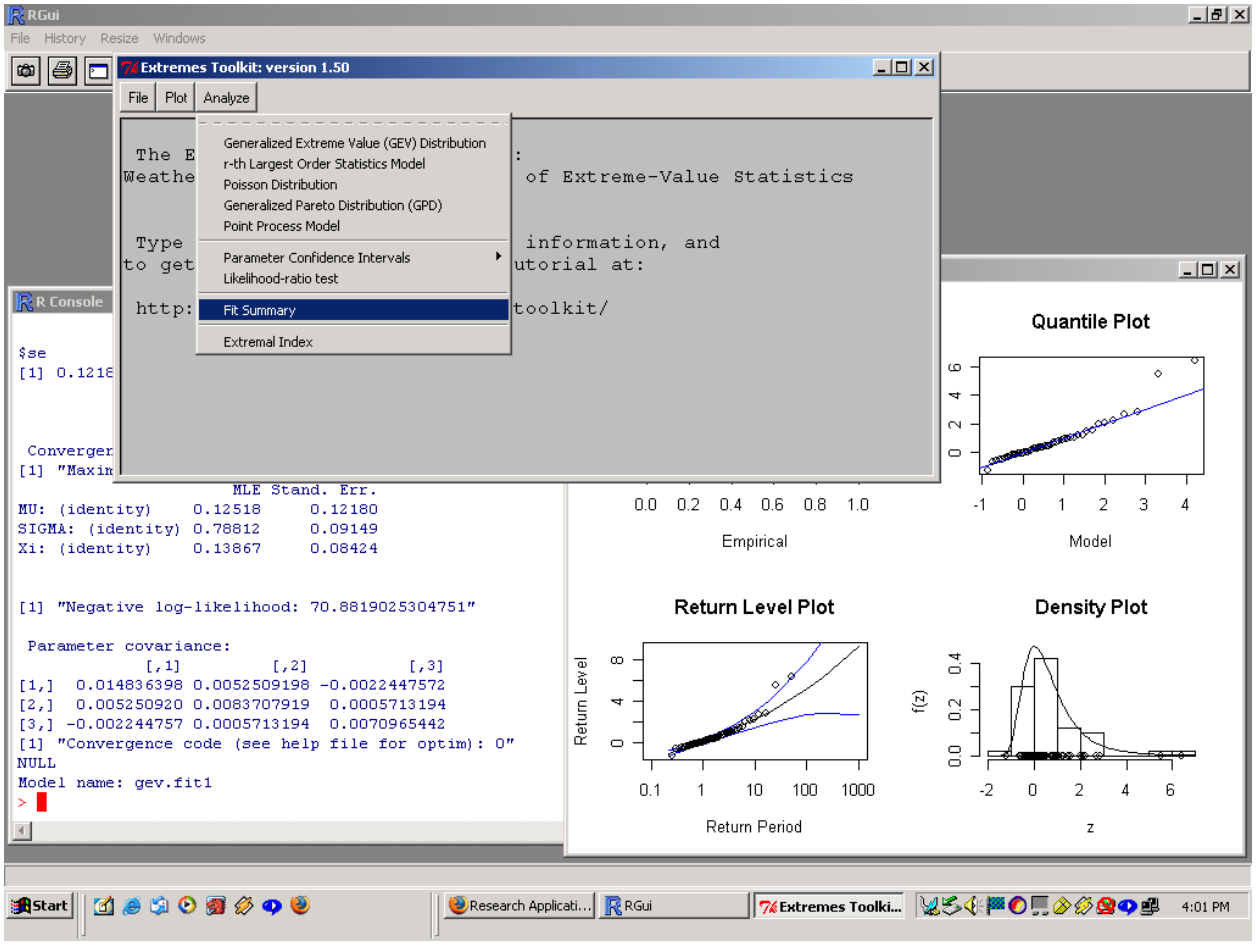
Extremes Toolkit (extRemes):
Weather and Climate Applications of Extreme Value Statistics[1]

Eric Gilleland[2] and Richard W. Katz[3]



[1]This toolkit is funded by the National Science Foundation (NSF) through the National Center for Atmospheric Research (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. This tutorial is for version 1.50 (July, 2005).

[2]*Corresponding author address*: NCAR, Research Applications Laboratory (RAL), P.O. Box 3000, Boulder, CO 80307-3000, U.S.A.

[3]NCAR, Institute for the Study of Society and Environment (ISSE)

Summary: The Extremes Toolkit (`extRemes`) 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 [14] was chosen in conjunction with a Tcl/Tk interface. R is a GNU-license product available at www.r-project.org. Tcl/Tk is a popular GUI development platform also freely available for Linux, Unix and the PC (see section 8.0.22 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

GEV	Generalized Extreme Value
GPD	Generalized Pareto Distribution
MLE	Maximum Likelihood Estimator
POT	Peaks Over Threshold
PP	Point Process

Contents

Chapter 1

Preliminaries

Once `extRemes` has been installed (see `http://www.isse.ucar.edu/extremevalues/evtk.html` for installation instructions), the toolkit must be loaded into R (each time a new R session is invoked). Instructions for loading `extRemes` into your R session are given in section 1.1. Once the toolkit is loaded, then data to be analyzed must be read into R, or simulated, as an "ev.data" object (a dataset readable by `extRemes`). Instructions for reading various types of data into R are given in section 1.2.1, and for simulating data from the GEV distribution or GPD in sections 1.2.2 and 1.2.3. Finally, section 1.2.4 discusses creating an "ev.data" object from within the R session. For a quick start to test the toolkit, follow the instructions from section 1.2.2.

1.1 Starting the Extremes Toolkit

It is assumed here that `extRemes` is already installed, and it merely needs to be loaded. If `extRemes` has not yet been installed, please refer to the `extRemes` web page at
`http://www.esig.ucar.edu/extremevalues/evtk.html`
for installation instructions.

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

```
> library( extRemes)
```

The main `extRemes` dialog should now appear. If it does not appear, please see section 8.0.20 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()
```

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

Table 1.1: *U.S. total economic damage (in billion $) due to floods (USDMG) by hydrologic year from 1932-1997. Also gives damage per capita (DMGPC) and damage per unit wealth (LOSSPW). See Pielke and Downton [12] for more information.*

1.2 Data

The Extremes Toolkit allows for both reading in existing datasets (i.e., opening a file), and for the simulation of values from the generalized extreme value (GEV) and generalized Pareto (GP) distributions.

1.2.1 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 1.1 represents a typical *common* dataset; in this case data representing U.S. flood damage. See Pielke and Downton [12] or Katz *et al.* [9] for more information on these data.

An *R source* dataset is a dataset that has been *dumped* from R. These typically have a .R or .r extension. That is, it is written in R source code from within R itself. Normally, these are not the types of files that a user would need to load. However, `extRemes` and

many other R packages include these types of datasets for examples. It is easy to decipher if a dataset is an *R source* file or not. For example, the same dataset in Table 1.1 would look like the following.

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

Apart from the *Flood* data, all other datasets included with the toolkit are *R source* datasets.

Data loaded by `extRemes` are assigned to 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 are 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 are 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 `George$finance` (or `George[["finance"]]`) at the R prompt.

Example 1: Loading a Common Dataset

Here we will load the common dataset, **Flood.dat**, which will be located in the `extRemes` **data** directory. From the main toolkit dialog, select **File** > **Read Data**. A new window appears for file browsing. Go to the `extRemes` data directory and select the file **Flood.dat**; another new window will appear that allows you to glance at the dataset (by row) and has some additional options. That is,

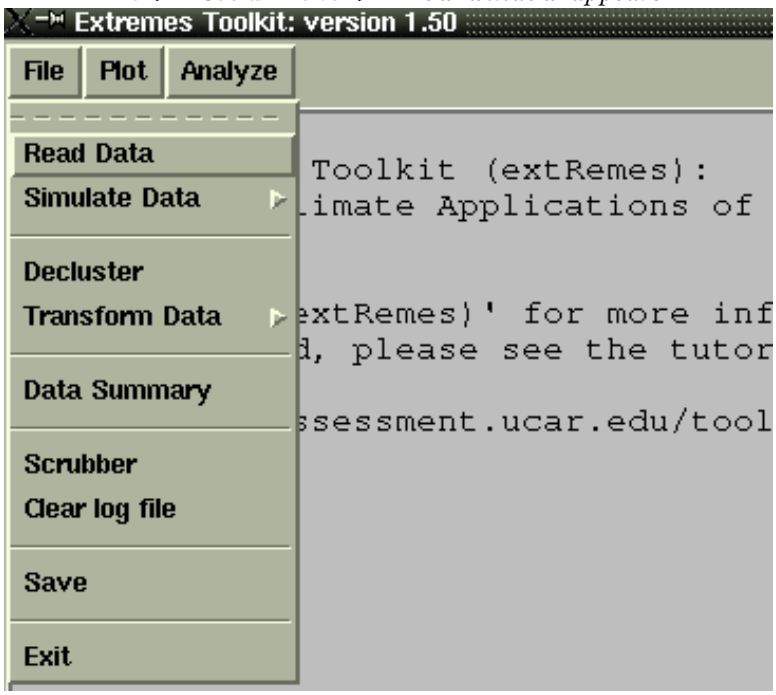
- **File > Read Data >** *New window appears.*
- *Browse for file* **Flood.dat > Open >** *Another new window appears.*

Leave the **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 **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 **Save As (in R)** name, say **Flood**, and click **OK**. A message in the R console should display that the file was read in correctly. The steps for this example, once again, are:

- 1. **File > Read Data >** *New window appears.*
- 2. *Browse for file* **Flood.dat > Open >** *Another new window appears.*
- 3. *Check* **Header**
- 4-5. *Enter* **Flood** *in* **Save As (in R)** *field* **> OK**.
- Message appears saying that file was successfully opened.

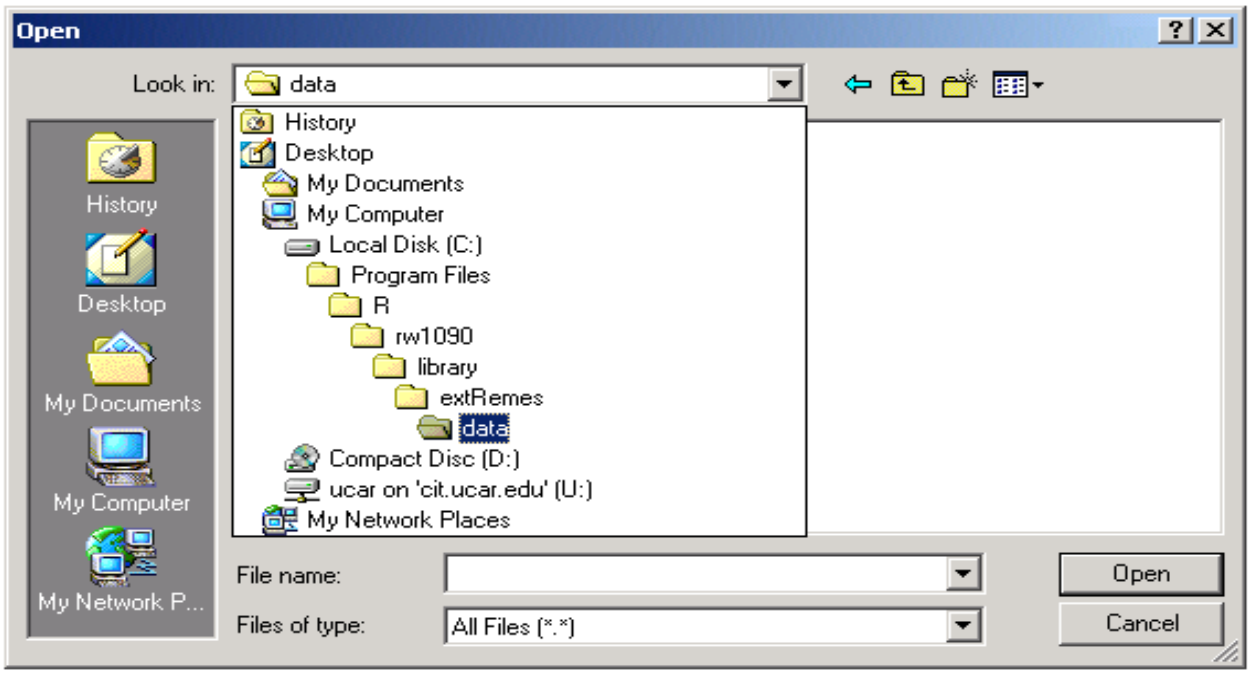
Each of the above commands will look something like the following on your computer screen. Note that the appearance of the toolkit will vary depending on the operating system used.

1. **File > Read Data >** *New window appears.*

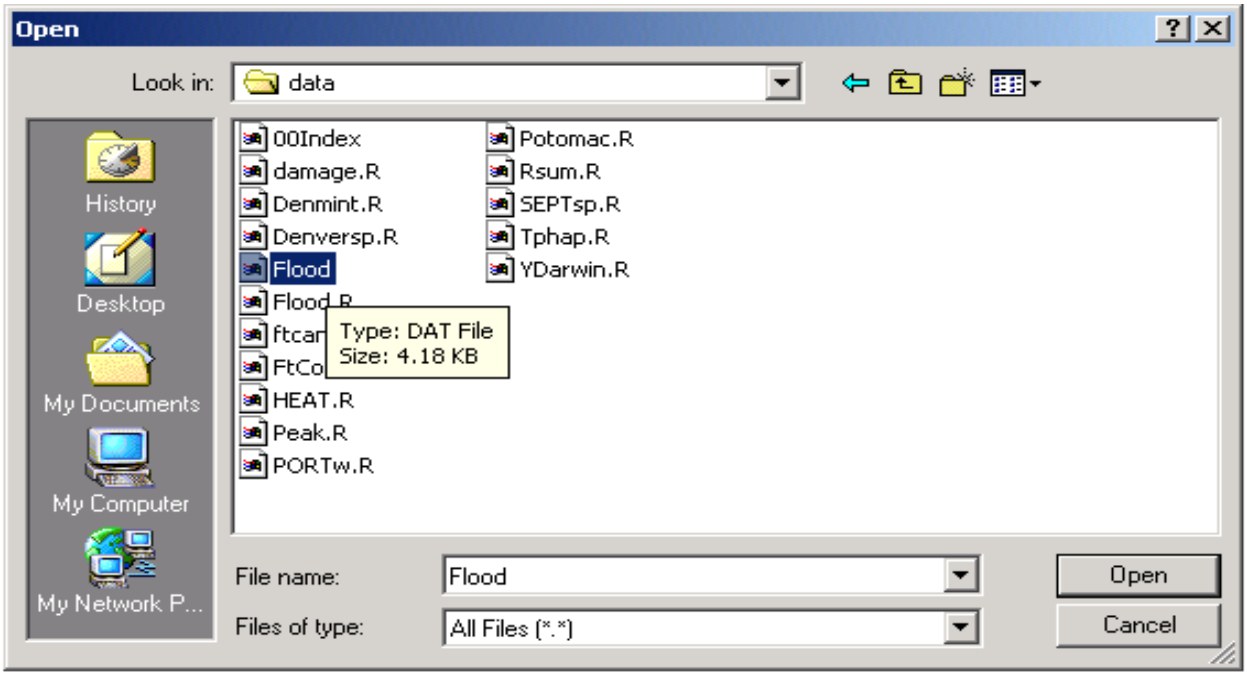


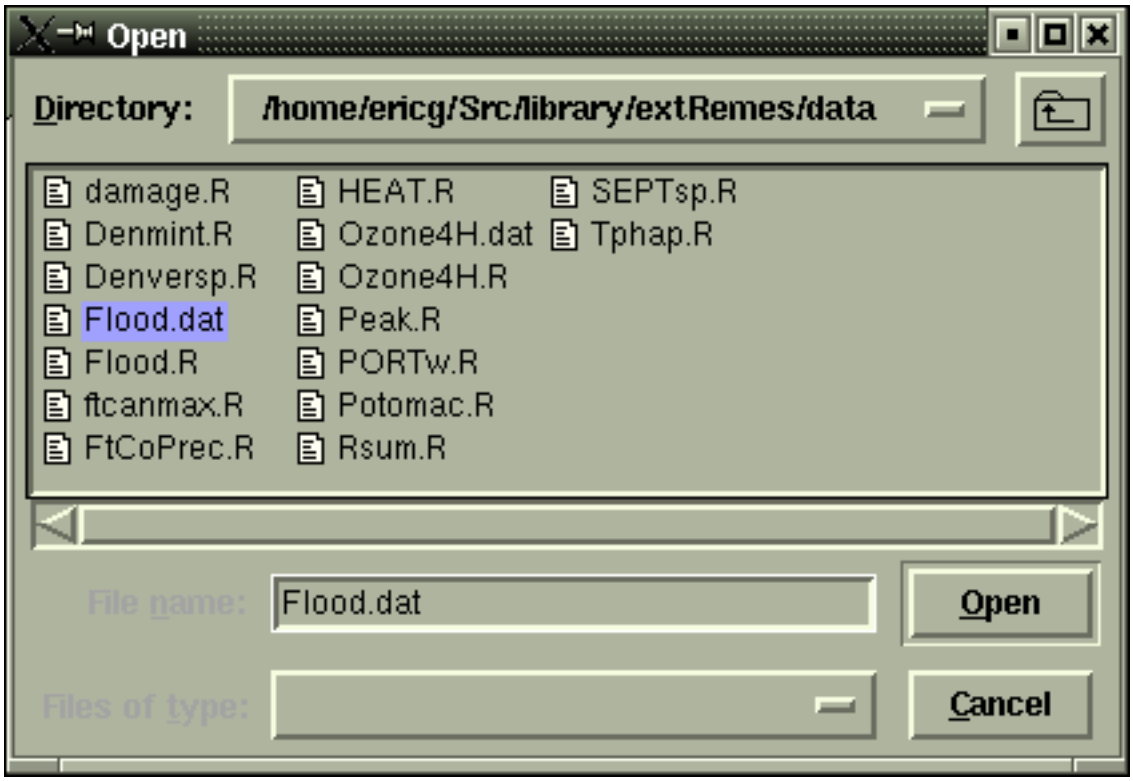
2. *Browse for file* **Flood.dat**[1] > **Open** > *Another new window appears.*

Note that the window appearances are system dependent. The following two screenshots show an example from a Windows operating system (OS), and the following shows a typical example from a Linux OS. If you cannot find these datasets in your `extRemes` data directory (likely with newer versions of R), then you can obtain them from the web at `http://www.isse.ucar.edu/extremevalues/data/`



[1]Note: there is also an R source file in this directory called **Flood.R**





3. *Check* **Header**

4-5. *Enter* **Flood** *in* **Save As (in R)** *field* > **OK**.


File Type
Common
R source
Delimiter:
Header
Save As (in R)
Flood
OK
Cancel

Message appears saying that file was successfully opened along with summary statistics for each column of the dataset. The current R workspace is then automatically saved with the newly loaded data.

