Extremes Toolkit

Weather and Climate Applications of Extreme Value Statistics

Eric Gilleland, Rick Katz & Greg Young

\[1\]

\[1\]National Center for Atmospheric Research (NCAR). This toolkit is funded by the National Science Foundation (NSF) through the NCAR Weather and Climate Impact Assessment Science Initiative, with additional support from the NCAR Geophysical Statistics Project (GSP). Initial work on the toolkit was performed by Greg Young. We thank Stuart Coles for permission to use his S-PLUS functions.
**Summary:** The Extremes Toolkit is designed to facilitate the use of extreme value theory in applications oriented toward weather and climate problems that involve extremes, such as the highest temperature over a fixed time period. This effort is motivated by the continued use of traditional statistical distributions (normal, lognormal, gamma, ...) in situations where extreme value theory is applicable. The goal is to write a GUI prototype to interact with a high-level language capable of advanced statistical applications. Computational speed is secondary to development time. With these guidelines, the language R was chosen in conjunction with a Tcl/Tk interface. R is a GNU-license product available at [www.r-project.org](http://www.r-project.org). Tcl/Tk is a popular GUI development platform also freely available for Linux, Unix and the PC (see section 8.3 for more details).

While the software can be used without the graphical interface, beginning users of R will probably want to start by using the GUI. If its limitations begin to inhibit, it may be worth the investment to learn the R language. The majority of the code was adapted by Alec Stephenson from routines by Stuart Coles. Coles’ book [3] is a useful text for further study of the statistical modeling of extreme values.

This toolkit and tutorial do not currently provide for fitting models for multivariate extremes or spatiotemporal extremes. Such functionality may be added in the future, but no plans currently exist and only univariate methods are provided.

**Hardware requirements:** Tested on unix/Linux and Windows 2000

**Software requirements:** R (version 1.7.0 or greater) and Tcl/Tk (included with R >= 1.7.0 for Windows)
**Abreviations and Acronymns**

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Full Form</th>
</tr>
</thead>
<tbody>
<tr>
<td>GEV</td>
<td>Generalized Extreme Value</td>
</tr>
<tr>
<td>GPD</td>
<td>Generalized Pareto Distribution</td>
</tr>
<tr>
<td>MLE</td>
<td>Maximum Likelihood Estimator</td>
</tr>
<tr>
<td>POT</td>
<td>Peaks Over Threshold</td>
</tr>
<tr>
<td>PP</td>
<td>Point Process Model</td>
</tr>
</tbody>
</table>
Contents

1 Starting the Extremes Toolkit ................................................. 1

2 Data ...................................................................................... 1
   2.1 Loading a dataset ............................................................. 1
   2.2 Simulating data from a GEV distribution ......................... 6
   2.3 Simulating data from a GPD distribution ......................... 12

3 Block Maxima Approach ....................................................... 14
   3.1 Fitting a GEV distribution ............................................. 15
   3.2 Return level and shape parameter ($\xi$) (1-\(\alpha\))% confidence limits .......... 19
   3.3 Fitting a GEV distribution with a covariate ...................... 20

4 Frequency of Extremes ......................................................... 23
   4.1 Fitting a Poisson distribution ........................................ 23
   4.2 Fitting a Poisson distribution with a covariate ............... 24

5 Fitting a Generalized Pareto Distribution (GPD) ................. 25
   5.1 Fitting a GPD .............................................................. 25
   5.2 Return level and shape parameter ($\xi$) (1-\(\alpha\))% confidence bounds .......... 34
   5.3 Threshold Selection: Mean Residual Life Plot .................. 35
   5.4 Threshold Selection: Fitting GPD Over a Range of Thresholds ...... 37

6 Peaks Over Threshold (POT)/Point Process (PP) Approach ...... 37
   6.1 Fitting a Point Process Model ....................................... 39
   6.2 Nonconstant Thresholds .............................................. 43

7 Extremes of Dependent and/or Nonstationary Sequences ......... 49
   7.1 Declustering ............................................................. 51
   7.2 Incorporating nonstationarity into a point process approach ...... 53

8 Details .................................................................................... 60
   8.1 Trouble Shooting ........................................................ 60
   8.2 Is it Really Necessary to Give a Path to the library Command Every Time? 61
   8.3 Software Requirements .............................................. 62
   8.4 The Underlying Functions .......................................... 63
   8.5 Miscellaneous ............................................................ 64

A Appendix A: Generalized Extreme Value distribution ............. 64
B  Appendix B: Threshold Exceedances 66
   B.1  Generalized Pareto Distribution 66
   B.2  Peaks Over Threshold (POT)/Point Process (PP) Approach 66
   B.3  Selecting a Threshold 67

C  Appendix C: Dependence Issues 67
   C.1  Probability and Quantile Plots for Non-stationary Sequences 67
1 Starting the Extremes Toolkit

To start the Extremes Toolkit, open an R session and from the R prompt, type

> library( extRemes)

The main Extremes Toolkit dialog should now appear. If it does not appear, please see section 8.1 to troubleshoot the problem. If at any time while extRemes is loaded this main dialog is closed, it can be re-opened by the following command.

> extremes.gui()

2 Data

The Extremes Toolkit allows for both the reading in of existing datasets and for the simulation of values from the generalized extreme value (GEV) distribution.

2.1 Loading a dataset

The general outline for reading in a dataset to the extreme value toolkit is

• File -> Read Data -> New window appears
• Browse for file and Select -> Another new window appears
• Enter options -> assign a Save As (in R) name -> OK -> Status message displays.
• The data should now be loaded in R as an ev.data list object.

There are two general types of datasets that can be read in using the toolkit. One type is referred to here as common and the other is R source. Common data can take many forms as long as any headers do not exceed one line and the rows are the observations and the columns are the variables. For example, table 1 represents a typical common dataset; in this case data representing U.S. flood damage. See Pielke and Downton [7] or Katz et al. [6] for more information on these data.

An R source dataset is a dataset that has been dumped from R and 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. However, it is useful for packages to include these types of datasets for examples. It is easy to decipher if a dataset is an R
Table 1: U.S. total economic damage (in billion $) due to floods (USDMG) by year from 1932-1997. Also gives damage per capita (DMGPC) and damage per unit wealth (LOSSPW). See Pielke and Downton [7] for more information.

source file or not. For example, the same dataset in table 1 would look like the following.

```r
“Flood” <-
structure(list(OBS = c(1, 2, 3, 4, ..., 64, 65, 66),
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, 22.5410, 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”))
```

Data loaded in by the Extremes Toolkit is loaded as a list object with class attribute “ev.data”. A list object is a convenient way to collect and store related information in R. A list object can store different types of objects in separate components. For example, a character vector, a matrix, a function and maybe another matrix can all be stored as components in the same list object. When data is first loaded into the toolkit, it has three components: `data`, `name` and `file.path`. `data` is the actual data read in (or simulated), `name` is a character string giving the original file name, for example “Flood.dat”, and `file.path` is a character string giving the full path where the data was read from. When data is fit to a particular model, say a GEV distribution, then there will be a new component called `models` in the original list object. This new component is also a list whose components will include each fit. Specifically, each GEV fit will be assigned the name “gev.fit1”, “gev.fit2” and so on, where the first fit is “gev.fit1”, the second “gev.fit2”, etc... Component names of a list object can be found by using the R function `names` as shown in the example below.
To look at components of a list, type the list name followed by a dollar sign followed by the component name. For example, if you have a list object called George with a component called finance, you can look at this component by typing \texttt{George$finance} at the R prompt.

**Example 1: Loading a Common Dataset**

Here we will load the common dataset, \texttt{Flood.dat}, which should be located in the Extremes Toolkit data directory—depending on the system, it may be necessary to unzip the data files into the directory. Throughout this tutorial it is assumed that the `.dat` files are present in your version. However, if not, the data files with the `.r` or `.R` extensions are equivalent, but note that they are \texttt{R source} files. From the main toolkit dialog, select \texttt{File $\rightarrow$ Read Data}. A new window appears for file browsing. Go to the Extremes Toolkit (\texttt{extRemes}) data directory and select the file \texttt{Flood.dat}; another new window will appear that allows you to glance at the dataset (by row) and has some additional options. That is,

- \texttt{File $\rightarrow$ Read Data $\rightarrow$ New window appears.}
- \texttt{Browse for file \texttt{Flood.dat} $\rightarrow$ Open $\rightarrow$ Another new window appears.}

Leave the \texttt{Common} radiobutton checked and because the columns are separated by white space, leave the delimiter field blank; sometimes datasets are delimited by other symbols like commas “,” and if that were the case it would be necessary to put a comma in this field. Check the \texttt{Header} checkbutton because this file has a one line header. Files with headers that are longer than one line cannot be read in by the toolkit. Enter a \texttt{Save As} name, say \texttt{Flood}, and click \texttt{OK}. A message in the main toolkit window should display that the file was read in correctly. The steps for this example, once again, are:

- \texttt{File $\rightarrow$ Read Data $\rightarrow$ New window appears.}
- \texttt{Browse for file \texttt{Flood.dat} $\rightarrow$ Open $\rightarrow$ Another new window appears.}
- \texttt{Check \texttt{Header}}
- \texttt{Enter \texttt{Flood} in \texttt{Save As (in R) field} $\rightarrow$ OK.}
- \texttt{Message appears saying that file was successfully opened.}

Figure 1 shows a time series plot of one of the variables from these data, \texttt{USDMG}. Such a plot can be easily created using the toolkit. The general procedure is:

- \texttt{Plot $\rightarrow$ Time Series $\rightarrow$ New dialog window appears.}
- \texttt{Select \texttt{Flood} from \texttt{Data Object} field}
Figure 1: Time series plot of total economic damage from U.S. floods (in billion $).
• Select HYEAR from x-axis variable field

• Select USDMG from y-axis variable field –> OK

• Time series is plotted in a new window (Windows users may have to minimize other windows in order to see plot).

To see the names of the list object created use the R function names. That is,

> names( Flood)
[1] "data" "name" "file.path"

To look at a specific component, say name, do the following.

> Flood$name
[1] “Flood.dat”

To look at the first three rows of the flood data, do the following.

> Flood$data[1:3,]

**Example 2: Loading an R source Dataset**

The data used in this example were provided by Linda Mearns of NCAR. The file PORTw.R consists of maximum winter temperature values for Port Jervis, NY. While the file contains other details of the dataset, the maximum temperatures are in the seventh column, labeled “TMX1”. See Wettstein and Mearns [12] for more information on these data.

The first step is to read in the data. From the main window labeled “Extremes Toolkit”, select

File –> Read Data .

An additional window will appear that enables the browsing of the directory tree. Find the file PORTw.R, located in the data directory of the extRemes library. Highlight it and press the Select button.

