Naveau, 2002: Statistics of extremes in hydrology. *Water Resources Research*, **25**, 1287–1304.

Lahiri, S., 2003: *Resampling Methods for Dependent Data*. Springer-Verlag, New York.

Leadbetter, M. R., G. Lindgren, and H. Rootzén, 1983: *Extremes and related properties of random sequences and processes*. Springer-Verlag, New York, NY, 352pp.

Pielke, J., Roger A. and M. W. Downton, 2000: Precipitation and damaging floods: Trends in the United States, 1932–97. *J. Climate*, **13**, 3625–3637.

R Core Team, 2013: *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria.
URL http://www.R-project.org/

Reiss, R. and M. Thomas, 2007: *Statistical Analysis of Extreme Values: with applications to insurance, finance, hydrology and other fields*. Birkhäuser, 530pp., 3rd edition.

Resnick, S. I., 1987: *Extreme Values, Point Processes and Regular Variation*. Springer-Verlag, New York, NY, 320pp.

— 2007: *Heavy-tail phenomena: probabilistic and statistical modeling*. Springer series in operations research and financial engineering, Springer, New York, N.Y.
URL http://www.loc.gov/catdir/enhancements/fy0824/2006934621-d.html

Smith, R. L., 1989: Extreme value analysis of environmental time series: An application to trend detection in ground-level ozone (with discussion). *Statistical Science*, **4**, 367–393.

Smith, R. L. and T. S. Shively, 1995: A point process approach to modeling trends in tropospheric ozone. *Atmospheric Environment*, **29**, 3489–3499.

Thompson, D. W. J. and J. M. Wallace, 1998: The Arctic Oscillation signature in the wintertime geopotential height and temperature fields. *Geophysical Research Letters*, **25**, 1297–1300.

von Mises, R., 1936: La distribution de la plus grande de n valeurs. *Rev. Math. Union Interbalcanique*, **1**, 141–160.

Wettstein, J. J. and L. O. Mearns, 2002: The influence of the North Atlantic-Arctic Oscillation on mean, variance and extremes of temperature in the northeastern United States and Canada. *Journal of Climate*, **15**, 3586–3600.

Index