```
> [1] "Successfully opened file: Flood.dat"
                     OBS      HYEAR     USDMG    DMGPC    LOSSPW
N              66.00000   66.00000 66.000000 66.00000   66.0000
mean           33.50000 1964.50000  2.629076 12.92844  270.5659
Std.Dev.       19.19635   19.19635  3.168426 13.88106  293.8702
min             1.00000 1932.00000  0.116800  0.92420   14.5300
Q1             17.25000 1948.25000  0.690225  3.78565   92.1600
median         33.50000 1964.50000  1.395600  7.53605  163.9700
Q3             49.75000 1980.75000  3.381075 16.55457  333.7825
max            66.00000 1997.00000 17.167800 68.32760 1453.1300
missing values  0.00000    0.00000  0.000000  0.00000    0.0000

 Saving workspace (may take a few moments) ...
```

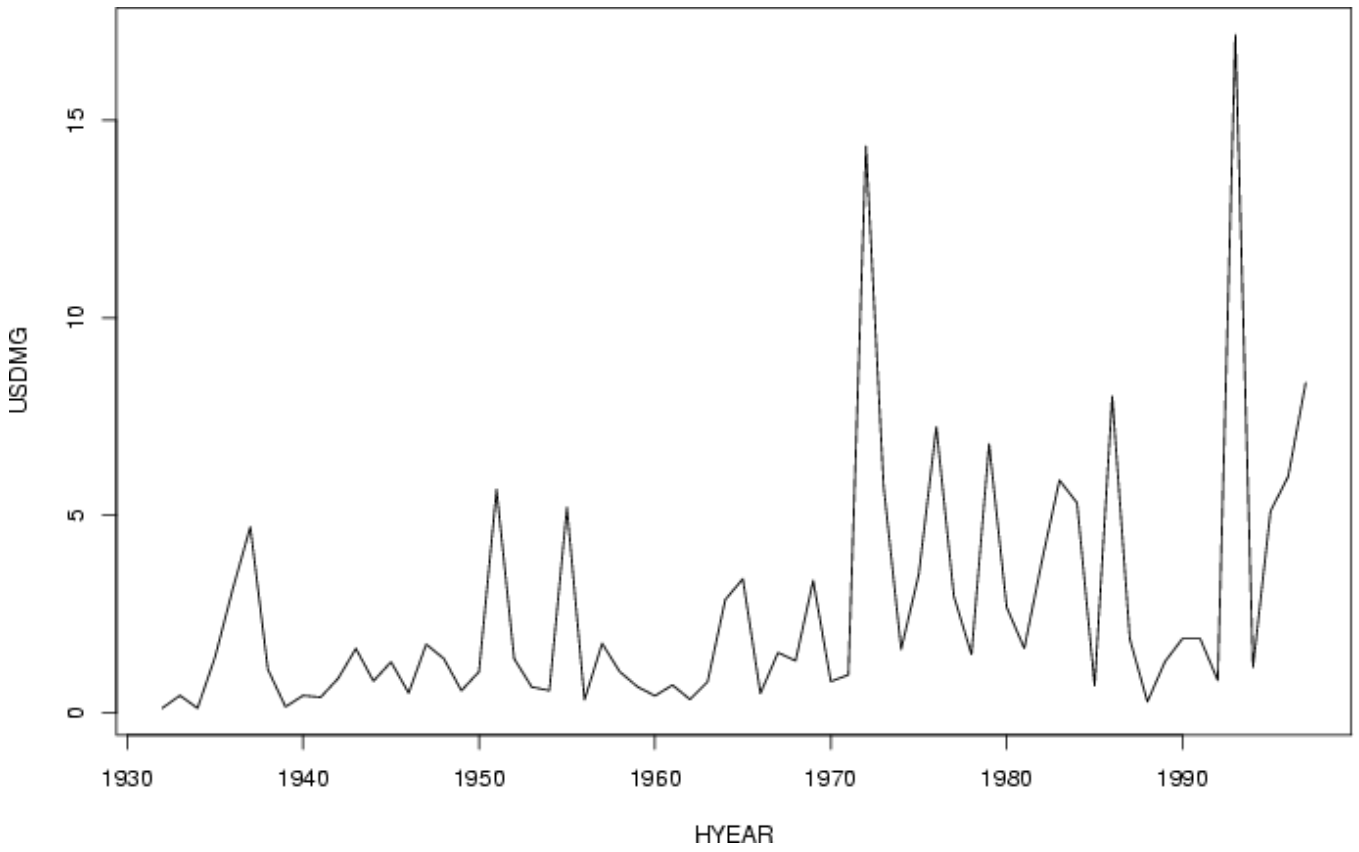
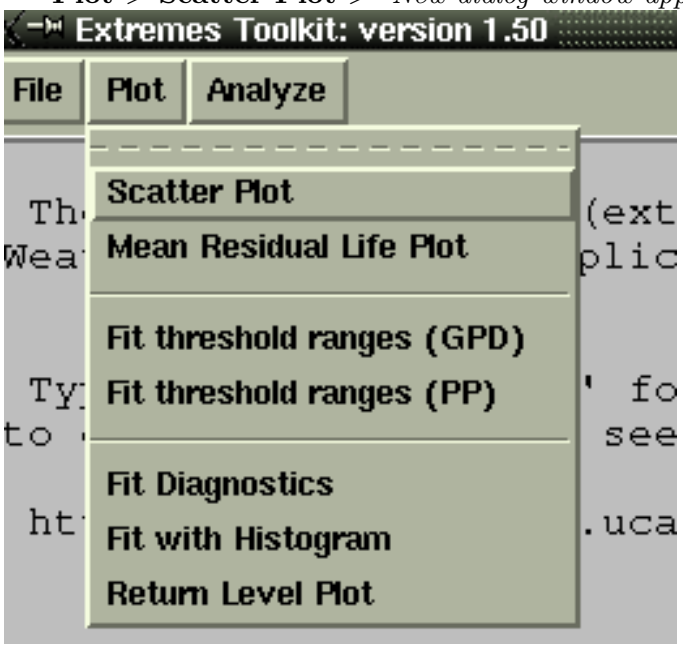


Figure 1.1: *Time series plot of total economic damage from U.S. floods (in billion $).*

Fig. 1.1 shows a time series plot of one of the variables from these data, **USDMG**. Although `extRemes` does not currently allow for time series data in the true sense (e.g., does not facilitate objects of class "ts"), such a plot can be easily created using the toolkit.

Plot > **Scatter Plot** > *New dialog window appears.*

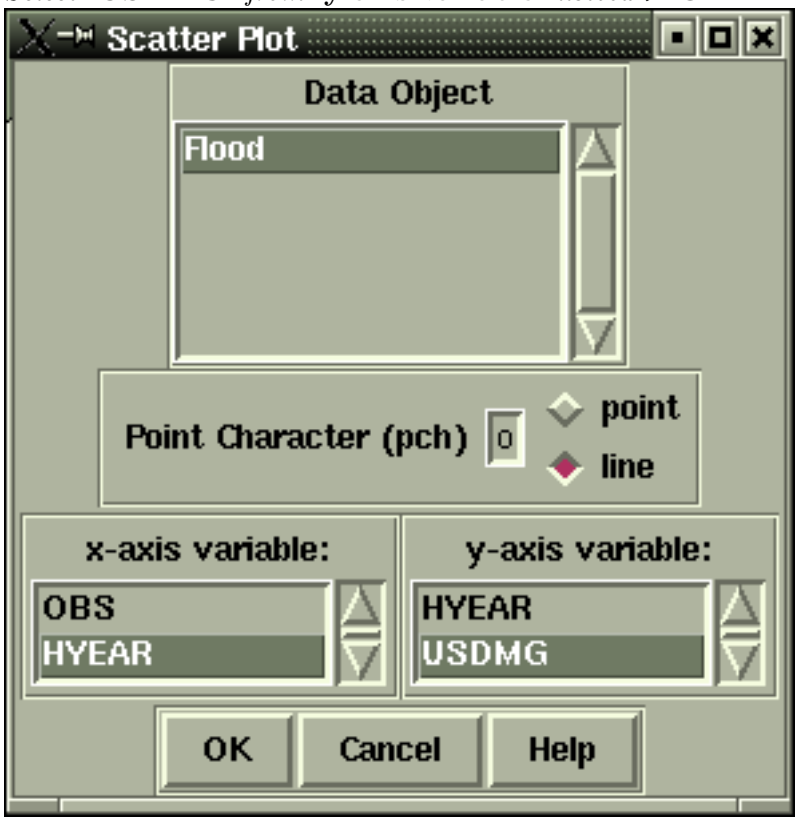


- *Select* **Flood** *from* **Data Object** *listbox.*

- *Select* **line** *from the* **Point Character (pch)** radiobuttons.

- *Select* **HYEAR** *from* **x-axis variable** *listbox.*

- *Select* **USDMG** *from* **y-axis variable** *listbox* > **OK**.



- Time series is plotted in a new window (it may be necessary 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 dataset, do the following.

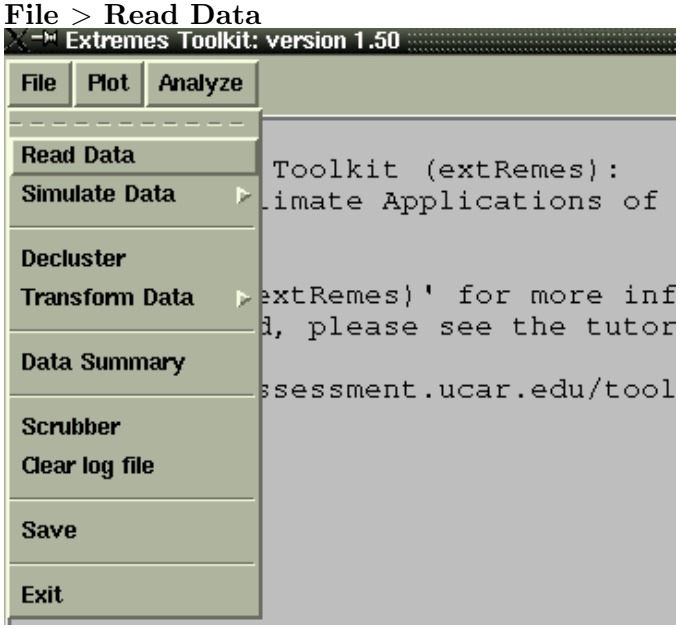
```
> Flood$data[1:3,]
```

Example 2: Loading an R source Dataset

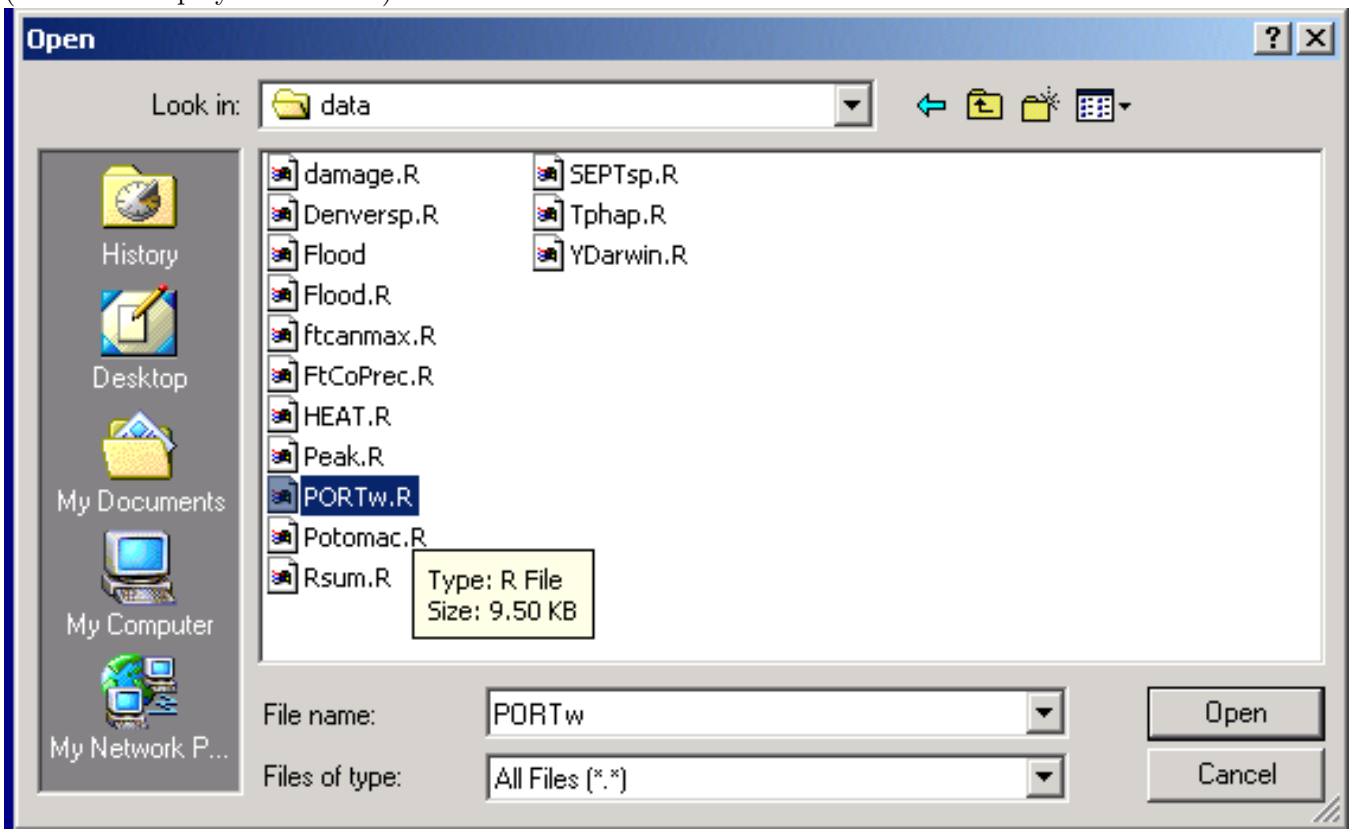
The data used in this example were provided by Linda Mearns of NCAR. The file *PORTw.R* consists of maximum winter temperature values for Port Jervis, N.Y. While the file contains other details of the dataset, the maximum temperatures are in the seventh column, labeled "TMX1". See Wettstein and Mearns [18] 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 click **Open** (or double click bf Portw.R).