• File –> Read Data -> New window appears

• Browse for file PORTw.R in extRemes data directory and Select

Another window will appear displaying the contents of the file and providing various options. Because these example data are R source data, check the radiobutton for R source under File type. R source datasets do not have headers or delimiters and these options can be ignored here.
• File -> Read Data ->

• Browse for file PORTw.R and Select ->

• Under File Type Check R source

For this example, enter the name PORT into the Save As (in R) field and click OK to load the dataset.

• File -> Read Data ->

• Browse for file and Select ->

• Under File Type Check R source ->

• Type PORT in Save As (in R) field -> OK

The status section of the Extremes Toolkit window displays the message that the file was successfully read. To verify this or to simply look at the dataset, select

File -> View Data

This routine brings up another window displaying the current dataset. Note that if no column names are contained in the file, each column will be labeled with “V” and a numerical index (as this is the convention in both R and S).

2.2 Simulating data from a GEV distribution

A fundamental family of distributions in extreme value theory are the generalized extreme value (GEV) family of distributions. To learn more about this class of distributions see section A.

The general procedure for simulating (from a GEV) data is:

• File -> Simulate Data -> Generalized Extreme Value (GEV) ->

• Enter options and a Save As name -> Generate -> Plot of simulated data appears

• The simulated dataset will be saved as an ev.data object.

In order to generate a dataset by sampling from a GEV, select

File -> Simulate Data -> Generalized Extreme Value (GEV)
from the main Extremes Toolkit window. The simulation window displays several options specific to the GEV. Namely, the user is able to specify the location (mu), the scale (sigma) and shape (xi) parameters. In addition, a linear trend in the location parameter may be chosen as well as the size of the sample to be generated. As discussed in section 2.1, it is a good idea to enter a name in the Save As field. After entering the options, press the Generate button to generate and save a simulated dataset. The status section of the main window displays the parameter settings used to sample the data and a plot of the simulated data, such as in figure 2, is produced.

For example, simulate a dataset from a GEV distribution (using all the default values) and save it as gevsim1. That is,

- File \(\rightarrow\) Simulate Data \(\rightarrow\) Generalized Extreme Value (GEV)

- Enter gevsim1 in the Save As field \(\rightarrow\) Generate

- Plot appears, message on main toolkit window displays parameter choices and an object of class “ev.data” is saved.

Once a dataset has been successfully loaded or simulated, work may begin on its analysis.
The Extremes Toolkit provides for fitting GEV, Poisson and generalized Pareto (GPD) distributions as well as fitting the GEV indirectly by the point process (PP) approach. For the above example, fit a GEV distribution to the simulated data. Results will differ from those shown here as the data is generated randomly each time. To fit a GEV to the simulated data, do the following.

- **Analyze** -> **Fit GEV** -> *New window appears*
- **Select** `gevsim1` *from the Data Object field.*
- **Select** `gev.sim` *from the Response field.*
- **Check the** *Plot diagnostics* *button.* -> **OK**

A plot similar to the one in figure 3 should appear. For information on these plots please see section 3.1. Briefly, the top two plots should not deviate much from the straight line and the histogram should match up with the curve. The return level plot gives an idea of the expected return level for each return period. The maximum likelihood estimates (MLE) for the parameters of the fit shown in figure 3 were found to be \( \hat{\mu} \approx -0.31 \) (0.15), \( \hat{\sigma} \approx 0.9 \) (0.13) and \( \hat{\xi} \approx 0.36 \) (0.15) with a negative log-likelihood value for this model of approximately 84.07. Again, these values should differ from values obtained for different simulations. Nevertheless, the location parameter, \( \mu \), should be near zero, the scale parameter, \( \sigma \), near one and the shape parameter, \( \xi \), near 0.2 as these were the parameters of the true distribution from which the data was simulated. An inspection of the standard errors for each of these estimates (shown in parentheses above) reveals that the location parameter is two standard deviations below zero, the scale parameter is well within the first standard deviation from one and the shape parameter is only about one standard deviation above 0.2, which is quite reasonable.

It is also possible to incorporate a linear trend in the location parameter when simulating from a GEV distribution using this toolkit. That is, it is possible to simulate a GEV distribution with a nonconstant location parameter of the form \( \mu(t) = \mu_0 + \mu_1 t \), where \( \mu_0 = 0 \) and \( \mu_1 \) is specified by the user. For example, to simulate from a GEV with \( \mu_1 = 0.3 \) do the following.

- **File** -> **Simulate Data** -> **Generalized Extreme Value (GEV)**
- **Enter** *0.3 in the Trend field and gevsim2 in the Save As field* -> **Generate.**

The trend should be evident from the scatter plot. Now, first fit the GEV without a trend in the location parameter.

- **Analyze** -> **Fit GEV**
Figure 3: Diagnostic plots for GEV fit to a simulated dataset.
Figure 4: Simulated data from GEV distribution with trend in location parameter fit to GEV distribution without a trend.

- Select `gevsim2` from the Data Object field.
- Select `gev.sim` from the Response field.
- Check the Plot diagnostics button. → OK.

A plot similar to that of figure 4 should appear. As expected, it is not an exceptional fit.

Next fit these data to a GEV, but with a trend in the location parameter.

- Analyze → Fit GEV
- Select `gevsim2` from the Data Object field.
- Select `gev.sim` from the Response field.
- Select `obs` from the Location Parameter (mu) field (leave identity as link function).
- Check the Plot diagnostics button. → OK.
Figure 5: Simulated data from GEV distribution with trend in location parameter fit to GEV distribution with a trend.

Notice that only the top two diagnostic plots are plotted when incorporating a trend into the fit as in figure 5. The fit appears, not surprisingly, to be much better. In this case, the MLE for the location parameter is \( \hat{\mu} \approx 0.27 + 0.297 \cdot t \), where \( t \) is the index variable, \( \text{obs} \), and associated standard errors are 0.285 and 0.01 respectively; both of which are well within one standard deviation of the true values (\( \mu_0 = 0 \) and \( \mu_1 = 0.3 \)) that we used to simulate this dataset. Note that these values should be slightly different for different simulations, so your results will likely differ from these here. Values for this particular simulation for the other parameters were also within one standard deviation of the true values.

A more analytic method of determining the better is to do a likelihood-ratio test. Using the toolkit try the following.

- **Analyze** \( \rightarrow \) **Likelihood-ratio test**
  - **Select** \( \text{gevsim2} \) \( \text{ from the Data Object field.} \)
  - **Select** \( \text{gev.fit1} \) \( \text{ from the Select base fit (M0) field.} \)
  - **Select** \( \text{gev.fit2} \) \( \text{ from the Select comparison fit (M1) field \( \rightarrow \) OK} \).
In the case of the data simulated here, the likelihood-ratio test overwhelmingly supports, as expected, the model incorporating a trend in the location parameter with a likelihood ratio of about 117 compared with a 0.95 quantile of the $\chi^2_1$ distribution of only 3.8415 and p-value approximately zero.

2.3 Simulating data from a GPD distribution

It is also possible to sample from a Generalized Pareto Distribution (GPD) using the toolkit. For more information on the GPD please see section 5.1. The general procedure for simulating from a GPD is as follows.

- **File** $\rightarrow$ **Simulate Data** $\rightarrow$ **Generalized Pareto** (GP)
- _Enter options and a Save As name_ $\rightarrow$ **Generate**
- A scatter plot of the simulated data appears and a message on main toolkit window displays chosen parameter values and an object of class “ev.data” is created.

Figure 6 shows the scatter plot for one such simulation. As an example, simulate a GP dataset in the following manner.

- **File** $\rightarrow$ **Simulate Data** $\rightarrow$ **Generalized Pareto** (GP)
- _Leave the parameters on their defaults and enter gpd1sim in the Save As field_ $\rightarrow$ **Generate**
- A scatter plot of the simulated data appears and a message on main toolkit window displays chosen parameter values and an object of class “ev.data” is created.

You should see a plot similar to that of figure 6, but not the same as each simulation will yield different values. The next step would be to fit a GPD to these simulated data. To fit a GP to these data, do the following.

- **Analyze** $\rightarrow$ **Generalized Pareto Model**
- _Select gpd1sim from the Data Object field._
- _Select gpd.sim from the Response field._
- _Check Plot diagnostics button_
- _Enter zero in the Threshold field_ $\rightarrow$ **OK**
Figure 6: Scatter plot of one simulation from a GPD using the default values for parameters.
Plots similar to those in Figure 7 should appear, but results will vary for each simulated set of data. Results from one simulation had the following MLE’s for parameters (with standard errors in parentheses): $\hat{\sigma} \approx 1.14$ (0.252) and $\hat{\xi} \approx 0.035$ (0.170). As with the GEV example these values should be close to those of the default values chosen for the simulation. In this case, the scale parameter is well within one standard deviation from the true value and the shape parameter is nearly one standard deviation below its true value.

Note that we used the default selection of a threshold of zero. It is possible to use a different threshold by entering it in the Threshold field. The result is the same as adding a constant (the threshold) to the simulated data.

3 Block Maxima Approach

One approach to working with extreme value data is to group the data into blocks of equal length and fit the data to the maximums of each block, for example, annual maxima of daily precipitation amounts. The choice of block size can be critical as blocks that are too small can lead to bias and blocks that are too large generate too few block maxima, which
leads to large estimation variance [3]. The block maxima approach is closely associated with
the use of the GEV family. Note that all parameters are always estimated by maximum
likelihood (MLE), which requires iterative numerical optimization techniques. See Coles [3]
section 2.6 on parametric modeling for more information on this optimization method.

3.1 Fitting a GEV distribution

The general procedure for fitting a GEV distribution with the Extremes Toolkit is

- **Analyze** -> **Fit GEV** -> New window appears

- **Select data object from Data Object field** -> column names appear in various fields

- **Choose a response variable from the Response field** -> Response variable is removed
  as an option from other fields

- **Select other options as desired** -> **OK**

- A GEV distribution will be fit to the chosen response variable and stored in the same
  list object as the data used.

**Example 1: Port Jervis data**

Here we shall use the PORT dataset read in as per section 2.1 to illustrate fitting a GEV
using the toolkit. If you have not already loaded these data, please do so before trying this
example. Refer to section 2.1 for help loading them. Figure 8 shows a time series of the
annual (winter) maximum temperatures (degrees centigrade).

From the main window, select

**Analyze** -> **Fit GEV**

A new dialog window appears requesting the details of the fit. First, select PORT from the
Data Object field. Immediately, the fields for Response, Location parameter (mu),
Scale parameter (sigma) and Shape parameter (xi) should now contain the list of
covariates for these data.

- **Analyze** -> **Fit GEV** -> New window appears

- **Select data object from Data Object field** -> column names appear in various fields

In the Response field, click the left mouse button on the desired response variable, in
this case TMX1—corresponding to the annual maximum temperature. Immediately, this
variable is removed as a predictor from the fields: Location parameter (mu), Scale
parameter (sigma) and Shape parameter (xi).
Figure 8: Time series of Port Jervis annual (winter) maximum temperature (degrees centi-grade).
Choose TMX1 from the Response field → TMX1 is removed as an option from other fields

Click on the Plot diagnostics button, and for now ignore the rest of the fields and simply click on the OK button. An R graphics window appears displaying two goodness-of-fit plots, the probability plot and quantile plot along with a return-level plot and a density estimate plot as shown in figure 9. In the case of perfect fit, the data would line up on the diagonal of the probability and quantile plots.

Briefly, the quantile plot is a plot of the model quantiles against the data (empirical) quantiles. This is why a perfect fit would give a perfectly straight line. The more the quantile plot deviates from a straight line, the worse the fit. The return level plot plots the return period against the return level and also includes 95% confidence interval. The return level is the level (in this case temperature) that is expected to occur, on average, once every \( m \) time points (in this case years). The return period is the amount of time expected to wait for the exceedance of a particular return level. For example, in figure 9, one would expect the maximum winter temperature for Port Jervis to exceed about 24 degrees centigrade on average every 100 years. Refer to Coles [3] chapter 3 for more details.

Plot diagnostics → OK → GEV is fit and plot diagnostics displayed

In the status section of the main window, several details of the fit are displayed. The maximum likelihood estimates of each of the parameters are given, along with their respective standard errors. In this case, \( \mu \) has a maximum likelihood estimate (MLE) of about 15.14 (0.39745), \( \sigma \) is about 2.97 (0.27523) and \( \xi \) is about -0.22 (0.0744). The negative log likelihood for the model (172.7426) is also displayed.

Note that figure 9 can be redrawn by clicking on Fit Diagnostics from the Plot menu of the main toolkit dialog window.

Plot -> Fit Diagnostics

A new dialog appears. Click on PORT in the Data Object field and then select gev.fit from the Select a fit listbox. Finally, click OK. Note that if you fit another GEV distribution, say using covariates, to this dataset, the diagnostic plots will reflect the most recent fit.

It may be of interest to incorporate a covariate into one or more of the parameters of the GEV. For example, the dominant mode of large-scale variability in mid-latitude North-
Figure 9: GEV fit diagnostics for Port Jervis winter maximum temperature dataset. Quantile plot and return level plot units are degrees centigrade.
ern Hemisphere temperature variability is the North Atlantic Oscillation-Arctic Oscillation (NAO-AO). Such a relationship should be investigated by including these indices as a covariate in the GEV. The next section 3.3 explores the inclusion of one of these variables as a covariate.

3.2 Return level and shape parameter ($\xi$) (1-$\alpha$)% confidence limits

Confidence intervals may be estimated using the toolkit for both the $m$-year return level and shape parameter ($\xi$) of both the GEV and GP distributions. The estimates are based on finding the intersection between the respective profile likelihoods and $\frac{1}{2}c_{1,\alpha}$, where $c_{1,\alpha}$ is the $1-\alpha$ quantile of a $\chi^2_1$ distribution (see Coles [3] section 2.6.5 for more information). The general procedure for estimating confidence limits for return levels and shape parameters of the GEV distribution with the toolkit is as follows.

- **Analyze** -> Parameter Confidence Intervals -> GEV fit
- **Select an object from the Data Object field.**
- **Select a fit from the Select a fit field.**
- **Enter search limits for both return level and shape parameter ($\xi$) (and any other options) -> OK**

**Example: Port Jervis Data Continued**

MLE estimate for 100-year return levels in the above GEV fit for the Port Jervis data are found to be somewhere between 20 and 25 degrees (using the return level plot) and $\hat{\xi} \approx -0.2$. We will use these values to help us find a reasonable search range for estimating the confidence limits. To find confidence limits, do the following.

- **Analyze** -> Parameter Confidence Intervals -> GEV fit
- **Select PORT from the Data Object field.**
- **Select gev.fit1 from the Select a fit field.**
- **Enter 22 in the Lower limit of the Return Level Search Range and 28 in the Upper limit field.**
- **Enter $-0.4$ in the Lower limit of the Shape Parameter ($\xi$) Search Range and $0.1$ in the Upper limit field -> OK.**

Estimated confidence limits should now appear in the main toolkit dialog. In this case, the estimates are given by about 22.42 to 27.18 degrees for the 100-year return level.
and about -0.35 to -0.05 for $\xi$ indicating that this parameter is significantly below zero. Of course, it is also possible to find limits for other return levels (besides 100-year) by changing this value in the m-year return level field. Also, the profile likelihood plots, as shown in figure 10, can be plotted by clicking on the check button for this feature. In this case, our estimates seem reasonable because the dashed vertical lines appear to intersect the likelihood at the same point as the lower horizontal line in both cases.

### 3.3 Fitting a GEV distribution with a covariate

The general procedure for fitting a GEV distribution with a covariate is similar to that of fitting a GEV without a covariate, but with one additional step. The procedure is:

- **Analyze -> Fit GEV -> New window appears**
- **Select data object from Data Object field -> column names appear in various fields**
- **Choose a response variable from the Response field -> Response variable is removed as an option from other fields**
• Select variables to use as covariates from Location parameter (mu), Scale parameter (sigma) and/or Shape parameter (xi) fields

• select which link function to use for each of these choices → OK

• A GEV distribution will be fit to the chosen response variable and stored in the same list object as the data used.

**Example 2: Port Jervis data with a covariate**

To demonstrate the ability of the Toolkit to use covariates, we shall continue with the Port Jervis data and fit a GEV on TMX1, but with the Atlantic Oscillation index, AOindex, as a covariate with a linear link to the location parameter. See Wettstein and Mearns [12] for more information on this index.

**Analyze → Fit GEV.**

Select PORT from the Data Object field. Again, select TMX1 from the Response field. Select AOindex from the Location parameter (mu) list. Keep the link as identity and press OK.

• Select PORT from Data Object field → variables now listed in some other fields

• Select TMX1 from the Response field → TMX1 removed from other fields

• Select AOindex from Location parameter (mu) list → OK

• A GEV fit on the Port Jervis data is performed with AOindex as a covariate in the location parameter.

The status window now displays information similar to the previous example, with one important exception. Underneath the estimate for MU (now the intercept) is the estimate for the covariate trend in mu as modeled by AOindex. In this case,

\[ \mu = 15.25 + 1.15 \cdot AOindex \]

Figure 11 shows the diagnostic plots for this fit. Note that only the probability and quantile plots are displayed. Note that the quantile plot is plotted on the Gumbel scale. See the appendix section C.1 for more details. Note also that the values plotted are transformed variables, please see appendix C.1 for more on this as well.

A test can be performed to determine if this model with AOindex as a covariate is an improvement over the previous fit without a covariate using the toolkit. Specifically, the test compares the likelihood-ratio, \( D = 2 \cdot (l_1 - l_0) \), where \( l_0 \) and \( l_1 \) are the likelihoods for each
Figure 11: GEV fit diagnostics for Port Jervis winter maximum temperature dataset with AOindex as a covariate. Both plots are generated using transformed variables and therefore the units are not readily interpretable. See section C.1 for more details.
of the two models ($l_0$ must be nested in $l_1$), to a $\chi^2$ quantile, where $\nu$ is the difference in the number of estimated parameters. In this case, we have three parameters estimated for the example without a covariate and four parameters for the case with a covariate because $\mu = b_0 + b_1 \cdot AOindex$ giving us the new parameters: $b_0$, $b_1$, $\sigma$ and $\xi$. So, for this example, $\nu = 4 - 3 = 1$. See Coles [3] for details on this test. Note that the model without a covariate was stored as `gev.fit1` and the model with a covariate was stored as `gev.fit2`; each time a GEV is fit using this data object, it will be stored as `gev.fitN`, where $N$ is the $N$th fit performed. The general procedure is:

- **Analyze** -> **Likelihood-ratio test** -> *New window appears.*

- Select a data object. In this case, PORT from the **Data Object** field. Values are filled into other fields.

- Select fits to compare. In this case, `gev.fit1` from **Select base fit (M0) field** and `gev.fit2` from **Select comparison fit (M1) field** -> OK.

- Test is performed and results displayed in main toolkit window.

For this example, the likelihood-ratio is about 11.89, which is greater than the 95% quantile of the $\chi^2_1$ distribution of 3.8415, suggesting that the covariate $AOindex$ model is a significant improvement over the model without a covariate. The p-value of 0.000565 further supports this claim.

In addition to specifying the covariate for a given parameter, the user also has the ability to indicate what type of link function should relate that covariate to the parameter. The two available link functions (identity and exponential) are indicated by the buttons to the right of the covariate list boxes. This example used the *identity* link function.

### 4 Frequency of Extremes

Often it is of interest to look at the frequency of extreme event occurrences. Similar results apply to the frequency of extremes as applies to occurrences. That is, under regularity conditions, the frequency of extreme events approaches a Poisson process. See section B.2 for more details.

#### 4.1 Fitting a Poisson distribution

The Extremes Toolkit also provides for fitting the Poisson distribution, although not in the detail available for the GEV distribution. The Poisson distribution is useful for data that involves random sums of rare events. For example, a dataset containing the numbers of
A frequency of extremes per year is included with this toolkit named \emph{Rsum.dat}.

**Analyze \rightarrow Fit Poisson.**

A window appears for specifying the details of the model, just as in the GEV fit. Without a trend in the mean, only the rate parameter is currently estimated; in this case the maximum likelihood estimate (MLE) for the rate parameter is simply the mean of the data. If a covariate is given, the generalized linear model fit is used from R (see R documentation for \texttt{glm}). Currently, this toolkit provides only for fitting Poissons with the “log” link function.

**Example: Hurricane Count Data**

Load the Extremes Toolkit dataset \emph{Rsum.dat} as per section 2.1 and save it as \texttt{Rsum}.

This dataset gives the number of hurricanes per year (from 1925 to 1995) as well as the ENSO state and total monetary damage. More information on these data can be found in Pielke and Landsea [8] or Katz [4]. A simple fit without a trend in the data is performed in the following way.

- **Analyze \rightarrow Fit Poisson \rightarrow New window appears.**
- **Select \texttt{Rsum} from Data Object field.**
- **Select \texttt{Ct} from Response field \rightarrow OK.**
- MLE for rate parameter (\(\lambda\)) along with the variance and \(\chi^2\) test for equality of the mean and variance is displayed in the main toolkit window.

For these data, the MLE for the rate parameter is found to be 1.817, indicating that on average there were nearly two hurricanes 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, \(\lambda\). As per Katz [4], the estimated variance is shown to be 1.752, which is only slightly less than that of the mean (1.817). The \(\chi^2\) 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 of section 3.1, 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.