(Windows display shown here)



Another window will appear 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.

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

A message is displayed that the file was successfully read along with a summary of the data. 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).

1.2.2 Simulating data from a GEV distribution

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

The general procedure for simulating data from a GEV distribution 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 1.2.1, it is a good idea to enter a name in the **Save As** field. After entering the options, click on **Generate** 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 Fig. 1.2, is produced.

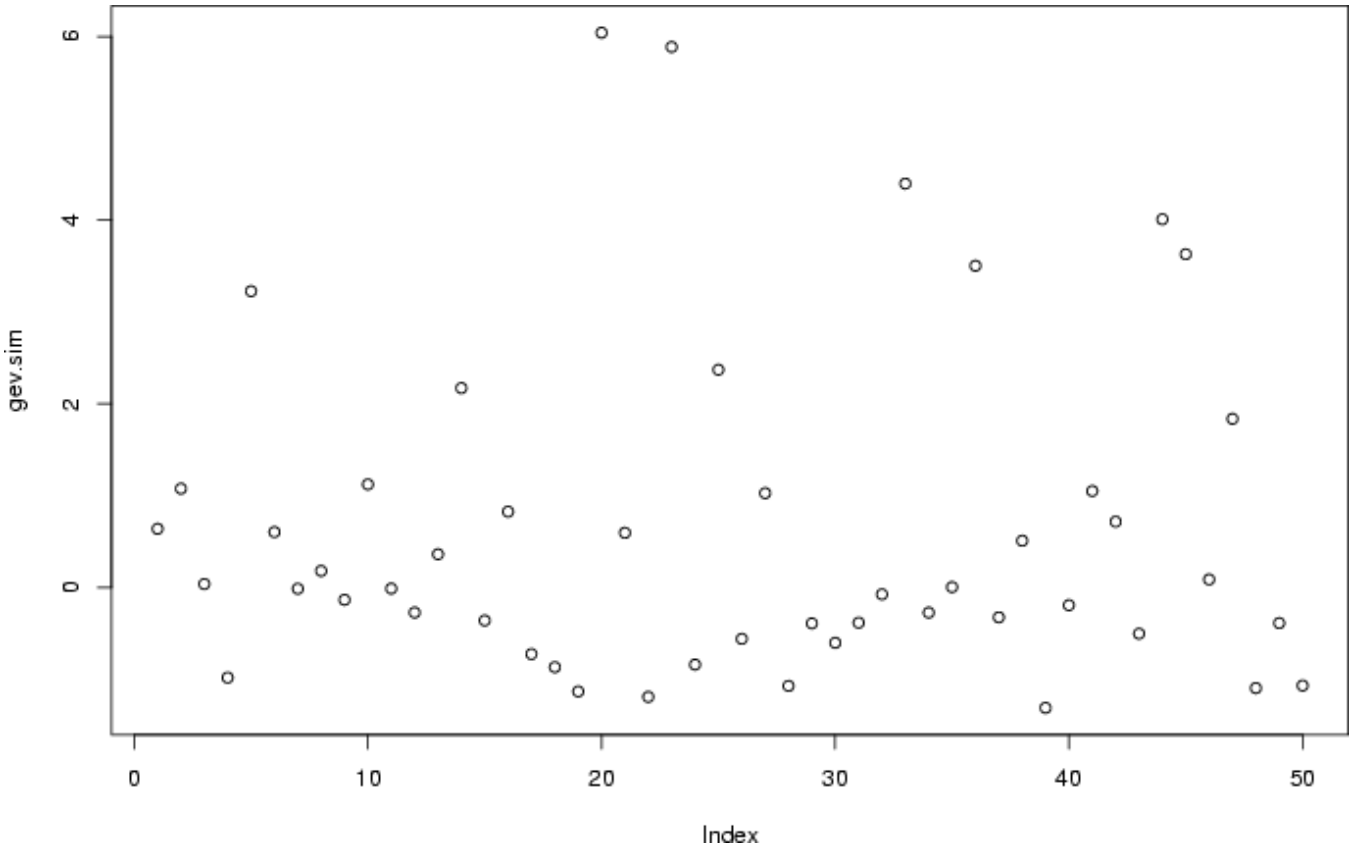
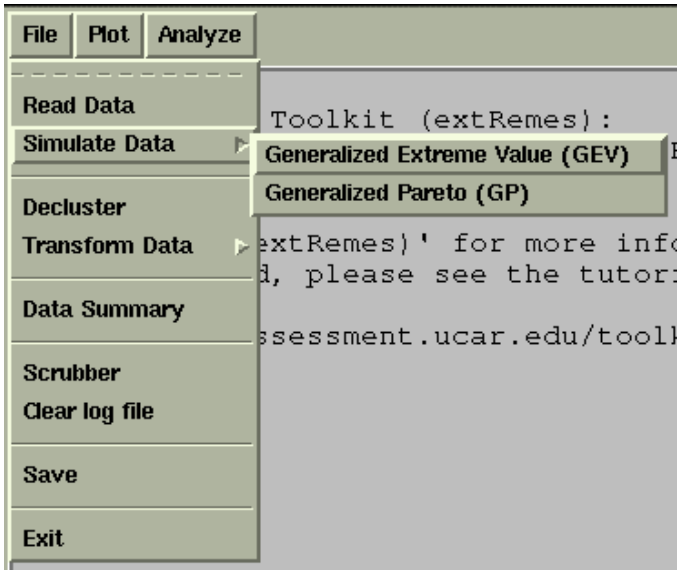


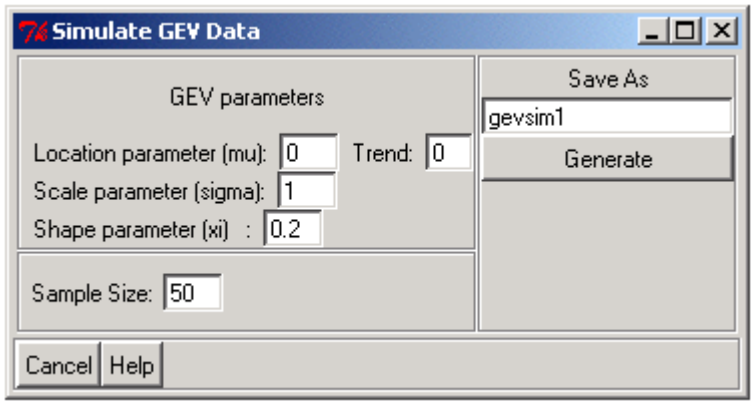
Figure 1.2: *Plot of data simulated from a GEV distribution using all default values:* $\mu = 0$, *trend*$= 0$, $\sigma = 1$, $\xi = 0.2$ *and sample size*$= 50$.

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

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



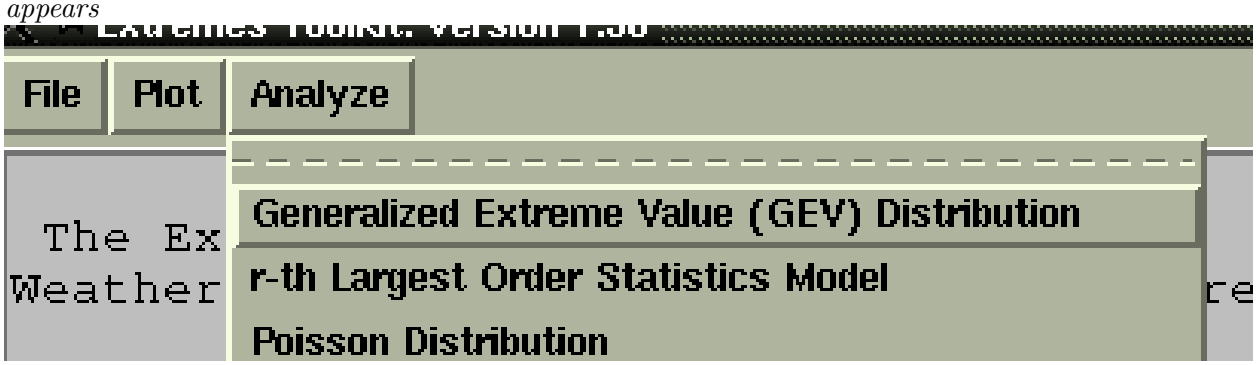
- *Enter* **gevsim1** *in the* **Save As** *field* > **Generate**



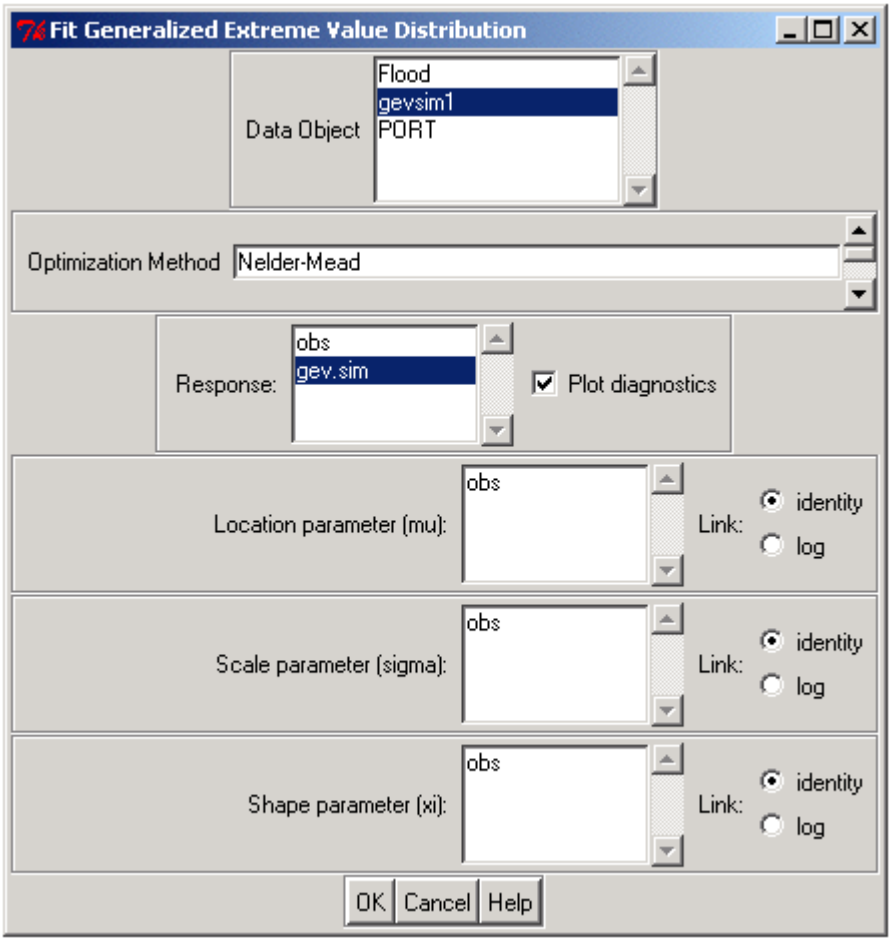
- Plot appears, message on main toolkit window displays parameter choices and an object of class "ev.data" is saved with the name **gevsim1**.

Once a dataset has been successfully loaded or simulated, work may begin on its analysis. The Extremes Toolkit provides for fitting data to the GEV, Poisson and generalized Pareto (GPD) distributions as well as fitting data to 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 are generated randomly each time. To fit a GEV to the simulated data, do the following.

- **Analyze** > **Generalized Extreme Value (GEV) Distribution** > *New window appears*