### 4.2 Fitting a Poisson distribution with a covariate

The general procedure for fitting a Poisson with a trend (using the Rsum dataset with ENSO state as a covariate) is as follows.
• **Analyze** –> **Fit Poisson** –> New window appears.

• **Select Rsum from Data Object field.**

• **Select Ct from Response field.**

• **Select EN from Trend variable field** –> **OK.**

• Fitted rate coefficients and other information are displayed in main toolkit window.

EN for this dataset represents the ENSO state. A plot of the residuals is created if the plot diagnostics checkbutton is engaged. The fitted model is found to be:

\[
\log(\lambda) = 0.575 - 0.25 \cdot \text{EN}
\]

with standard errors of 0.08999 and 0.114824 for the intercept and slope terms respectively. The slope term is significant at the 5% level indicating that the inclusion of the ENSO term as a covariate is reasonable.

5 Fitting a Generalized Pareto Distribution (GPD)

The Extremes Toolkit provides for fitting GPD models as well as some tools for threshold selection. Sometimes using only block maximum can be wasteful if it ignores much of the data. It is often more useful to look at exceedances over a given threshold instead of simply the maximum (or minimum) of the data. For more information on the GPD see section B.1.

Note: the r-th largest order statistic model has essentially been replaced by the following methods in practice, but the Extremes Toolkit does facilitate r-th largest model fitting as it is often desired for pedagogical reasons. For help on using the r-th largest model, see Coles [3] and [2]. This method is not covered in this tutorial.

5.1 Fitting a GPD

The general procedure for fitting a GPD with the Extremes Toolkit is:

• **Analyze** –> **Generalized Pareto Model** –> New window appears

• **Select a data object from Data Object field** –> Covariates appear in various fields

• **Select a response variable from Response field** –> Selected response is removed from other fields
Fitting a Generalized Pareto Distribution (GPD)

Figure 12: Time series of U.S. hurricane damage (in billions $ U.S.).

- Enter a threshold → other options → OK
- A GPD will be fit and results will appear in main toolkit window.

Example 1: Hurricane Damage

For this example, load the Extremes Toolkit dataset, damage.dat and save it as damage. Figure 12 shows the time series of these data from 1925 to 1995. The data are economic damage of hurricanes in billions of U.S. dollars. These data correspond to the count data discussed in section 4.1. To learn more about these data, please see Pielke and Landsea [8] or Katz [4]. The time series shows that there was a particularly large assessment of economic damage early on (in 1926) of 70 billion dollars. After this time, assessments are much smaller than this value.

Diagnostic plots for the GPD fit for these data with economic damage, Dam, as the response variable and a threshold of 6 billion dollars is shown in figure 13. The fit looks pretty good considering the one rather large outlier from 1926 and only 18 values over the threshold. A histogram similar to the one in figure 13 can be plotted by choosing Fit with Histogram from the Plot menu of the main toolkit dialog. From the new window that appears, select damage from the response field and select gpd.fit1 from the Select a fit.
FITTING A GENERALIZED PARETO DISTRIBUTION (GPD)

Figure 13: GPD fit for hurricane damage data using a threshold of 6 billion dollars.

field. Select a breaks algorithm and click OK. The histogram shown in figure 14 used the Friedman-Diaconis algorithm; the third choice in the breaks listbox. The histogram is still a little misleading in that it looks like the lower end point is at 5 billion dollars instead of 6 billion dollars and that it still does not appear to be a good fit to the GPD. In such a case, it is a good idea to play with the histogram in order to make sure that this appearance is not simply an artifact of the R function, hist, before concluding that it is a bad fit. In fact, the histogram shown in figure 15 looks better. It is currently not possible to produce this histogram directly from the toolkit. This histogram was produced in the following manner. From the R prompt:

```r
> range(damage$models$gpd.fit1$dat)
[1] 6.293 72.303
> brks <- seq(6.293, 72.303, .15)
> hist(damage$models$gpd.fit1, breaks=brks)
```

The estimated scale parameter, $\sigma$, for this distribution is about 4.6 billion dollars with a standard error of about 1.82 billion dollars. The estimated shape parameter, $\xi$, is about
0.5 (0.304). The model has an associated negative log-likelihood of about 54.65.

**EXAMPLE 2: FORT COLLINS PRECIPITATION DATA**

An example of a dataset where more information can be gathered using a threshold exceedance approach is the Fort Collins precipitation dataset. Read in the file *FtCoPrec.dat* from the data directory in the *extRemes* library and assign it to an object called *Fort*—it may take a few seconds to load this dataset.

- **File** -> **Read Data** -> *New window appears*
  - **Browse to extRemes data directory and select** *FtCoPrec.dat* *New window appears*
  - **Select common** from *Data Type* field ->
  - **Check the header checkbutton** ->
  - **Enter Fort in Save As (in R) field** -> OK

- Data will be read in as an *ev.data* object with the name *Fort*. It may take a few seconds to load this large dataset.
Figure 15: Histogram for GPD fit for hurricane damage data using a threshold of 6 billion dollars and a specialized vector for the breaks. See text for more details.
Figure 16 shows a scatter plot of the monthly daily maximum precipitation at this location. This plot can be made with the toolkit by selecting Plot from the toolkit main menu and then Scatter Plot. Select Fort and then month from the x-axis listbox and Prec from the y-axis listbox. Then click on the OK button.

- Plot -> Scatter Plot -> New window appears
- Select Fort from Data Object it field -> Covariates appear in other fields
- Select month from x-axis listbox and Prec from y-axis listbox -> OK
- Plot in figure 16 should appear.

To fit a GPD model using the toolkit, select Generalized Pareto Model from the Analyze menu of the main toolkit window. That is,
Analyze -> Generalized Pareto Model -> New window appears

Again, a new window will appear with several options. Select the Fort dataset from the Data Object field—the covariates year and Prec should now appear in the Response, Scale parameter (sigma) and Shape parameter (xi) fields.

Select Fort from Data Object field -> Covariates appear in other fields

Click on Prec in the Response field—Prec should now disappear from the Scale parameter (sigma) and Shape parameter (xi) fields.

Select Prec in Response field -> Prec is removed from other fields

Click on the Plot diagnostics checkbox and enter 0.40 into the Threshold field, which corresponds to 0.40 inches as was used by Katz et al. [6]. For now, we will ignore the rest of the fields and simply click OK.

Plot diagnostics -> Enter 0.40 in the Threshold field -> OK

A plot similar to that of figure 17 should appear along with summary statistics for the GPD fit in the main toolkit window. This fit yields maximum likelihood estimates for the scale parameter (sigma) of 0.33972 inches (standard error of 0.01644 inches) and for the shape parameter (xi) 0.18684 (0.0374) and a negative log-likelihood of about 110. Note that we are ignoring, for now, the annual cycle that is evident in figure 16.

Figure 17 can be reproduced at any time (provided another GPD fit is not performed on the Fort data object) by selecting Fit Diagnostics from the Plot menu of the main toolkit window.

Plot -> Fit Diagnostics

In the window that appears, click on Fort in the Data Object field, select gpd.fit1 from the Select a fit field and click OK.

Select Fort from Data Object field -> gpd.fit1 -> OK
Figure 17: Diagnostic plots for the GPD fit of the Fort Collins Precipitation data using a threshold of 0.40 in.
Figure 18: Histogram of GPD fit to Fort Collins precipitation (inches) data using the Friedman-Diaconis algorithm for determining the number of breakpoints.

Note that when fitting the GP model the default value for the number of observations per year is 365 days, which is appropriate for these data. Although this has no bearing on the GP fit, it is used when plotting the fit diagnostics.

Figure 18 shows a histogram of the data along with the model fit using the Friedman-Diaconis algorithm for binning (see the help file for hist in R for more details).

The general procedure for plotting a histogram of a fitted function using the Extremes Toolkit is:

- Plot $\rightarrow$ Fit with Histogram $\rightarrow$ New window appears
- Select an object from Data Object $\rightarrow$ Fitted model names appear in Select a fit field
- Select an algorithm from the Breaks Algorithm listbox and click OK
- Histogram is plotted.
5.2 Return level and shape parameter ($\xi$) (1-$\alpha$)% confidence bounds

Confidence intervals may be estimated using the toolkit for both the return level and shape parameter ($\xi$) of both the GEV and GP distributions. The estimates are based on finding the intersection between the respective profile likelihoods and $\frac{1}{2}c_{1,\alpha}$, where $c_{1,\alpha}$ is the $1 - \alpha$ quantile of a $\chi^2_1$ distribution (see Coles [3] section 2.6.5 for more information).

**EXAMPLE: Fort Collins precipitation data**

To estimate the confidence limits for the GPD shape parameter, select **GPD fit** from the **Parameter Confidence Intervals** menu of the **Analyze** menu of the main toolkit window. That is,

**Analyze $\rightarrow$ Parameter Confidence Intervals $\rightarrow$ GPD fit**

For the Fort Collins precipitation data the MLE for the 100-year return level is around 5 inches and for the shape parameter ($\xi$) it is about 0.19, so we want to search for the confidence limits over ranges that include these values and is wide enough to capture the actual limits. It may be necessary to play with the ranges here in order to get values that are reasonable. Specifically, if you get confidence values that are equal to one or more of your range values, then you will probably need to use a larger range. If you check the **plot profile likelihoods** checkbutton, you will obtain plots of the profile likelihoods along with vertical dashed lines where the estimated confidence limits are found. If these dashed lines intersect the profile plot at the same place as the lower horizontal line, then the estimate is reasonably accurate. For this example, we will use 3 to 7 inches for the 100-year return level and 0.1 to 0.3 for the shape parameter. ($\xi$).

- **Select Fort** from **Data Object field**
- **Select gpd.fit1** from **Select a fit field**
- **Leave the default values of 100 and 365 in the m-year return level and Number of obs. per year fields.**
- **Enter 4 in the Lower limit field of the Return Level Search Range and 7 in the Upper limit field.**
- **Enter 0.1 in Lower limit field of the Shape Parameter (xi) Search Range and enter 0.3 in the Upper limit field $\rightarrow$ OK**

Confidence intervals (in this case 95%) are shown in the main toolkit dialog. For the 100-year return level they are approximately (4.24, 6.82) inches and for the shape parameter about 0.12 to 0.27, which suggests that the shape parameter is very likely greater than zero.
5 FITTING A GENERALIZED PARETO DISTRIBUTION (GPD)

Figure 19: Profile log-likelihood plots for GPD 100-year return level (inches) and shape parameter ($\xi$) for Fort Collins precipitation data.

Visual inspection of the dashed vertical lines in figure 19 act as a guide to the accuracy of the displayed confidence limits; here the estimates shown appear reasonable as the dashed vertical lines (for both parameters) appear to intersect the profile likelihood in the same location as the (lower) horizontal line. Note that the confidence interval for the 100-year return level includes 4.63 inches, the amount recorded for the high precipitation event of July 1997.