- *Select* **gevsim1** *from the* **Data Object** *listbox.*
- *Select* **gev.sim** *from the* **Response** *listbox.*
- *Check the* **Plot diagnostics** *checkbutton.* > **OK**



A plot similar to the one in Fig. 1.3 should appear. For information on these plots please see section 2.0.5. 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 Fig. 1.3 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, μ, should be near zero, the scale parameter, σ, near one and the shape parameter, ξ, 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.

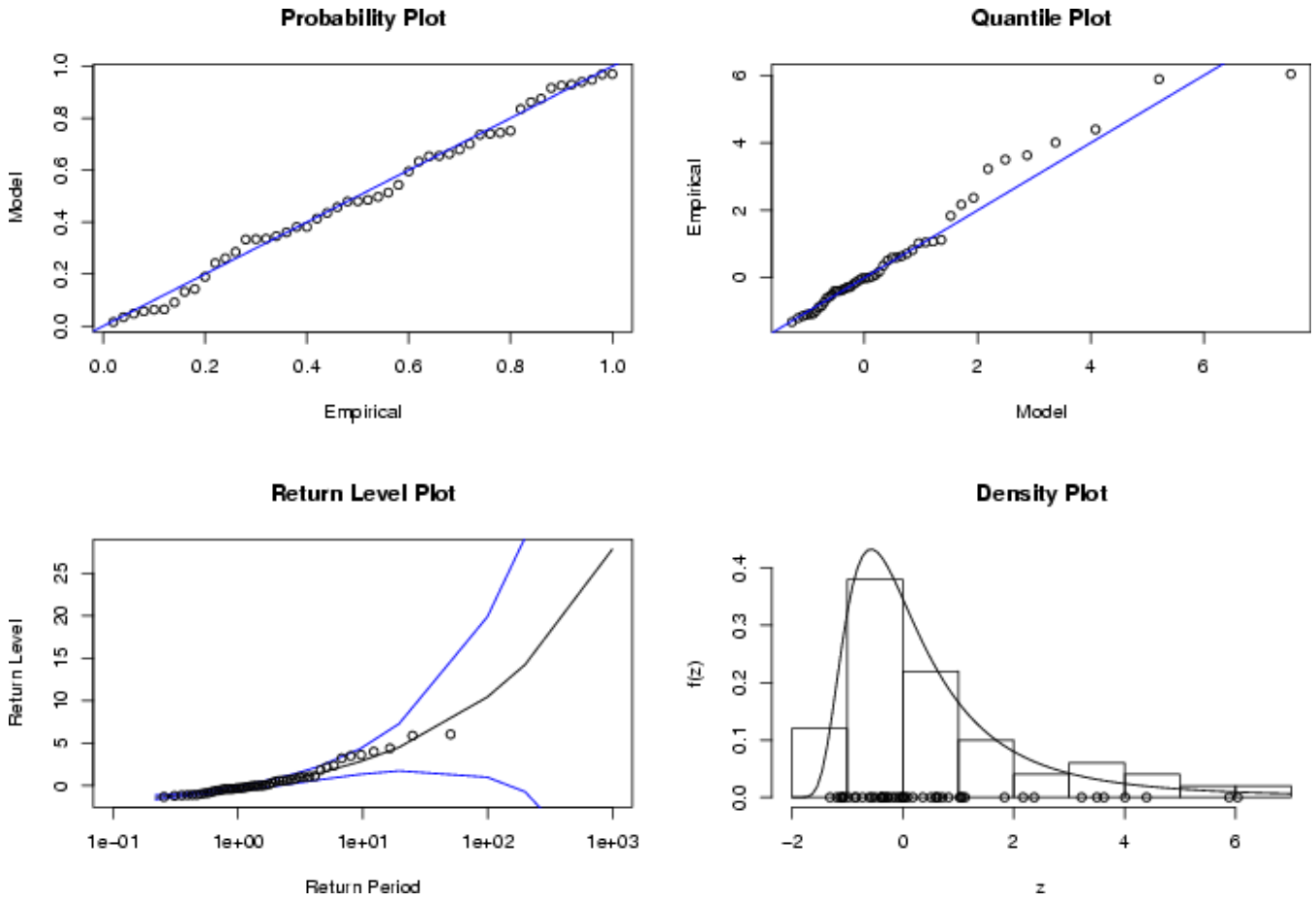
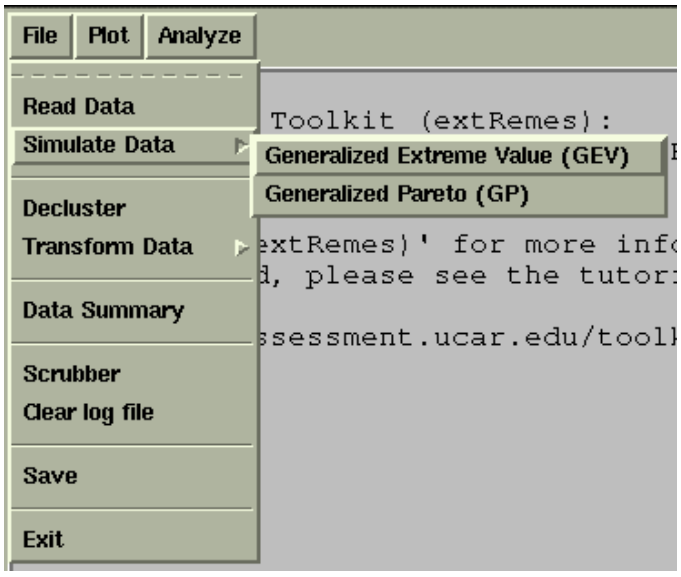


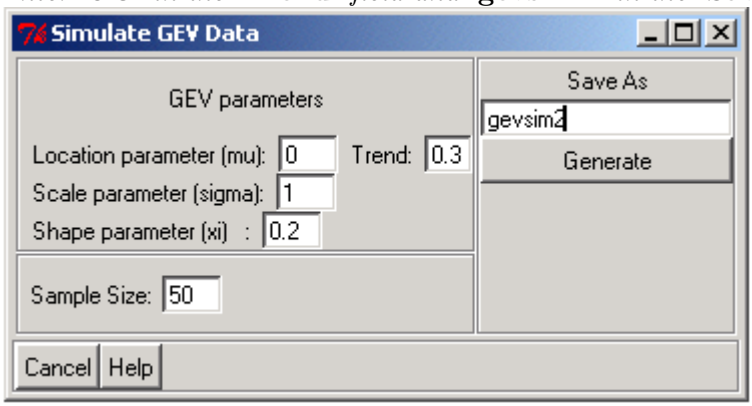
Figure 1.3: *Diagnostic plots for GEV fit to a simulated dataset.*

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 μ_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)**

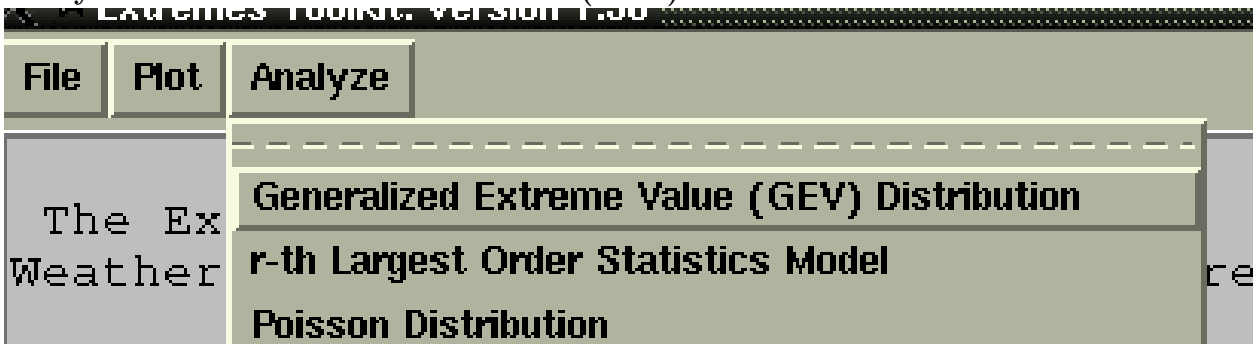


- *Enter* **0.3** *in the* **Trend** *field and* **gevsim2** *in the* **Save As** *field* > **Generate**.

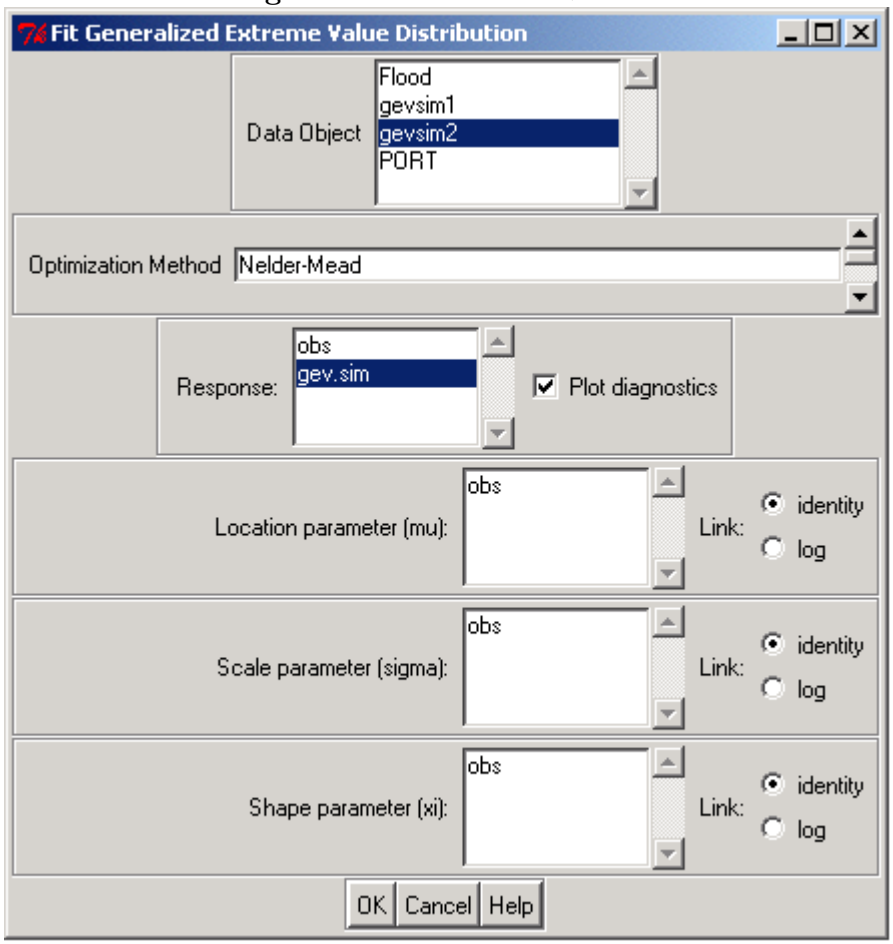


The trend should be evident from the scatter plot. Now, first fit the GEV without a trend in the location parameter.

- **Analyze > Generalized Extreme Value (GEV) Distribution**



- *Select* **gevsim2** *from the* **Data Object** *listbox.*
- *Select* **gev.sim** *from the* **Response** *listbox.*
- *Check the* **Plot diagnostics** *checkbutton.* > **OK**.



A plot similar to that of Fig. 1.4 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.

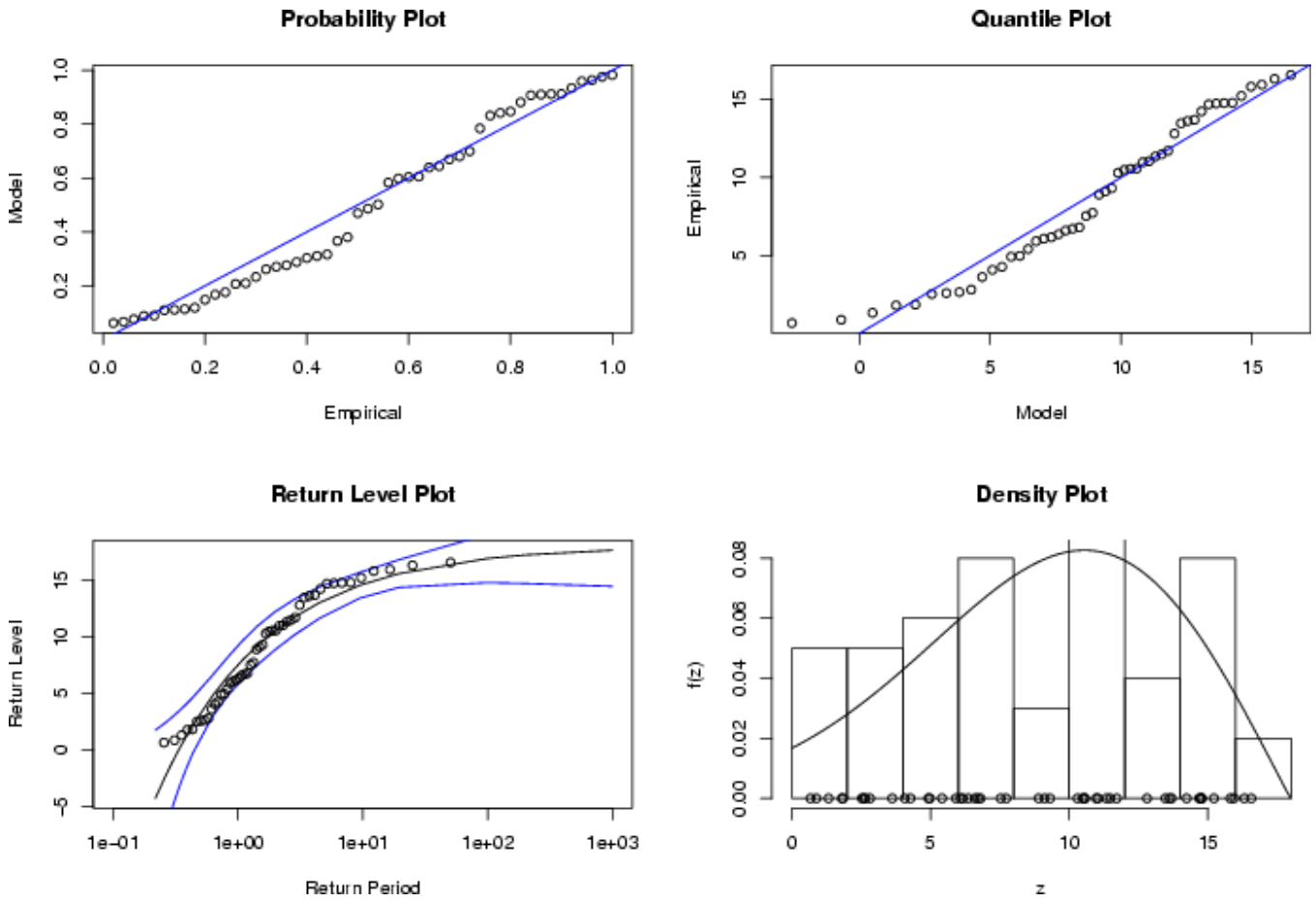
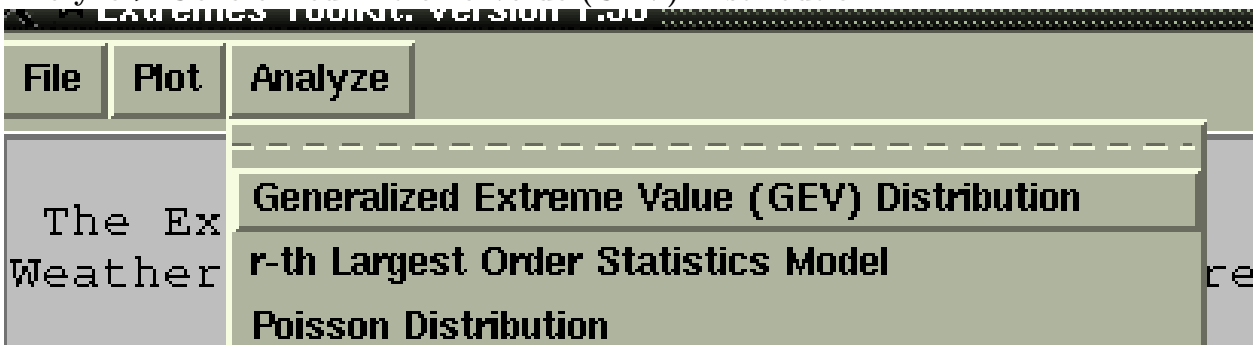
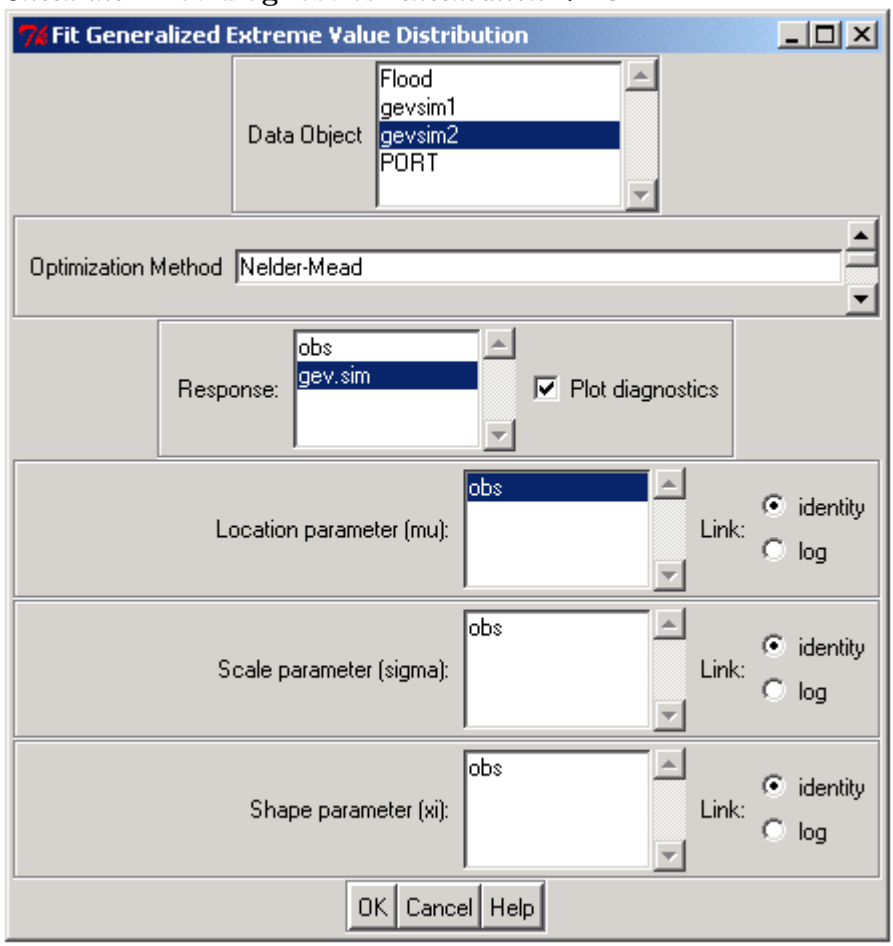


Figure 1.4: *Simulated data from GEV distribution with trend in location parameter fit to GEV distribution without a trend.*

- **Analyze > Generalized Extreme Value (GEV) Distribution**



- *Select* **gevsim2** *from the* **Data Object** *listbox.*
- *Select* **gev.sim** *from the* **Response** *listbox.*
- *Select* **obs** *from the* **Location Parameter (mu)** *listbox (leave identity as link function).*
- *Check the* **Plot diagnostics** *checkbutton.* > **OK**.