5.3 Threshold Selection: Mean Residual Life Plot

Mean residual life plots, also referred to as mean excess plots in statistical literature, can also be plotted with the Extremes Toolkit. For more information on the mean residual life plot (and threshold selection) see section B.3. The general procedure for plotting a mean residual life plot using this toolkit is:

- **Plot** -> **Mean Residual Life Plot** -> *New window appears*

- **Select an object from** **Data Object** **field** -> **Variables appear in** **Select Variable field. Select one.**
Figure 20: Mean Residual Life Plot of Fort Collins precipitation data. Thresholds \( u \) vs Mean Excess precipitation (in inches).

- Choose other options \( \rightarrow \) OK
- Mean residual life plot appears.

**EXAMPLE: FORT COLLINS PRECIPITATION**

Figure 20 shows the mean residual life plot for the Fort Collins precipitation dataset. Interpretation of a mean residual life plot is not always simple in practice. The idea is to find the lowest threshold where the plot is nearly linear; taking into account the 95% confidence bounds. For the Fort Collins data, it is especially difficult to interpret, which may be due to the annual cycle (seasonality) that is being ignored here.

The mean residual life plot seen in figure 20 can be created from the toolkit by selecting **Mean Residual Life Plot** from the **Plot** menu of the main toolkit dialog box.

**Plot \( \rightarrow \) Mean Residual Life Plot**

A new dialog box should appear. Select **Fort** from the **Data Object** field. **month**, **day**, **year** and **Prec** should now appear in the **Select Variable** field—select **Prec** as this
is the dependent variable. Notice that you may also change the confidence level and the number of thresholds to plot. Here, just leave them as their defaults (95% and 100) and click on OK.

5.4 Threshold Selection: Fitting GPD Over a Range of Thresholds

The second method of trying to find a threshold involves fitting the GPD distributions for several different thresholds and checking for stability in the parameter estimates. The general procedure for fitting threshold ranges to a GPD is:

- **Plot -> Fit Threshold Ranges (GPD) -> New window appears**
- **Select a data object from Data Object field -> Variables appear in Select Variable field. Select one**
- **Enter upper and lower limits and number of thresholds in remaining fields -> OK**
- **If routine fails, try different ranges.**
- If successful, plot will appear.

Figures 21 show plots from having fit the GPD model for a range of 30 thresholds from 0.20 to 0.60. Figure 21 suggests that for the GPD model a threshold of 0.40 is probably appropriate.

**Example: Fort Collins precipitation**

To create the plot from figure 21 using the toolkit, select **Fit Threshold Ranges (GPD)** from the **Plot** menu on the main toolkit dialog.

**Plot -> Fit Threshold Ranges (GPD)**

Again, this brings up a new dialog box. First, select **Fort** from the **Data Object** field. Again, **month, day, year** and **Prec** should now appear in the **Select Variable** field. Select **Prec**. Enter 0.20 in the **Minimum Threshold** field, 0.60 in the **Maximum Threshold** field and 30 in the **Number of thresholds** field, then click **OK**. Note that you can try different values here as well, but keep in mind that the program will fail for certain threshold values. So, if you have trouble getting this function to work, keep trying different threshold ranges until it works.

6 Peaks Over Threshold (POT)/Point Process (PP) Approach

The GPD model from the previous chapter looks at exceedances over a threshold and those values are fit to a generalized Pareto distribution. A more theoretically appealing way
Figure 21: GPD fits for a range of 30 thresholds from 0.20 inches to 0.60 inches for the Fort Collins precipitation dataset.
to analyze extreme values is to use a point process characterization. This approach is consistent with a Poisson process for the occurrence of exceedances of a high threshold and the GPD for excesses over this threshold. Inferences made from such a characterization can be obtained using other appropriate models from above [3]. However, there are good reasons to consider this approach. Namely, it provides a nice interpretation of extremes that unifies all of the previously discussed models. For example, the parameters associated with the point process model can be converted to those of the GEV parameterization. In fact, the point process approach can be viewed as an indirect way of fitting the GEV distribution that makes use of more information about the upper tail of the distribution than does the block maxima approach. Additionally, the likelihood enables a more natural formulation of non-stationarity in threshold excesses than is obtained from the generalized Pareto model [3].

6.1 Fitting a Point Process Model

Figure 22 is not quite as easy to interpret as figure 21 for the GPD because of the fewer thresholds, but it seems that a threshold anywhere in the 0.30 to 0.40 range would be appropriate.

To create the plot in figure 22 select Fit Threshold Ranges (PP) from the Plot menu on the main toolkit dialog.

**Plot -> Fit Threshold Ranges (PP)**

This will bring up a new dialog. Follow the same instructions as for the GPD version (section 5.4), but using 0.31 instead of 0.20 and 10 instead of 30 thresholds and notice that there is a slider where you can select how many observations there are per year—the Number of obs per year field. Because there are approximately 365 observations per year in the Fort Collins dataset, leave this on the default of 365.

Once a threshold is selected, a point process model can be fit. Figure 23 shows diagnostic plots (probability and quantile plots) for such a fit.

To fit the Fort Collins precipitation data to a point process model do the following.

- **Analyze -> Point Process Model**
- **Select** Fort from the Data Object field.
- **Select** Prec from the Response field.
- **Check the** Plot diagnostics button and enter 0.4 in the Threshold value(s)/function field -> OK
Figure 22: Point process model fits for a range of 10 thresholds from 0.31 inches to 0.60 inches for the Fort Collins precipitation dataset.
Figure 23: Diagnostic plots for Fort Collins precipitation (inches) data fit to a point process model.
Maximum Likelihood estimates found for this fit are: $\hat{\mu} \approx 1.39$ inches (0.042), $\hat{\sigma} \approx 0.52$ inches (0.035) and $\hat{\xi} \approx 0.19$ (0.037) parameterized in terms of the GEV distribution for annual maxima, with negative log-likelihood of about -1247.55.

It is also possible to include a seasonal trend in the model; either within the model parameters or within the threshold for threshold exceedance models. To include a trend in the scale parameter, it is first necessary to perform a transformation to the months. To do this, click on Trigonometric Transformation from the Transform Data submenu of the File menu. That is,

File $\rightarrow$ Transform Data $\rightarrow$ Trigonometric Transformation

Next, select Fort from the Data Object field and then month from the Variables to Transform field. Finally, change the value of Period from the default of 365.25 to 12 and click OK. That is,

- Select Fort from the Data Object field.
- Select month from the Variables to Transform field.
- Change Period from 365.25 to 12.
- Click OK.

There should now be two added columns to the Fort Collins precipitation data matrix with the names month.sin12 and month.cos12. Now, we are ready to incorporate a seasonal cycle into the scale parameter of the GPD for the Fort Collins precipitation data. The process for this example is:

- From the Point Process Model dialog, select Fort from the Data Object field.
- Select Prec from the Response field.
- Select both month.sin12 and month.cos12 from the Scale parameter (sigma) field and check the exponential link for the scale parameter.
- Enter 0.4 in the Threshold field.
- Check plot diagnostics and click OK.

The model fit has the scale parameter as $\log(\hat{\sigma}(t)) \approx -0.661 + 0.005 \sin(2\pi t/12) - 0.03 \cos(2\pi t/12)$, with standard errors of 0.06786 inches, 0.0142 inches and 0.01751 inches.
Figure 24: Point process model fit diagnostics for Fort Collins precipitation (inches) data with a seasonal cycle in the scale parameter.

respectively. The shape parameter is estimated as about 0.18 (0.037) and the negative log-likelihood is about -1248.98.

A comparison of this model with the original fit without a seasonal cycle incorporated in the scale parameter gives a likelihood-ratio of about 2700, which is much greater than the associated $\chi^2_{2,0.05}$ quantile of 5.9915 and the associated p-value of approximately 0 is very small indicating improvement in the model fit by including a seasonal cycle. Probability and quantile plots shown in figure 24. The plots look good, particularly the probability plot, indicating that the model assumptions are reasonable. However, it may still be a good idea to investigate varying the threshold for seasonal effects, which will be illustrated in example 2 of the next section (6.2).

### 6.2 Nonconstant Thresholds

Sometimes data will have cyclical patterns that may be of interest when selecting a threshold. Some, such as engineers, may only be interested in the absolute maximum event, but others, such as climatologists, may be interested in modeling exceedances not only of the absolute maximum, but also in exceedances during a lower point in the cycle. For exam-
ple, figure 25 shows hourly precipitation at Denver, Colorado for the month of July from 1949-1990. Clearly, there is a diurnal cycle where precipitation is much less in the morning than in the afternoon. It may be necessary to use a smaller threshold in the morning than in the afternoon.

**Example 1: Denver Precipitation Data**

To illustrate how to use a time-varying threshold using the toolkit, first load the Denver precipitation data for the month of July. This dataset is available in the data directory of the toolkit and is called `Denversp.dat`. Follow the instructions from section 2.1 to load this dataset and save it (in R) as `Denver`. See Katz and Parlange [5] for more information on these data.

Although it is possible to use a time-varying threshold to fit a generalized Pareto model, care must be taken because of the dependence of the interpretation of the parameter $\sigma$ with the threshold. For this reason, it is preferable to use the point process model with parameterization in terms of GEV distribution for dealing with problems of this type.

Currently, the toolkit dialog windows only support time-varying thresholds using the point process approach. Nevertheless, the underlying software written by Stuart Coles (and ported into R by Alec Stephenson) do allow the user to apply time-varying thresholds to
the GPD model using the function \texttt{gpd.fit}. To see how to do this, click on the \textbf{Help} button on the GPD fit dialog box (help file will appear in R session window) or simply type \texttt{help(gpd.fit)} from your R session window.

It is currently not possible to include a time-varying threshold as part of the toolkit or the object, but it is possible to perform a point process model fit using a time-varying threshold. However, the function or vector determining the variable threshold must be created from within R. For the \texttt{Denver} precipitation data a threshold for the morning (say until noon) and a different threshold for the afternoon (say noon until midnight) should be sufficient for accounting for the diurnal cycle in the threshold. Of course, it is also possible to allow the parameters of the point process/GEV distribution to depend on time of day. Here, we look only at varying the threshold and not these parameters. So, from the R prompt, create a vector so that the threshold for the morning will be 0.10 inches and for the afternoon will be 0.40 inches using the following commands. Be careful not to mix up parenthesese ‘(’ with brackets ‘[’]. Brackets are used in R to subscript a vector or matrix. For example, if \texttt{foo} is a vector, then the command \texttt{foo[5]} will display the fifth element of foo.

\begin{verbatim}
> u.denver <- numeric(31247)
> morning.id <- Denver$data[,"Hour"] <= 12
> u.denver[morning.id] <- 0.1
> u.denver[!morning.id] <- 0.4
> u.denver
\end{verbatim}

Now, we are ready to fit a point process model using a time-varying threshold. Select \textbf{Point Process Model} from the \textbf{Analyze} menu of the main toolkit window.

\textbf{Analyze -> Point Process Model}

Select \texttt{Denver} from the \textbf{Data Object} field followed by \texttt{Prec} from the \textbf{Response} field. Now, check \texttt{vector} in the \textbf{Threshold} field and then type \texttt{u.denver} into the \textbf{Value(s)/function} input field and click \textbf{OK}.

The display for the main toolkit window should give summary information about the fit. In this case, the location parameter, \(\mu\), has an MLE of about .15 inches with associated standard error of roughly 0.049 inches. The scale parameter \(\sigma\) is about 0.32 inches (0.063 inches) and the shape parameter \(\xi\) is about -0.09 (0.12) and the negative log-likelihood is about 59.4. Figure 26 shows the probability and quantile plots for this fit.

Because we have two different thresholds for night and day, it might be a good idea
Figure 26: Probability and quantile plots for Denver precipitation data. Because the threshold varies with time, plots are shown for transformed variables and units of quantile plot do not directly correspond to the original data values. See appendix C.1 for more details.
to fit a model that reflects possible differences in model parameters for day and night. It is possible to incorporate such an effect into any of the three parameters, $\mu$, $\sigma$ or $\xi$, but generally it is better to look at only $\mu$ and $\sigma$. In this case, we will first need to create a new data field to identify day or night. For example, we can set up an indicator vector that is 0 if it corresponds to a data point from the daytime and 1 if it corresponds to a nighttime data point. This can be done from the R session as follows.

```r
> t.denver <- numeric(31247)
> t.denver[ Denver$data[,"Hour"] > 12 ] <- 1
> Denver$data <- cbind( Denver$data, t.denver)
```

Now simply fit a point process model to these data as above, but using `t.denver` in the Location parameter ($\mu$) field. Figure 27 shows the results from fitting the above model, but with a binary variable for time indicating morning or night. The fit is clearly improved. Parameter estimates are now:

\[
\hat{\mu} \approx -0.0537 + 0.28033 \cdot t \text{ inches (0.1092 inches and 0.1115 inches respectively)}
\]
\[
\hat{\sigma} \approx 0.276 \text{ inches (0.112 inches)}
\]
\[
\hat{\xi} \approx -0.06 \text{ (0.1194)}
\]

where $t$ is either 0 or 1 for day or night respectively.

**Example 2: Fort Collins Precipitation Data**

As in example 1 of this section, it will be necessary to create a vector from the R prompt that will be used as the nonconstant threshold. There are many ways to decide upon a threshold for these data. One could have a single threshold for each season, similar to example 1, or one might use a trigonometric function to vary the threshold for each season. The latter will be employed here.

```r
> mths <- Fort$data[,"month"]
> u.fortcollins <- 0.475+5*(-0.03*cos(2*pi*mths/12))
```

Figure 28 shows a plot of the Fort Collins precipitation data with both the previously used constant threshold of 0.4 inches and the above cyclical threshold. The following commands created the plot in figure 28.

```r
> prec <- Fort$data[,"Prec"]
> plot( mths, Fort$data[,"Prec"], xlab="Month", ylab="precipitation (inches)", xaxt="n")
```
Figure 27: Point process fit diagnostics for Denver precipitation data for the month of July (1949-1990) using a time-varying threshold to reflect the diurnal cycle and a binary time trend variable indicating morning or night in the location parameter.
Figure 28: Fort Collins precipitation data with constant threshold of 0.4 inches (solid black line) and nonconstant (cyclic) threshold (solid blue line).

```R
> abline( h=0.4)
> lines( mths[order(mths)], u.fortcollins[order(mths)], col="blue")
```

Fitting a point process model using `u.fortcollins` to fit a nonconstant (seasonal) threshold gives parameter values of $\hat{\mu} \approx 1.40$ inches (0.043 inches), $\hat{\sigma} \approx 0.53$ inches (0.034 inches) and $\hat{\xi} \approx 0.16$ (0.040) and associated negative log-likelihood of about -619.64. Note that the shape parameter estimate is close to that found when including a seasonal cycle in the scale parameter from section 6.1. Inspection of the diagnostic plots (figure 29) suggests that the model assumptions seem reasonable. For different cycles in the threshold with higher peaks in the summer months resulted in rather poor fits suggesting that too much data is lost and so the lower thresholds are necessary.

7 Extremes of Dependent and/or Nonstationary Sequences

Much of the theory applied thus far assumes independence of the data, which may often not be the case when looking at extreme values because of the tendency for extreme conditions to persist over several observations. The most natural generalization of a sequence of
Figure 29: Probability and quantile plots for fitting a point process model to the Fort Collins precipitation (inches) data with a seasonal cycle incorporated into the threshold.
independent random variables is to a stationary series, which is quite realistic for many physical processes. Here the variables may be mutually dependent, but the stochastic properties are homogeneous over time (see Coles [3] Chapter 5). Extreme value theory still holds, without any modification, for a wide class of stationary processes; for example, for a Gaussian autoregressive moving average process. With modification the theory can be extended to an even broader class of stationary processes.

7.1 Declustering

Clustering of extremes can introduce dependence in the data that subsequently invalidates the log-likelihood associated with the GPD for independent data. The most widely adopted method for dealing with this problem is declustering, which filters the dependent observations to obtain a set of threshold excesses that are approximately independent. Specifically, one uses some empirical rule to define clusters of exceedances, identifies the maximum within each cluster and fits the GPD to the cluster maxima assuming independence among cluster maxima.

One simple way to determine clusters is to first specify a threshold and define clusters to be wherever there are consecutive exceedances of this threshold. Once a certain number of observation, say \( r \), fall below the threshold, the cluster is terminated. There are issues regarding how large both the threshold and \( r \) should be and improper choices can lead to either bias or large variance. Therefore, the sensitivity of results should be checked for different choices of threshold and \( r \). See Coles [3] chapter 5 for more on this method and chapter 9 for some alternatives to declustering.

Currently, this toolkit provides for declustering the data using the above method, but generally declustering is a more involved process that should be carefully done by the user and is not generally supported by the toolkit itself. The general procedure for declustering data with the toolkit is as follows.

- **File \(\rightarrow\) Decluster**
  - **Select data from the Data Object field**
  - **Select the variable you wish to decluster from the Variable to Decluster field**
  - **Select the variable you wish to decluster by (optional) from the Decluster by field**
  - **Enter desired threshold (or vector of thresholds) in the Threshold field**
  - **Enter a number for \( r \) \(\rightarrow\) OK.**

**Example: Phoenix Minimum Temperature**
The Phoenix minimum temperature data included with this toolkit represents a time series of minimum (maximum also available with the included dataset) temperatures (in degrees Fahrenheit) for July through August 1948 to 1990 from the U.S. National Weather Service Forecast Office at the Phoenix Sky Harbor Airport. For more information on these data, please see Tarleton and Katz [11] or Balling et al. [1]. Temperature is a good example of data which may have dependency issues because of the tendency of hot (or cold) days to follow other hot (or cold) days. For this example, please read in the Tphap.R or Tphap.dat dataset and save it (in R) as Tphap.

It is of interest with this dataset to look at the minimum temperatures. To do this, we must first transform the data by taking the negative of the MinT variable so that our extreme value distribution theory for maximums apply to the minimum. That is, 
\[-\max(-X_1, \ldots, -X_n) = \min(X_1, \ldots, X_n),\]
so the same theory applies to minima as applies to maxima under this transformation. This can be performed quite easily with the toolkit. Simply choose Negative from the Transform Data submenu of the File menu on the main toolkit window. That is,

File -> Transform Data -> Negative

A new dialog will appear. Select Tphap from the Data Object field and then MinT from the Variables to Transform field and then click OK.

After reading these data into R and taking the negative of the minimum temperature do the following.

- File -> Decluster
  - Select Tphap from the Data Object field
  - Select MinT.neg from the Variable to Decluster field
  - Select Year from the Decluster by field
  - Enter \(-70\) in the Threshold field
  - Leave the default of 1 in the r field -> OK.
  - It is a good idea to try several values of r to try to find the “best” set of clusters.

It is also possible to plot the data with vertical lines where the clusters are by clicking on the Plot data checkbox. However, often, as in this case, the amount of data and relatively large number of clusters produced create a messy, illegible plot. Therefore, leave this box unchecked for this example. A message will be displayed on the main toolkit window.
that 169 clusters were found and that the declustered data was assigned to \texttt{MinT.neg.u-70r1dcbyYear}. This column has been added to the original data matrix using this name (where \texttt{u-70} corresponds to the threshold of -70 and \texttt{r1} corresponds to \( r \) being 1). Note that because this process reduces the number of data points, values below the threshold have been filled in so that this declustered data would have the correct dimensions in order to be added to the original data matrix. These values will not affect any POT analyses. The optional use of the \texttt{Decluster by} feature ensures that, in this case, values from one year will not be clustered with values from another year; an important feature with these data as there are several years worth of data, but for only two months of each year.

The next step would be to perform a GPD fit on this newly declustered data. So, select \texttt{Generalized Pareto Model} from the \texttt{Analyze} menu of the main toolkit window.

\texttt{Analyze -> Generalized Pareto Model}

From the dialog window that appears, select \texttt{Tphap} again from the \texttt{Data Object} field and then \texttt{MinT.neg.u-70r1dcbyYear} from the \texttt{Response} field. Enter -70 in the threshold field and click on \texttt{OK}. One thing to be careful about, in general, is that the number of points per year (\texttt{npy}) may be different once the data has been declustered. This will not affect parameter estimates for the GPD, but can affect subsequent calculations such as return levels, which are usually expressed on an annual scale. Figure 31 shows the diagnostic plots for this fit, which indicate that the model assumptions are fairly reasonable, but not very good in the extreme upper (really lower) tail. However, a look at the diagnostic plots in figure 30 for the same fit to the original data suggests that the declustering (with \( r = 1 \)) did not impact the model assumptions greatly. Figure 32 shows the results from declustering with a 90 degree threshold and inclusion of a seasonal term and the upper (lower) tail looks better. The same analyses for \( r = 2 \) did not improve the diagnostic plots.

7.2 Incorporating nonstationarity into a point process approach

Often with real data, such as temperature data, there is some day-to-day dependence in the data. That is, it is likely that one cool day will follow another cool day. Dependence of extremes, however, is not necessarily as high as the data values and so it may not be a problem. Nevertheless, there are methods available to handle such dependence in extreme value analysis. One such method is to decluster the data as was done in section 7.1. Another idea is to incorporate a trend in the location (and/or scale) parameter(s) of the extreme value distribution (GEV, GP, point process, etc...).