Notice that only the top two diagnostic plots are plotted when incorporating a trend into the fit as in Fig. 1.5. The fit appears, not surprisingly, to be much better. In this case, the MLE for the location parameter is $\hat{\mu} \approx 0.27 + 0.297 \cdot \texttt{obs}$ 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.

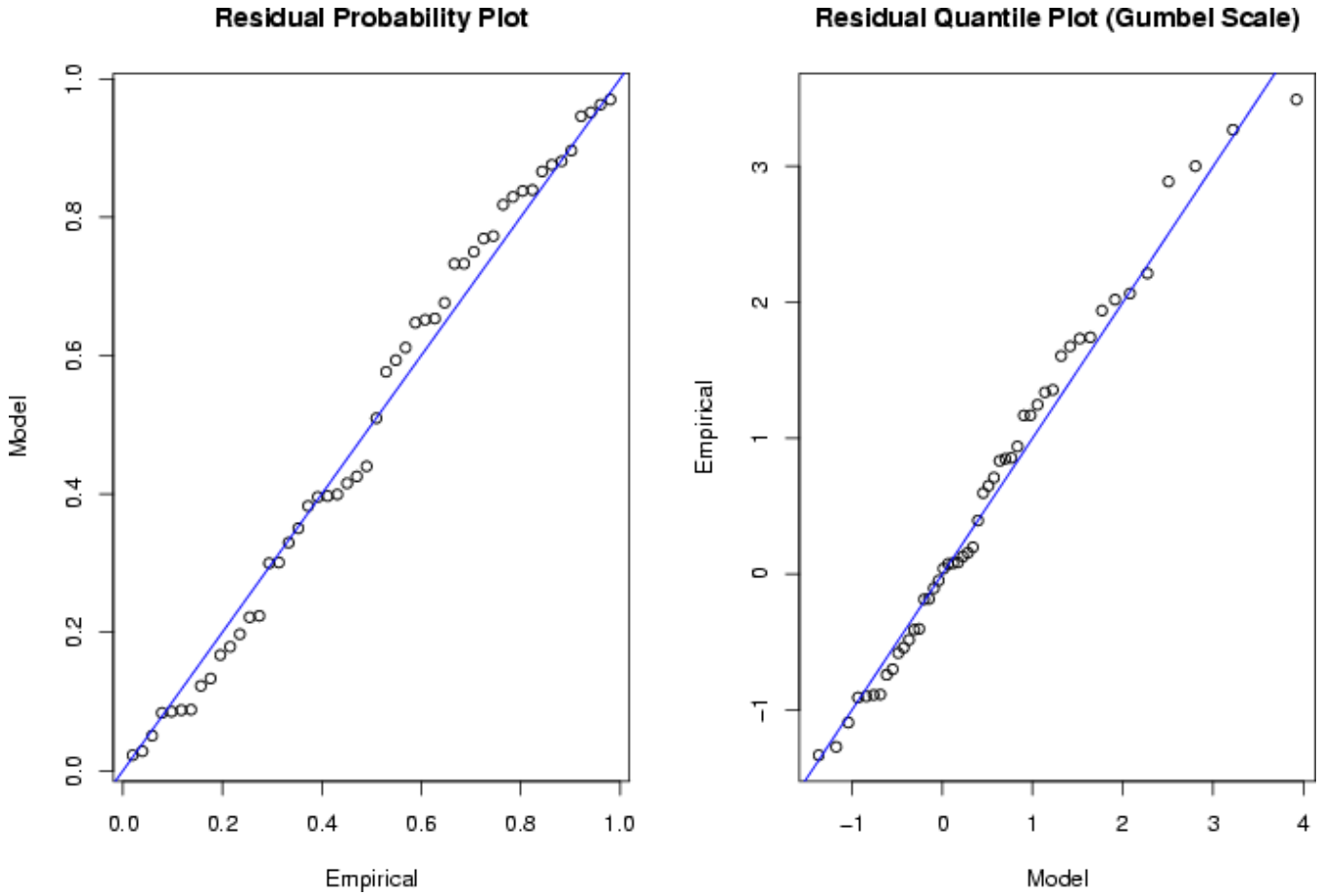
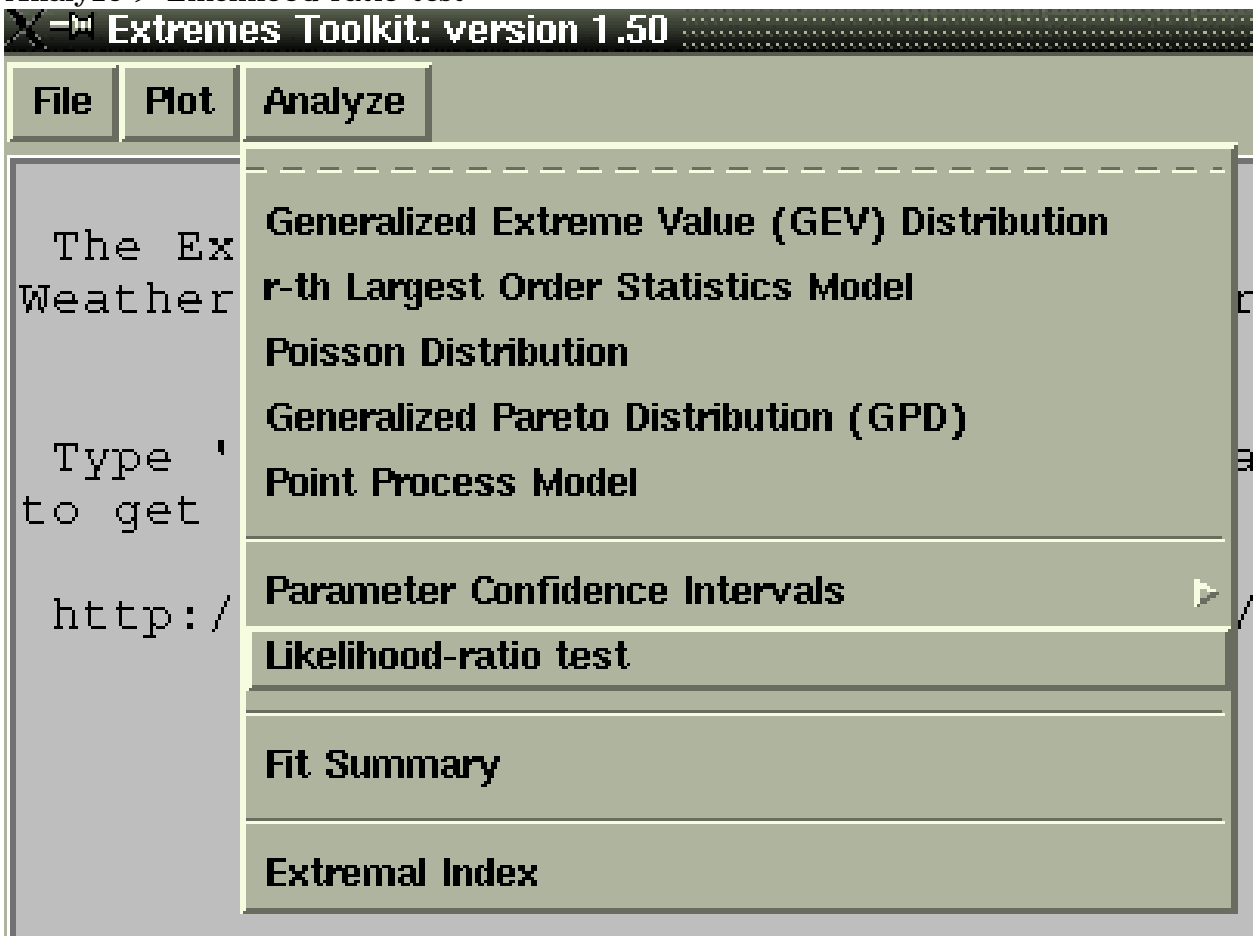


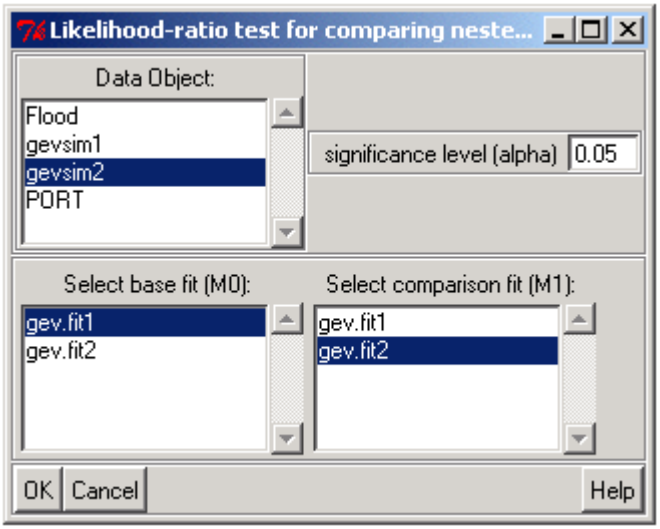
Figure 1.5: *Simulated data from GEV distribution with trend in location parameter fit to GEV distribution with a trend.*

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

- **Analyze > Likelihood-ratio test**



- *Select* **gevsim2** *from the* **Data Object** *listbox.*
- *Select* **gev.fit1** *from the* **Select base fit (M0)** *listbox.*
- *Select* **gev.fit2** *from the* **Select comparison fit (M1)** *listbox* > **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 χ^2_1 distribution of only 3.8415 and p-value approximately zero.

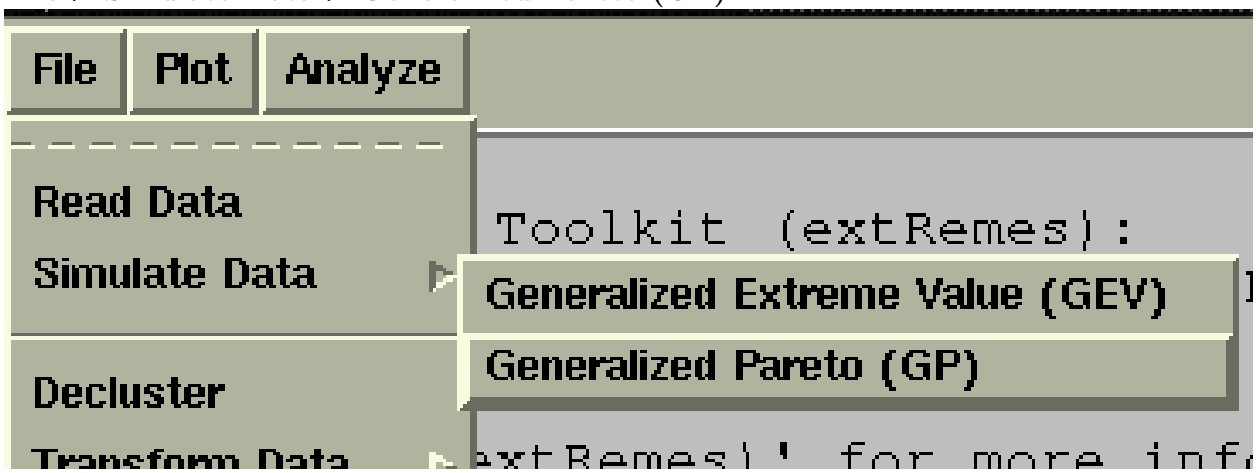
1.2.3 Simulating data from a GPD

It is also possible to sample from a Generalized Pareto Distribution (GPD) using the toolkit. For more information on the GPD please see section 5.0.10. The general procedure for simulating from a GPD is as follows.

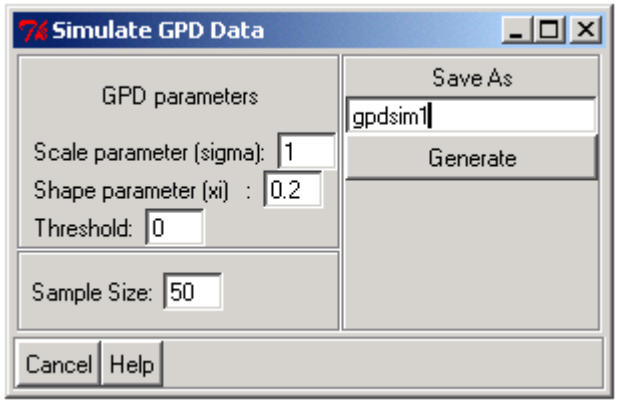
- **File > Simulate Data > Generalized Pareto (GP)**
- *Enter options and a* **Save As** *name* > **Generate**
- A scatter plot of the simulated data appears, a message on the main toolkit window displays chosen parameter values and an object of class "ev.data" is created.

Fig. 1.6 shows the scatter plot for one such simulation. As an example, simulate a GP dataset in the following manner.

- **File > Simulate Data > Generalized Pareto (GP)**



- *Leave the parameters on their defaults and enter* **gpdsim1** *in the* **Save As** *field* > **Generate**

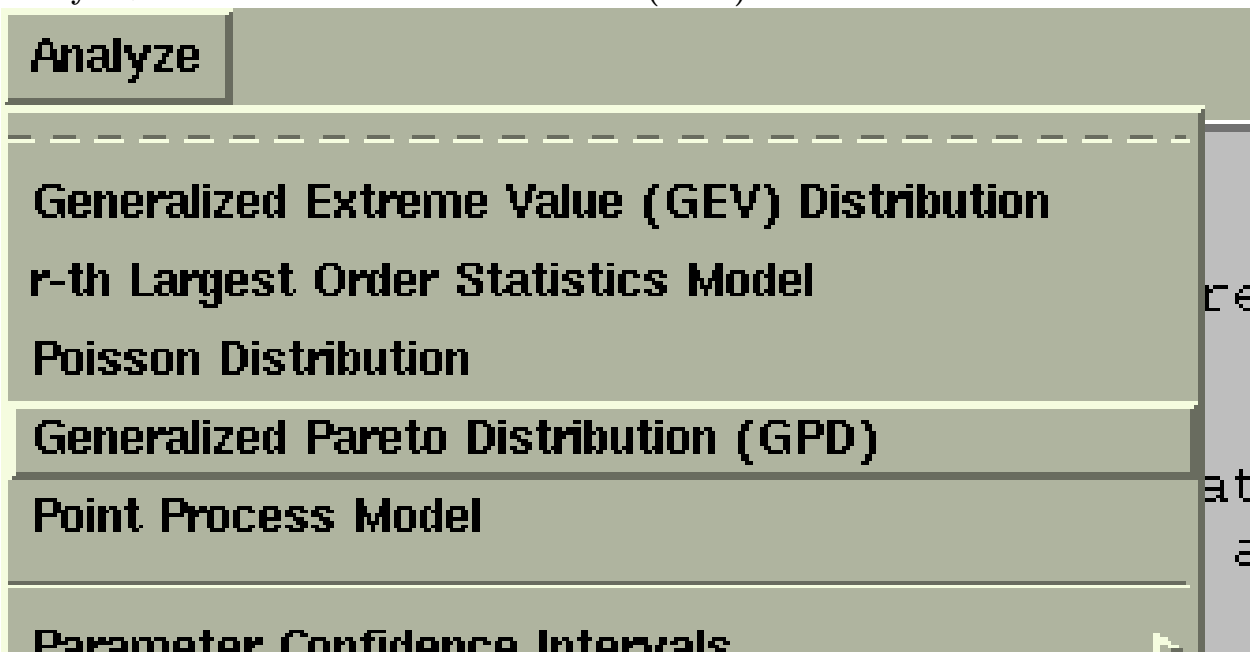


- A scatter plot of the simulated data appears and a message on main toolkit window displays chosen parameter values and an object of class "ev.data" is created.