\textbf{Example: Phoenix Minimum Temperature}

Here, we again look at the \texttt{Tphap} dataset used in section 7.1. The minimum temper-
Figure 30: Diagnostic plots for GP fit (\( u = 70 \)) to summer minimum temperatures (degrees Fahrenheit) for Phoenix, AZ
Figure 31: Diagnostic plots for GP fit ($u = 70$) to declustered ($r = 1$, $u = 70$) summer minimum temperatures (degrees Fahrenheit) for Phoenix, AZ
Figure 32: Diagnostic plots for GP fit to declustered \( (r = 1, u = 90) \) summer minimum temperatures (degrees Fahrenheit) for Phoenix, AZ
7 EXTREMES OF DEPENDENT AND/OR NONSTATIONARY SEQUENCES

Figure 33: Scatter plot of minimum temperature (degrees Fahrenheit) for Phoenix, AZ

ature data for Phoenix, Arizona, shown in figure 33, is a good example of data that clearly has an upward trend over time and possibly a varying standard deviation from one year to the next. The blue line in figure 33 is the least squares fit \( \mu_t = \beta_0 + \beta_1 \cdot (t - 1948) \), which has a significant positive slope.

Fit a point process model to the \textbf{MinT.neg} variable (the negative of the \textbf{MinT} variable) in the \textbf{Response} field of the \textbf{Fit GEV} dialog window. The fitted model gives \( \hat{\mu} \approx -63.78 \) degrees (0.254 degrees), but recall that this is for the negative of the data and so the location parameter is actually about 63.78 degrees. The other parameter values do not need to be transformed. So, \( \hat{\sigma} \approx 1.256 \) degrees (0.066 degrees), \( \hat{\xi} \approx -0.31 \) (0.014) and the negative log-likelihood is about -2275.28 and the probability and qq-plots shown in figure 34 suggest that the model assumptions are reasonable, although there appears to be some curvature to them; in particular, the extreme upper and lower tails suggest a very poor fit. As this is the first point process model fit performed on these data, the fitted object is stored as “pp.fit1” in the Tphap object.

To incorporate a trend in the location parameter of a point process for the Phoenix minimum temperature data, perform the following steps.

- **Analyze** -> **Point Process Model** -> *New window appears*
Figure 34: Point process fit diagnostics for Phoenix daily minimum temperature using a threshold of 70 degrees Fahrenheit (no trends).
Figure 35: Point process fit diagnostics for Phoenix daily minimum temperature using a threshold of 70 degrees Fahrenheit and a linear trend in the location parameter.

- Select Tphap from Data Object field. → Values added to other fields
- Select MinT.neg from Response field.
- Select Year from Location parameter (Mu): field.
- Enter −70 in the Threshold value(s)/function field → OK

The likelihood-ratio statistic comparing this fit to the fit without a trend shows that incorporation of the linear trend in year is not significant, but the same analysis with a threshold of 80 degrees Fahrenheit does show significance ($\alpha = 0.05$) with likelihood-ratio of about 7.55 (compared to $\chi^2_{10.95} = 3.84$) and associated p-value of about 0.006. Figure 35, on the other hand, indicates that there may be a problem with the underlying assumptions for this model, at least in the extreme upper tail (really the lower tail). A similar fit with terms for both month and year shows some improvement over this model, but not enough to argue for the added complexity and the quantile plot shows the same problem in the extreme upper tail. Incorporation of a trend in the scale parameter did not show any improvement over the model without any trends. It might be that the data needs to be declustered.
8 Details

8.1 Trouble Shooting

If the main toolkit dialog does not appear on startup or it appears, but many of the functions do not work, then check the following possible causes.

- It may be that R does not know where the extRemes library is located. R assumes that all libraries (packages) are in the same place. Often, however, a user may wish to have a package somewhere else; for example, a unix user who does not have root privileges cannot install packages in the location where R checks for them. In this case, it is necessary to tell R where the package is located. If, for example, the package is installed in the directory, /home/[user]/src/library, then the toolkit must be loaded into R using the following command.

  ```r
  > library( extRemes, lib.loc="/home/[user]/library")
  ```

- Another possible cause for the dialog to not appear is that this toolkit depends on the R package tcltk, which interfaces with the Tcl/Tk programming language. The Tcl/Tk programming language must also be installed on your system and R must know where to find it in order for the toolkit to work. Please see section 8.3 for more information on obtaining, installing and pointing R to Tcl/Tk.

- If you receive an error message that says,

  Error in eval(expr, envir, enclos) : Object "gev.diag" not found

then the package ismev is not loaded. In order to load this package from the R prompt, simply type:

  ```r
  > library( ismev)
  ```

Or, if it is installed in a library where R does not know to look, such as /home/[user]/src/library, then type:

  ```r
  > library( ismev, lib.loc="/home/[user]/src/library")
  ```

Of course, this package will not load if it has not been installed. If it is not installed, then it can be installed from the R prompt by the command:

  ```r
  > install.packages( "ismev")
  ```
If this does not work, then it is likely (if you are using unix or linux–see the next bullet point for Windows) that you do not have permission to write to the file where R wants to install packages. If this is the case, then it is possible to tell R to put it someplace else. For example, to install ismev in the directory /home/[user]/src/library, use the command:

```>
install.packages( "isme\v\n\nOnce ismev is installed on your system, it needs to be loaded into R (see above).

• In Windows, it probably says that ismev is not a package on CRAN. In this case, you may have to download the package source for ismev, install it in unix/Linux and ftp it over to Windows (and into your R library folder).

8.2 Is it Really Necessary to Give a Path to the library Command Every Time?

On the unix and linux platforms, if you do not have root privileges, then you will have had to type:

```>
library( extRemes, lib.loc="[path to extRemes library]"
```every time you want to load the extRemes library. Similarly for any other R package, like ismev, that you install onto your own space. It is possible to set up a file called .Rprofile that will be called by R every time you start an R session. Inside this file, you can tell it where to look for packages that you install. To make this file available to any R session it is necessary to put .Rprofile in your home directory. Assuming that your packages are in /home/[user]/src/library, the .Rprofile file should look something like:

``` .First <- function() {
  cat("Hello! You can put any R function that you want run upon start-up in here.")
  # Ok, this next command points R to where your packages exist.
  # Note that R will still look in the default path as well.
  .libPaths("/home/[user]/src/library")

  # Now you will no longer need to use the lib.loc argument to library
  # when calling a package located in /home/[user]/src/library.
} ```
.Last <- function() {
  cat("Good-bye! You can put any R function that you want run while exiting R here.")
}

8.3 Software Requirements

The following directions were current at the time this tutorial was first written and apply to R < 1.7.0, so please consult the Windows FAQ on the R project web site for more up-to-date directions.

First, Tcl/Tk libraries and R must be installed on the system. R comes with a large amount of documentation detailing installation. To install R, go to:

http://cran.r-project.org/index.html

To obtain the necessary Tcl/Tk software, go to:
http://dev.scriptics.com/

**Important!** The Tcl/Tk interface package, tcltk, for R versions < 1.7.0 only work with Tcl version 8.3.x and for R version 1.7.0, it only works with the newer Tcl/Tk version 8.4.x. For Windows users, R version 1.7.0 now installs Tcl/Tk for you by default. If you are on Windows and using R version 1.7.0, please see the Windows FAQ on the R project site (http://www.R-project.org) for more information. If you do not know which version of R you have, type (from the R prompt):

```
> R.version.string
```

For instructions on installing the Tcl/Tk software go to:


**Notes:**

- To install Tcl/Tk on unix, you may want to ask your systems administrator to do it for you as it is a rather onerous affair.

- In unix, you may have to set an environment variable to let R know where to find Tcl/Tk. Something like:
  ```
  setenv TCL_LIBRARY /opt/local/tcl8.3.2/lib/tcl8.3
  setenv TK_LIBRARY /opt/local/tk8.3.2/lib/tk8.3
  ```
Again, check with your system administrator about specifics to your system. For instance, the path to tcl8.3 will probably be different from the one given above. Ask your systems administrator where it is, or try the following unix/linux command.

```bash
> find /[base directory] -name init.tcl -print
```

Note that [base directory] should be replaced with the directory where you suspect tcl might be. Something like /opt (above example), /usr or something of the kind.

Once you have set the correct TCL_LIBRARY and TK_LIBRARY paths it is recommended that you enter these commands in your .login or .cshrc or other appropriate file so that these variables are set automatically in the future.

- In Windows, if you are using an R version < 1.7.0 you will also need to tell R where to find Tcl/Tk. It may behoove you to simply upgrade to version 1.7.0 (or greater). Otherwise, you will need to set an environment variable and possibly a path. This can be done from within your R session with the following type of commands (see the R-CRAN Windows FAQ for more information):

```r
> Sys.putenv("TCL_LIBRARY"="C:/Program Files/Tcl/lib/tcl8.3")
> Sys.putenv(PATH=paste(Sys.getenv("PATH"), "C:/Tcl/bin", sep = ";"))
```

Note that if you set the environment variable from within R, it will not remember this for the next session. Better to upgrade to R version 1.7.0 (or greater) and not have to worry about it ever again.

### 8.4 The Underlying Functions

The underlying functions that actually perform the extreme value analyses were written by Stuart Coles for S-Plus and were ported into R by Alec Stephenson. For information on these functions please see Coles [3] and more specifically the accompaniment to this book [2]. For information on the R port see the web page: http://www.maths.lancs.ac.uk/stephena/software.html.

The primary difference between the original S-PLUS version and the R port is that the R port uses the `optim` function for finding MLEs instead of the S-PLUS `nlm` function. The following notes are nearly verbatim from Alec Stephenson’s notes on the differences.

- As mentioned above, the R port uses the general purpose optimization function `optim`. If R cannot find this function make sure you have the latest version of R.

- Both R and S may give warning messages of the form ‘NaNs produced in: log(x)’. This is a result of evaluating `log` at a negative number and may occur when the likelihood
is evaluated outside of the valid parameter space. These warnings can generally be ignored.

- In the S version, the $conv element of the returned fit list is either true or false (T or F). When true, a local minimum has theoretically been found. In the R port, the $conv element is the return code provided by the optim function. See the help file for optim for the details. A local minimum has theoretically been found when this is zero.

- The optim function in R allows the user to select which optimization method to use. These may be selected from the extreme toolkit dialogs as well. The default method is Nelder-Mead. Another useful method is BFGS. Generally, if one method seems to fail try the other.