You should see a plot similar to that of Fig. 1.6, but not the same because each simulation will yield different values. The next logical step would be to fit a GPD to these simulated data.

To fit a GPD to these data, do the following.

- **Analyze** > **Generalized Pareto Distribution (GPD)**



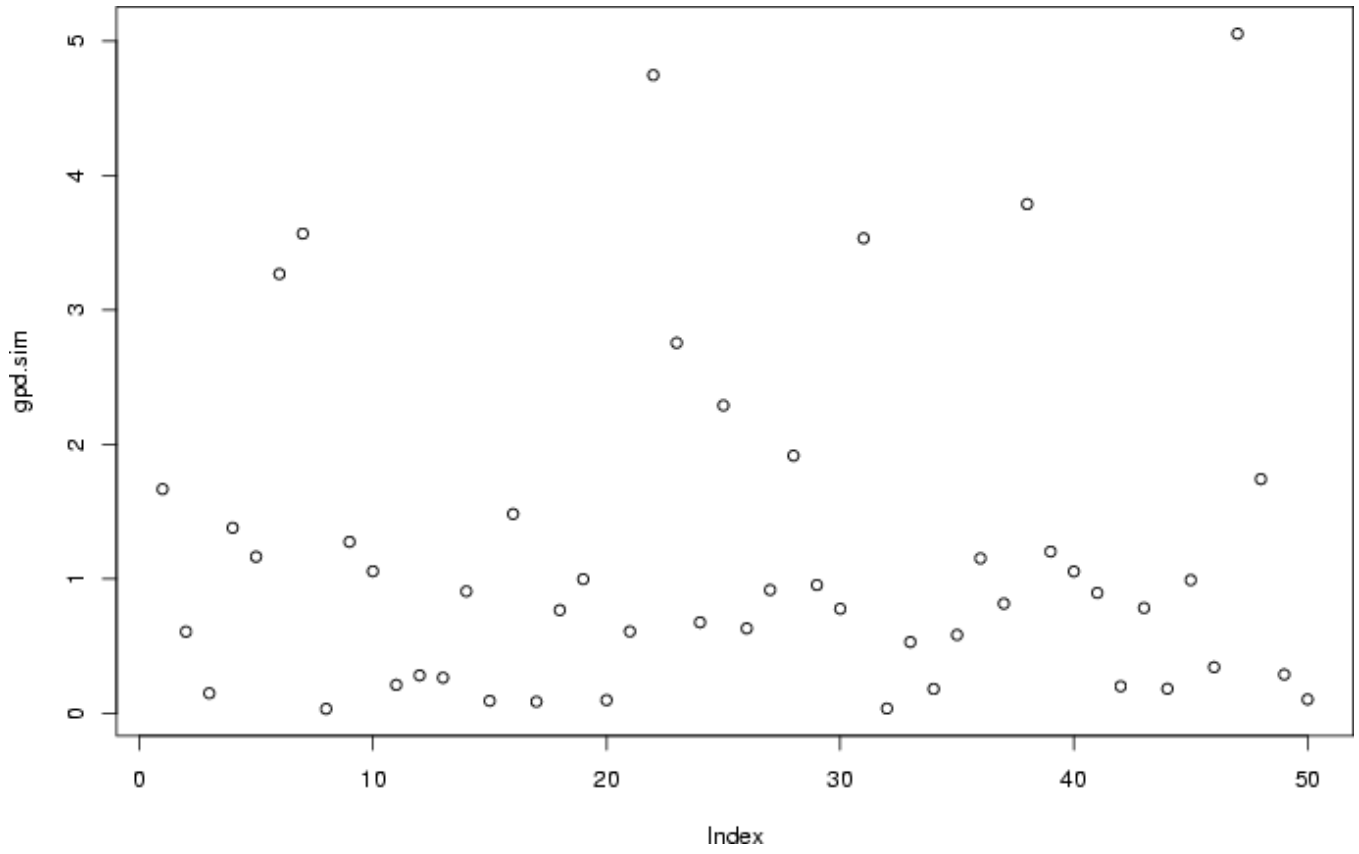
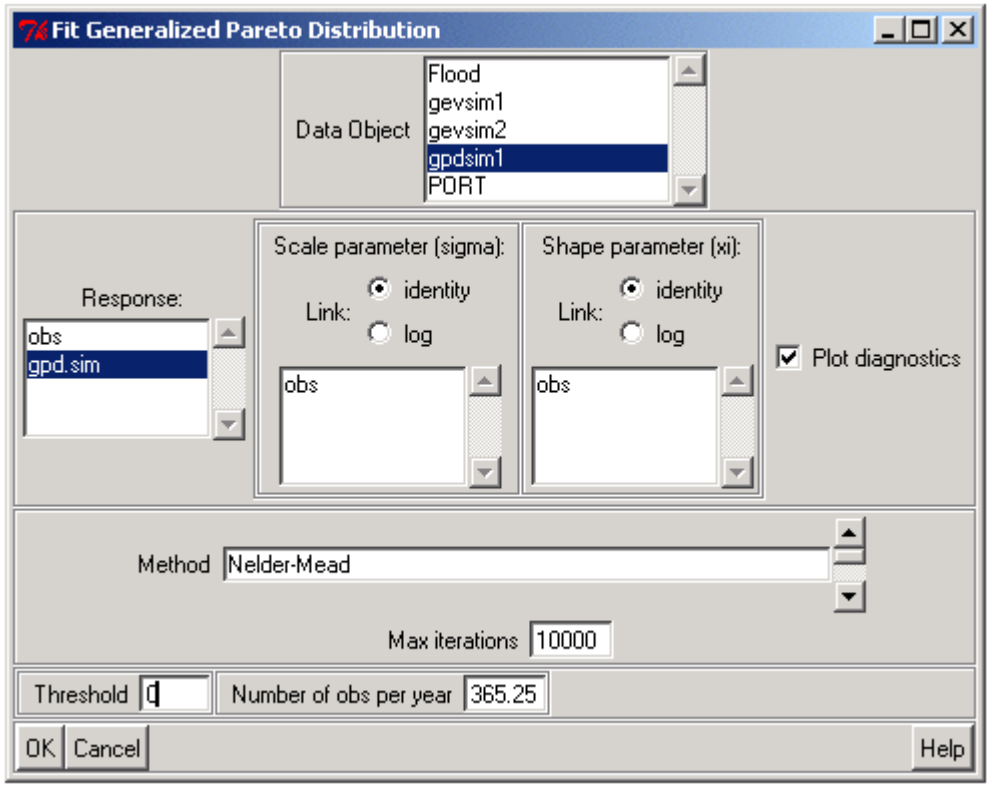


Figure 1.6: *Scatter plot of one simulation from a GPD using the default values for parameters.*

- *Select* **gpdsim1** *from the* **Data Object** *listbox.*
- *Select* **gpd.sim** *from the* **Response** *listbox.*
- *Check* **Plot diagnostics** *checkbutton*
- *Enter* 0 *(zero) in the* **Threshold** *field* > **OK**



Plots similar to those in Fig. 1.7 should appear, but again, 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.

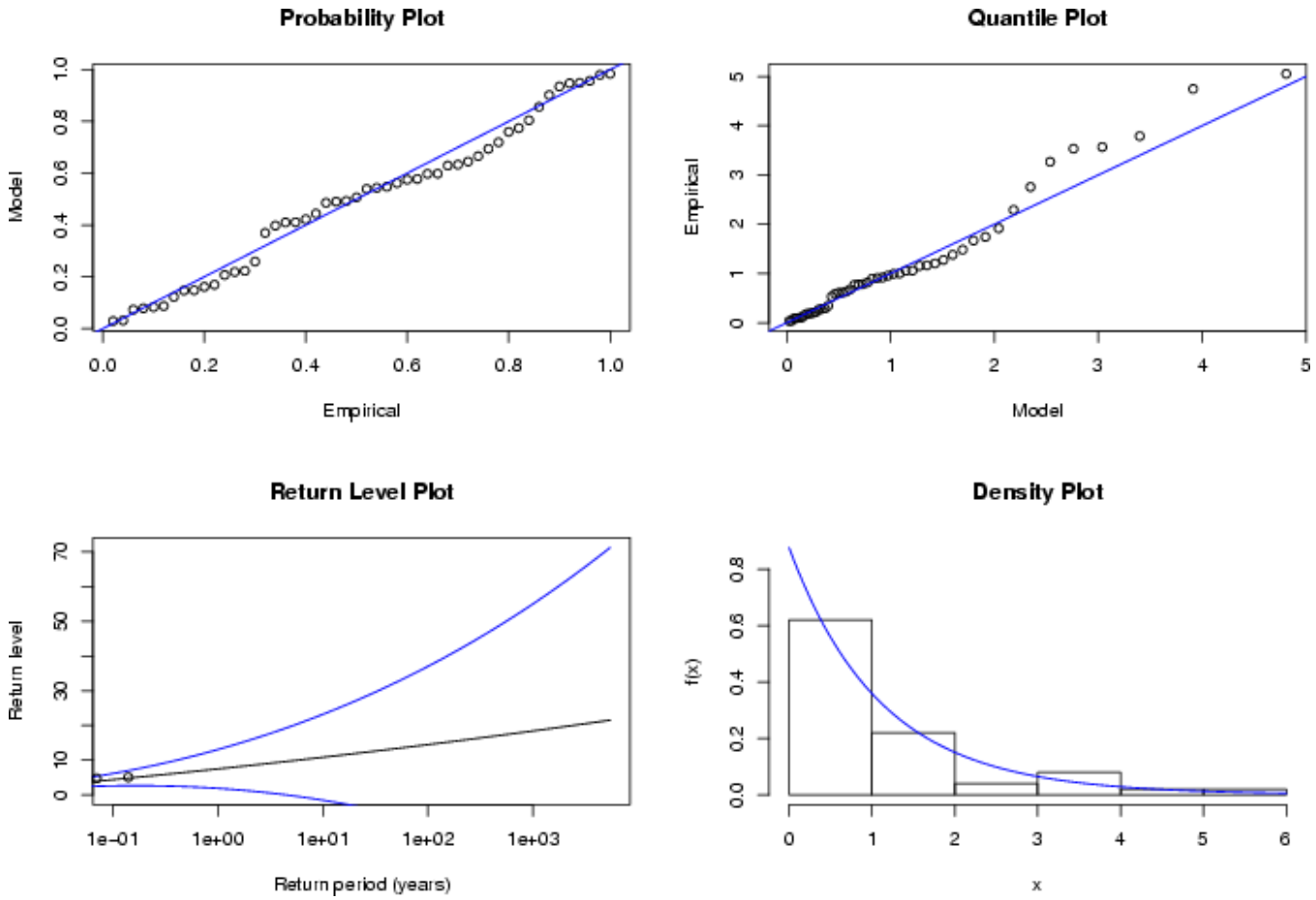


Figure 1.7: *Diagnostic plots from fitting one simulation from the GP distribution to the GP distribution.*

1.2.4 Loading an R Dataset from the Working Directory

Occasionally, it may be of interest to load a dataset either created in the R session working directory or brought in from an R package. For example, the internal toolkit functions are primarily those of the R package **ismev**, which consist of Stuart Coles' functions [3] and example datasets. It may, therefore, be of interest to use the toolkit to analyze these datasets. Although these data could be read using the toolkit and browsing to the `ismev` **data** directory as described in section 1.2.1, this section gives an alternative method. Other times, data may need to be manipulated in a more advanced manner than `extRemes` will allow, but subsequently used with `extRemes`.

An `extRemes` data object must be a list object with at least a component called `data`, which must be a matrix or data frame; the columns of which must be named. Additionally, the object must be assigned the class, `"ev.data"`.

Example: Loading the Wooster temperature dataset from `ismev` package

From the R session window.

```
> data( wooster)
> Wooster <- list( data=wooster)
> Wooster$data <- matrix( Wooster$data, ncol=1)
> colnames( Wooster$data) <- "Temperature"
> class( Wooster) <- "ev.data"
```

Chapter 2

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 (see Coles [3] Ch. 3). The block maxima approach is closely associated with the use of the GEV family. Note that all parameters are always estimated (with `extRemes`) by maximum likelihood estimation (MLE), which requires iterative numerical optimization techniques. See Coles [3] section 2.6 on parametric modeling for more information on this optimization method.

2.0.5 Fitting data to a GEV distribution

The general procedure for fitting data to a GEV distribution with `extRemes` is

- **Analyze** > **Generalized Extreme Value (GEV) Distribution** > *New window appears.*
- *Select data object from* **Data Object** *listbox* > *column names appear in other listboxes.*
- *Choose a response variable from the* **Response** *listbox* > *Response variable is removed as an option from other listboxes.*
- *Select other options as desired* > **OK**
- A GEV distribution will be fitted to the chosen response variable and stored in the same list object as the data used.

Example 1: Port Jervis data

This example uses the **PORT** dataset (see section 1.2.1) to illustrate fitting data to a GEV using `extRemes`. If you have not already loaded these data, please do so before trying

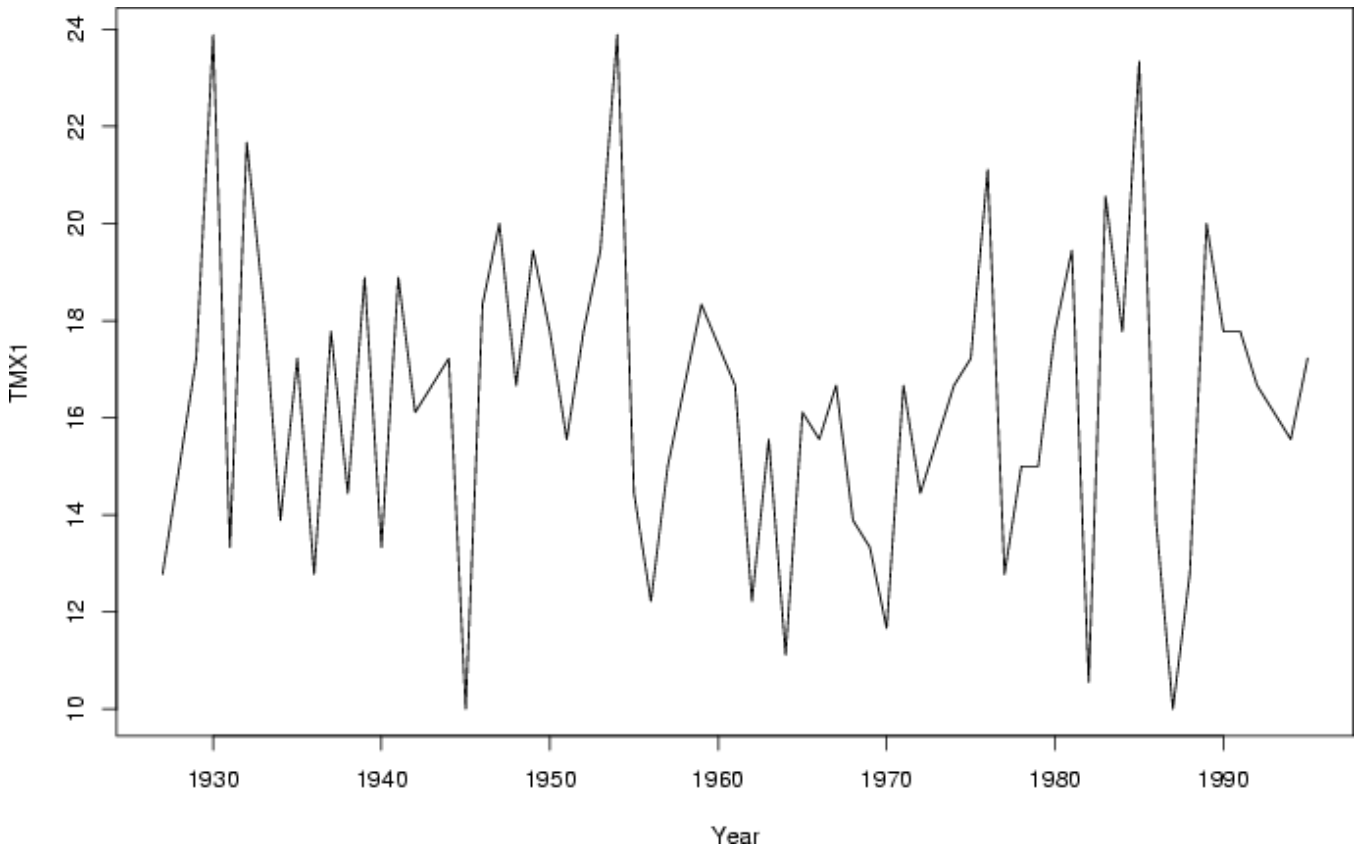


Figure 2.1: *Time series of Port Jervis annual (winter) maximum temperature (degrees centigrade).*

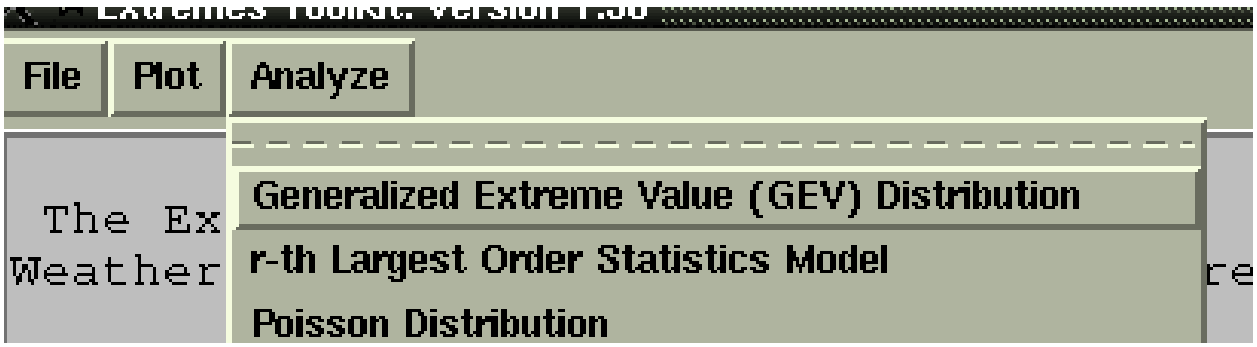
this example. Fig. 2.1 shows a time series of the annual (winter) maximum temperatures (degrees centigrade).

From the main window, select

Analyze > Generalized Extreme Value (GEV) Distribution.

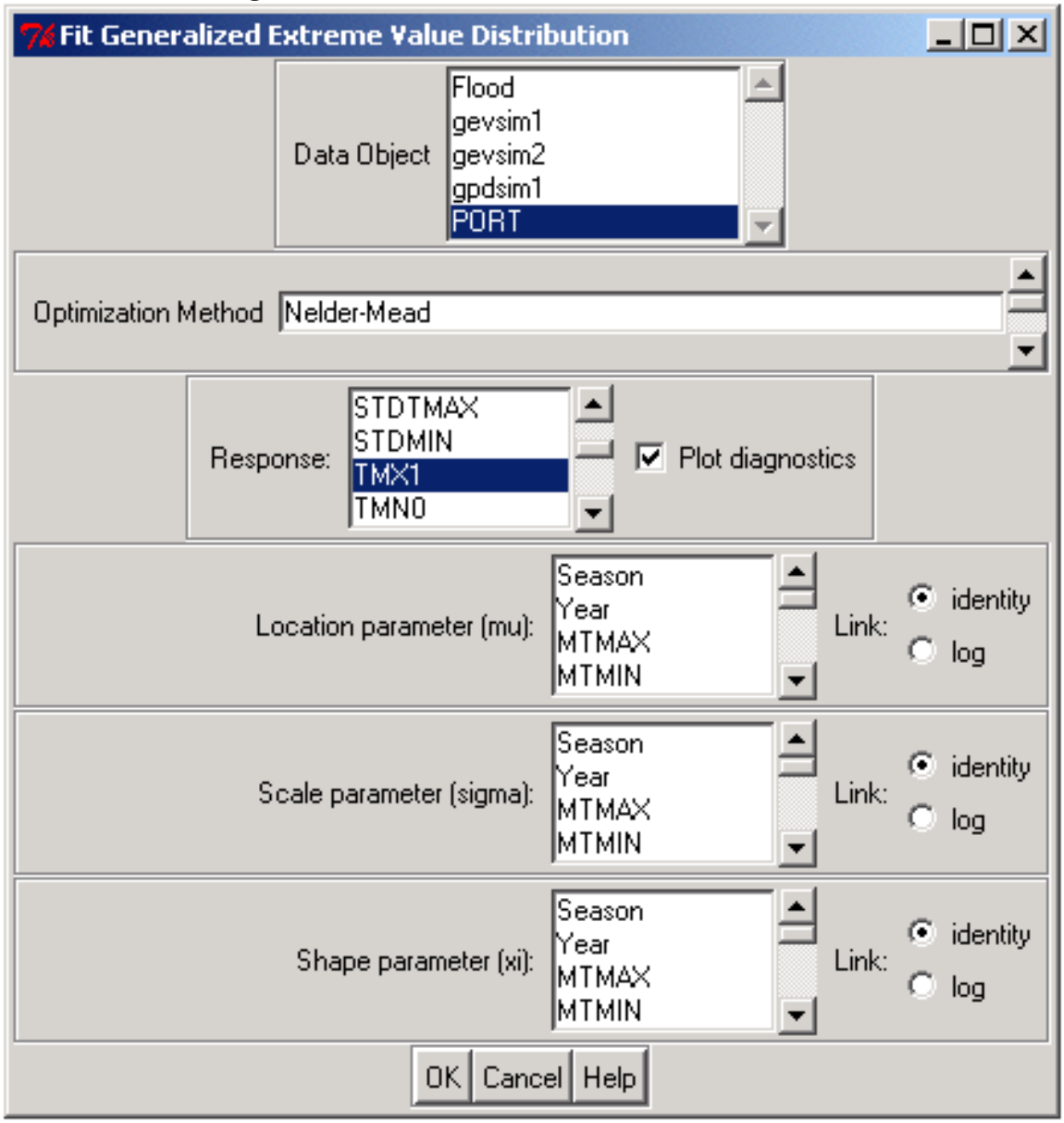
A new dialog window appears requesting the details of the fit. First, select **PORT** from the **Data Object** listbox. Immediately, the listboxes for **Response**, **Location parameter (mu)**, **Scale parameter (sigma)** and **Shape parameter (xi)** should now contain the list of covariates for these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

- **Analyze > Generalized Extreme Value (GEV) Distribution** > *New window appears*



- *Select* **PORT** *from* **Data Object** *listbox. Column names appear in other listboxes.*
- *Choose* **TMX1** *from the* **Response** *listbox* (*Note that* **TMX1** *is removed as an option from other listboxes*).

- *Click on the* **Plot diagnostics** *checkbutton* > **OK**.



- Here, we ignore the rest of the fields because we are not yet incorporating any covariates into the the fit.

An R graphics window appears displaying the probability and quantile plots, a return-level plot, and a density estimate plot as shown in Fig. 2.2. In the case of perfect fit, the data would line up on the diagonal of the probability and quantile plots.

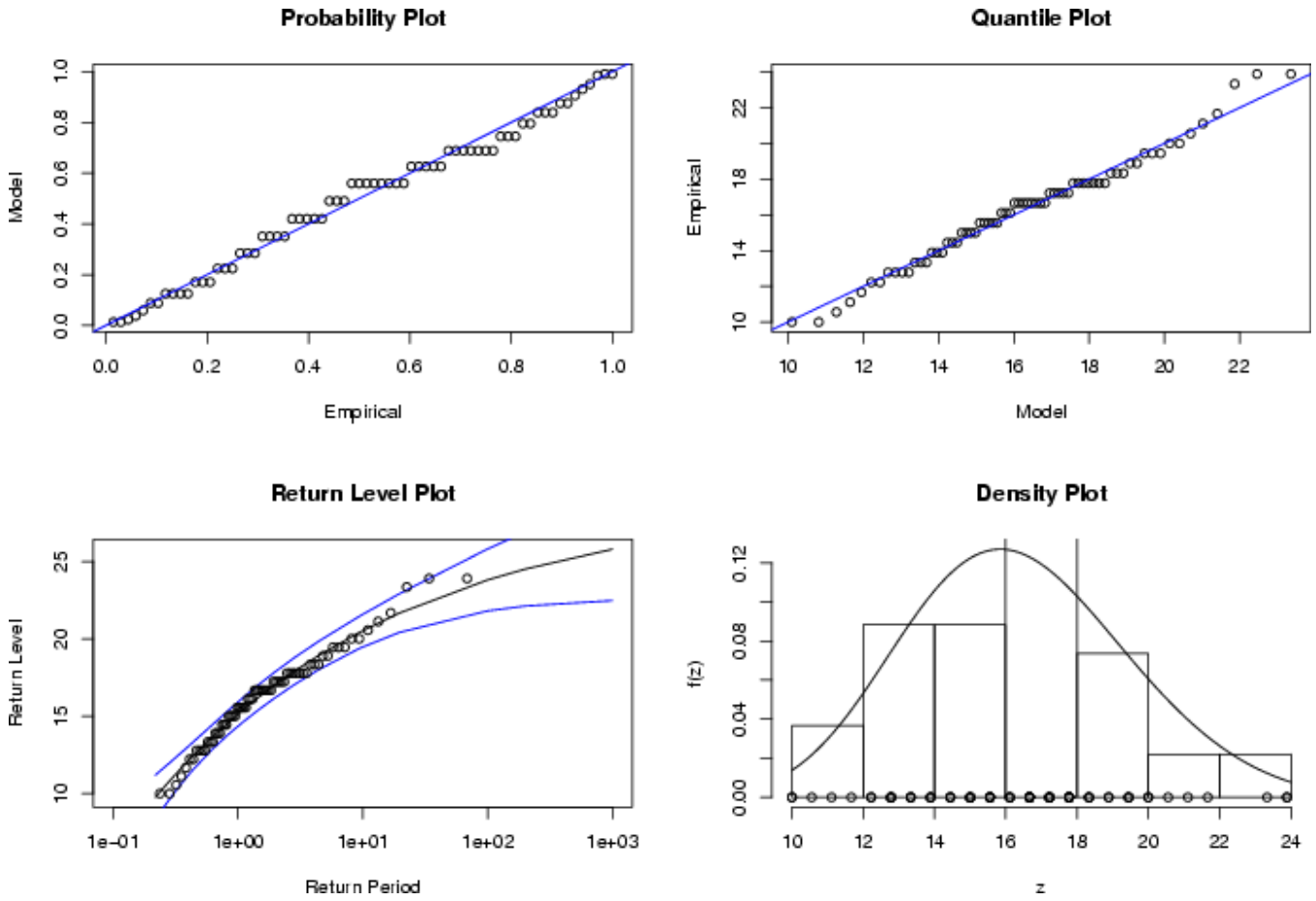


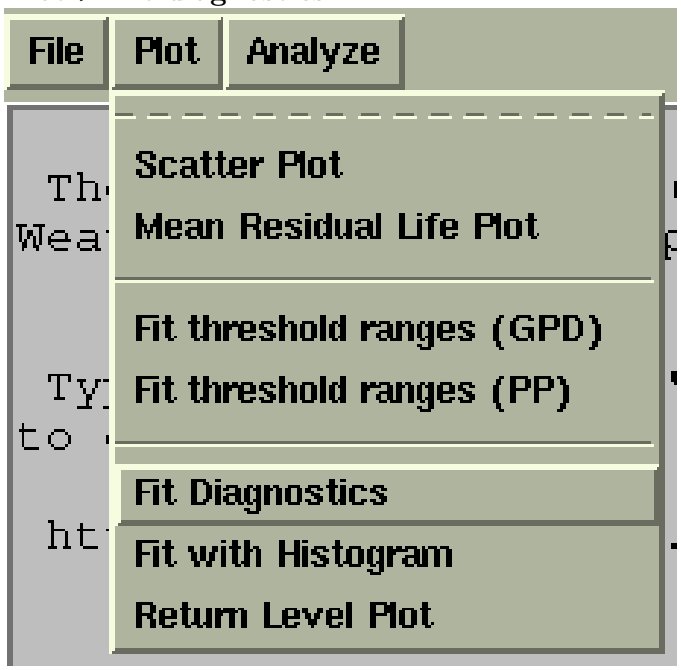
Figure 2.2: *GEV fit diagnostics for Port Jervis winter maximum temperature dataset. Quantile and return level plots are in degrees centigrade.*

Briefly, the quantile plot compares the model quantiles against the data (empirical) quantiles. A quantile plot that deviates greatly from a straight line suggests that the model assumptions may be invalid for the data plotted. The return level plot shows the return period against the return level, and shows an estimated 95% confidence interval. The return level is the level (in this case temperature) that is expected to be exceeded, 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 Fig. 2.2, 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] Ch. 3 for more details about these plots.