8.5 Miscellaneous

Whenever a GEV, GPD or PP model is fit to a data object, the entire fitted object is stored within the original data object. Because this toolkit uses Stuart Coles’ routines for the fits, the original data is duplicated in the fitted object. For larger datasets this can quickly increase the size of the .RData file and suck up memory. If you are using a relatively large dataset and are performing many fits on it, then it would be a good idea to remove fits that you no longer need. For example, if you want to remove the first GPD fit performed on the ev.data object foo, do the following from the R prompt:

\[\text{> foo$models$gpd.fit1 <- NULL}\]

A Appendix A: Generalized Extreme Value distribution

Let $X_1, \ldots, X_n$ be a sequence of independent identically distributed (i.i.d.) random variables with distribution function, $F$. Then let $M_n = \max\{X_1, \ldots, X_n\}$. For known $F$, the distribution of $M_n$ can be derived exactly for all values of $n$ because $\Pr\{M_n \leq u\} = \Pr\{X_1 \leq u; \forall i = 1, \ldots, n\}$, which by the fact that the $X_i$ are independent is equivalent to $\Pr\{X_1 \leq u\} \cdot \Pr\{X_2 \leq u\} \cdots \Pr\{X_n \leq u\}$ and because the $X_i$ are identically distributed this is equivalent to $(\Pr\{X_1 \leq u\})^n$. Thus, $\Pr\{M_n \leq u\} = (F(u))^n$. Note, however, that the independence assumption, which virtually never occurs for weather and climate variables, can be relaxed (see appendix C).

The problem with the above exact distribution is that $F$ is not generally known in practice and subsequently must be estimated. However, small discrepancies between $F$ and its estimate, say $\hat{F}$, can lead to large discrepancies between $F^n$ and $\hat{F}^n$. A widely accepted
alternative is to accept $F$ as unknown and look for approximate models for $F^n$ that can be estimated on the basis of the extreme data alone.

Of course, as $n$ increases $F^n$ quickly approaches zero due to the fact that $F$ is a distribution function and therefore yields values only between zero and one. That is, $F^n \to 0$ as $n \to \infty$. Thus, in order to achieve a nondegenerate distribution function it is necessary to find sequences of constants \( \{a_n > 0\} \) and \( \{b_n\} \) such that

\[
F^n \left( \frac{M_n - b_n}{a_n} \right) \to G(z)
\]

where $G(z)$ does not depend on $n$.

For example, suppose $F(x) = 1 - e^{-x}$ (exponential distribution). Then, $Pr\{ \frac{M_n - b_n}{a_n} \leq u \} = Pr\{ M_n \leq b_n + a_n u \} = F^n(b_n + a_n u)$. Letting $a_n = 1$ and $b_n = \log n$ yields the following.

\[
F^n(\log n + u) = [1 - \exp\{-\log n + u\}]^n = [1 - \frac{1}{n} e^{-u}]^n \to \exp(-e^{-u}) \text{ as } n \to \infty,
\]

which is a distribution known as the Gumbel distribution.

In fact, the Gumbel is one of three possible types of distributions to which $F^n$ can converge. The three types are:

I. Gumbel

\[
G(z) = \exp\{-\exp[-(\frac{z-\mu}{\sigma})]\}, \quad -\infty < z < \infty \quad \text{(Gumbel)}
\]

II. Fréchet

\[
G(z) = \begin{cases} 
0 & z \leq \mu, \\
\exp\{-\left(\frac{z-\mu}{\sigma}\right)^{-1/\xi}\} & z > \mu.
\end{cases}
\]

III. Weibull

\[
G(z) = \begin{cases} 
\exp\{-\left(\frac{z-\mu}{\sigma}\right)^{1/\xi}\} & z < \mu, \\
1 & z \geq \mu.
\end{cases}
\]

for parameters $\sigma > 0$, $\mu$ and $\xi > 0$.

The above three families of distributions can be combined into one family of distributions known as the generalized extreme value (GEV) family. Namely,

\[
G(z) = \exp\{-[1 + \xi(z-\mu)]^{-1/\xi}\}
\]
where \( \{ z : 1 + \xi(z - \mu)/\sigma > 0 \} \), \(-\infty < \mu, \xi < \infty\) and \( \sigma > 0 \). Please refer to Coles [3] for more information on the GEV family.

B Appendix B: Threshold Exceedances

Modeling only block maxima is wasteful if other other data on extremes are available [3]. Let \( X_1, X_2, \ldots \) be a sequence of independent and identically distributed (i.i.d.) random variables with distribution function \( F \). Now, for some threshold, \( u \), it follows that

\[
\Pr\{X > u + y | X > u\} = \frac{1 - F(u + y)}{1 - F(u)}, \ y > 0.
\]

If \( F \) is known, then so is the above probability. However, this is often not the case in practical applications and so approximations that are acceptable for high values of the threshold are sought–similar to using the GEV distributions for block maxima.

The generalized Pareto distribution (GPD) arises in the peaks over threshold (POT)/point process (PP) approach.

B.1 Generalized Pareto Distribution

Again, letting \( X_1, X_2, \ldots \) be a sequence of i.i.d. random variables with common distribution function, \( F \), and let \( M_n = \max\{X_1, \ldots, X_n\} \). Now, assuming \( F \) satisfies certain conditions (see Coles [3] for more information) then we have that \( \Pr\{M_n \leq z\} \approx G(z) \), where

\[
G(z) = \exp\{-[1 + \xi(\frac{z - \mu}{\sigma})]^{-1/\xi}\}
\]

for some \( \mu, \sigma > 0 \) and \( \xi \). Then for a large enough threshold, \( u \), the distribution function of \((X - u)\), conditional on \( X > u \), is approximately

\[
H(y) = 1 - (1 + \xi \frac{y}{\tilde{\sigma}})^{-1/\xi}
\]

defined on \( \{ y : y > 0 \) and \((1 + \xi \frac{y}{\tilde{\sigma}}) > 0\} \), where \( \tilde{\sigma} = \sigma + \xi(u - \mu) \). \( H(y) \) is referred to as the generalized Pareto distribution (GPD). Again, see Coles [3] for more information on the generalized Pareto distribution.

B.2 Peaks Over Threshold (POT)/Point Process (PP) Approach

The point process approach to the threshold excesses problem provides an interpretation of extreme value behavior that unifies all of the other models. Additionally, it leads directly to a likelihood that enables a more natural formulation of non-stationarity in threshold
excesses than can be obtained from the generalized Pareto model [3]. That is, in this approach, the times at which high threshold exceedances occur and the excess values over the threshold are combined into one process based on a two-dimensional plot of exceedance times and exceedance values. The asymptotic theory of threshold exceedances shows that under suitable normalization, this process behaves like a nonhomogeneous Poisson process [9]. For more information on this approach, see Coles [3], Smith [10] and Smith [9].

B.3 Selecting a Threshold

Selecting an appropriate threshold is a critical problem with the POT methods. Too low a threshold is likely to violate the asymptotic basis of the model; leading to bias; and too high a threshold will generate too few excesses; leading to high variance. The idea is to pick as low a threshold as possible subject to the limit model providing a reasonable approximation. Two methods are available for this: the first method is an exploratory technique carried out prior to model estimation and the second method is an assessment of the stability of parameter estimates based on the fitting of models across a range of different thresholds [3].

Suppose the raw data consist of a sequence of i.i.d. measurements $x_1, \ldots, x_n$ and let $x_{(1)}, \ldots, x_{(k)}$ represent the subset of data points that exceed a particular threshold, $u$. Define threshold excesses by $y_j = x_{(j)} - u$ for $j = 1, \ldots, k$. The first method requires plotting the points

\[
\{(u, \frac{1}{n_u} \sum_{i=1}^{n_u} (x_{(i)} - u)) : u < x_{\text{max}}\}.
\]

The resulting plot is called the mean residual life plot in engineering and the mean excess function in the extremes community.

C Appendix C: Dependence Issues

The asymptotic distribution approximation of maximums and exceedances over a threshold assumes that data are independent and identically distributed (iid), but this is often not the case with real data. Nevertheless, the results can still be used. There are a few different methods for dealing with this problem. One is to decluster the data so that cluster maxima are independent. Another is to incorporate the dependence into a trend. For some data, the results are still valid even without declustering or incorporating a trend.

C.1 Probability and Quantile Plots for Non-stationary Sequences

For non-stationary time series, it is possible to incorporate a trend (or covariate) into the parameters of the GEV, GPD or Point Process models. Subsequently, each time point
(or covariate value) has a different distribution associated with it. In order to plot model diagnostics, therefore, it is necessary to transform the data in such a way that each point has the same distribution. This can be accomplished in the following ways.

In the case of the GEV distribution, if we have $Z_t \sim \text{GEV}(\hat{\mu}(t), \hat{\sigma}(t), \hat{\xi}(t))$ then the standardized variables,

$$\tilde{Z}_t = \frac{1}{\hat{\xi}(t)} \log\{1 + \hat{\xi}(t)\left(\frac{Z_t - \hat{\mu}(t)}{\hat{\sigma}(t)}\right)\},$$

(1)

each have the standard Gumbel distribution with probability distribution function

$$P\{\tilde{Z}_t \leq z\} = \exp\{-e^{-z}\}, z \in \mathbb{R}.$$  

(2)

Probability and quantile plots can be made with (2) as the reference distribution [3]. Let $\tilde{z}_{1:n}, \ldots, \tilde{z}_{n:n}$ denote the ordered values of the transformed variables from (1), the probability plot consists of the pairs

$$\{(\frac{i}{n+1}, \exp(-\exp(-\tilde{z}_{i:n}))); i = 1, \ldots, n\}$$

and the quantile plot consists of

$$\{(-\log(1 - \frac{i}{n+1})), \tilde{z}_{i:n}); i = 1, \ldots, n\}$$

For the GPD, if we have $Y_t \sim \text{GP}(\hat{\sigma}(t), \hat{\xi}(t))$, where $t = 1, \ldots, k$ ($k$ threshold excesses) then the transformation

$$\tilde{Y}_t = \frac{1}{\hat{\xi}(t)} \log\{1 + \hat{\xi}(t)\left(\frac{Y_t - u(t)}{\hat{\sigma}(t)}\right)\}$$

(3)

follows the same standard exponential distribution for each of the $k$ excesses over the threshold, $u(t)$ ($u(t)$ may vary with time) [3].

In this case, the probability plot is formed by the pairs of points

$$\{(\frac{i}{k+1}, 1 - \exp(-\tilde{y}_{i:k})); i = 1, \ldots, k\}$$

and the quantile plot is formed by

$$\{(-\log(1 - \frac{i}{k+1})), \tilde{y}_{i:k}); i = 1, \ldots, k\}.$$
Finally, for the point process model, the transformation

$$\tilde{Y}_t = 1 + \xi(t)(\frac{Y_t - u(t)}{\hat{\sigma}(t) + \xi(t)(u - \hat{\mu}(t))})^{-1/\xi(t)}$$  \hfill (4)

is employed and the probability plot consists of the pairs

$$\{(\frac{i}{k+1}, \tilde{y}_{ik}); i = 1, \ldots, k\}$$

and the quantile plot consists of the pairs

$$\{(-\log(1 - \frac{i}{k+1}), -\log(1 - \tilde{y}_{ik})); i = 1, \ldots, k\}$$

References