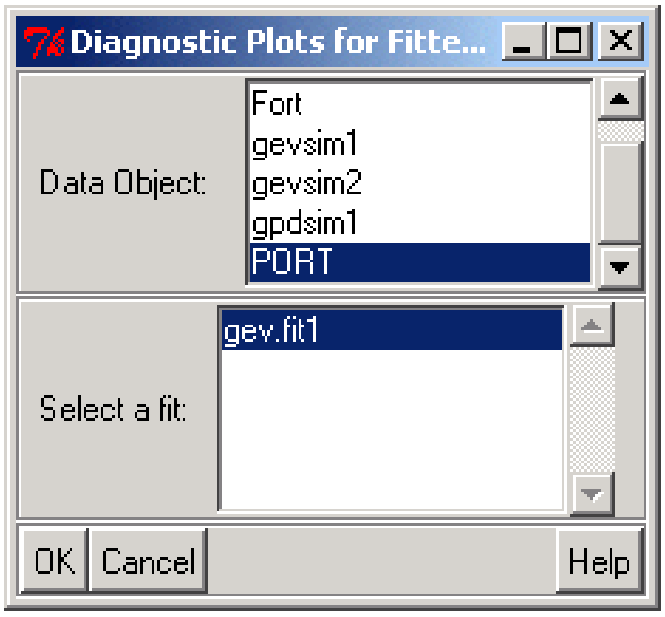
In the status section of the main window, several details of the fit are displayed. The maximum likelihood estimates of each of the parameters are given, along with their respective standard errors. In this case, $\hat{\mu} \approx 15.14$ degrees centigrade (0.39745 degrees), $\hat{\sigma} \approx 2.97$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

Note that Fig. 2.2 can be re-made in the following manner.

- **Plot > Fit diagnostics**



- *Select* **PORT** *from the* **Data Object** *listbox.*
- *Select* **gev.fit1** *from the* **Select a fit** *listbox* > **OK** > *GEV is fit and plot diagnostics displayed.*



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 Northern 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. Section 2.0.7 explores the inclusion of one of these variables as a covariate.

2.0.6 Return level and shape parameter (ξ) $(1-\alpha)$% confidence limits

Confidence intervals may be estimated using the toolkit for either the m-year return level or shape parameter (ξ) of either the GEV distribution or the GPD. The estimates are based on the profile likelihood method; finding the intersection between the respective profile likelihood values and $\frac{1}{2}c_{1,1-\alpha}$, where $c_{1,1-\alpha}$ is the distance between the maximum of the profile log-likelihood and the α quantile of a χ^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 using `extRemes` is as follows.

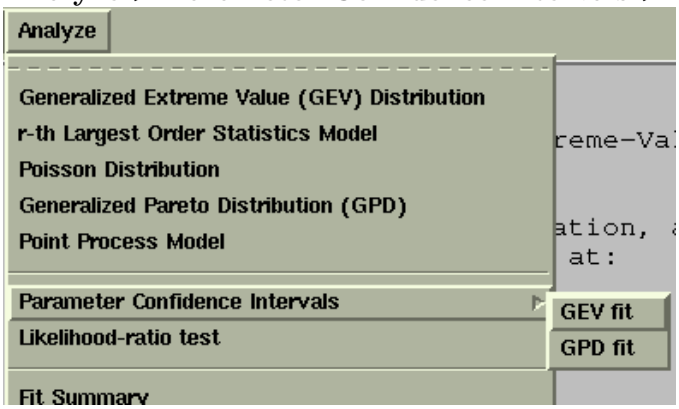
- **Analyze > Parameter Confidence Intervals > GEV fit**
- *Select an object from the* **Data Object** *listbox.*
- *Select a fit from the* **Select a fit** *listbox.*
- *Enter search limits for both return level and shape parameter (xi) (and any other options)* > **OK**

Example: Port Jervis Data Continued

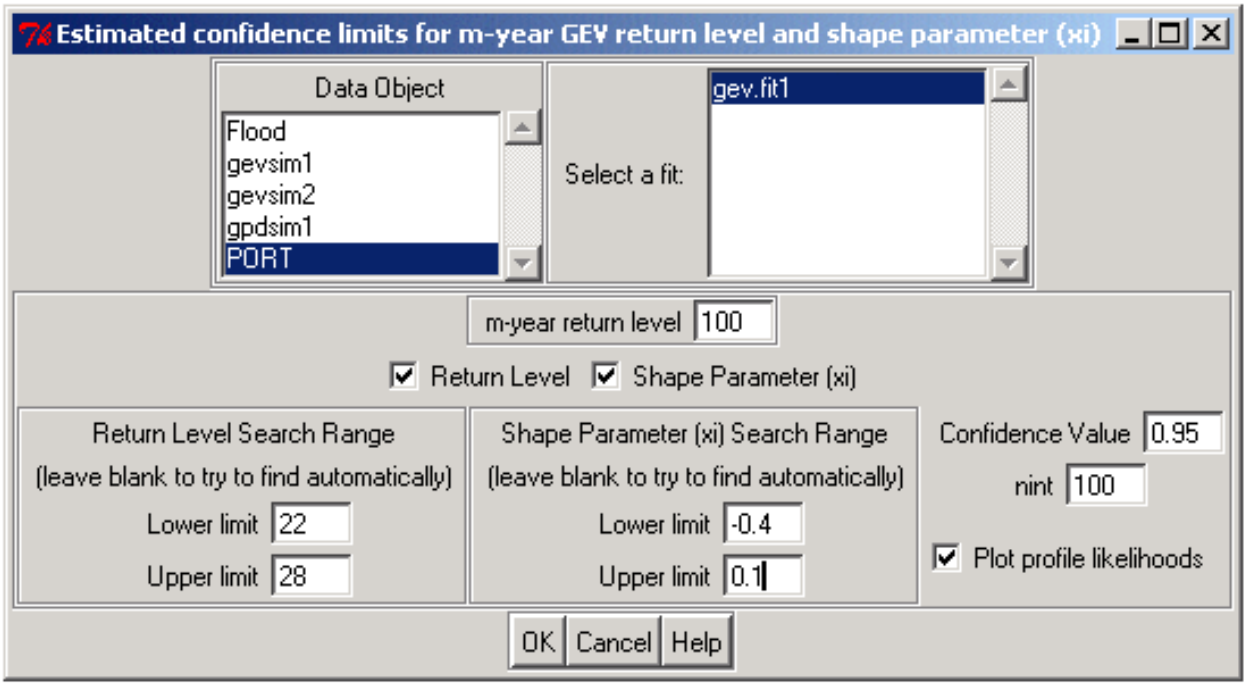
MLE estimate for 100-year return levels in the above GEV fit for the Port Jervis data are found to be somewhere between 20 and 25 degrees (using the return level plot), and $\hat{\xi} \approx -0.2$ (± 0.07). These values can be used in finding a reasonable search range for estimating the confidence limits. In the case of the return level one range that finds *correct*[5] confidence limits is from 22 to 28, and similarly, for the shape parameter, from -0.4 to 0.1. To find confidence limits, do the following.

[5] If the **Lower limit** (or **Upper limit**) field(s) is/are left blank, `extRemes` will make a reasonable guess for these values. Always check the **Plot profile likelihoods** checkbutton, and inspect the plots when finding limits automatically in order to ensure that the confidence intervals are correct or not. If they do not appear to be *correct* (i.e., if the dashed vertical line(s) does/do not intersect the profile likelihood at about where the lower horizontal line intersects the profile likelihood), the resulting plot might suggest appropriate limits to input manually.

- **Analyze > Parameter Confidence Intervals > GEV fit**



- *Select* **PORT** *from the* **Data Object** *listbox.*
- *Select* **gev.fit1** *from the* **Select a fit** *listbox.*
- *Enter* 22 *in the* **Lower limit** *of the* **Return Level Search Range** *and* 28 *in the* **Upper limit** *field.*[5]
- *Enter* -0.4 *in the* **Lower limit** *of the* **Shape Parameter (xi) Search Range** *and* 0.1 *in the* **Upper limit** *field* > **OK**.[5]



Estimated confidence limits should now appear in the main toolkit dialog. In this case, the estimates are given to be about 22.42 to 27.18 degrees for the 100-year return level and about -0.35 to -0.05 for $\hat{\xi}$ indicating that this parameter is significantly below zero (i.e.,

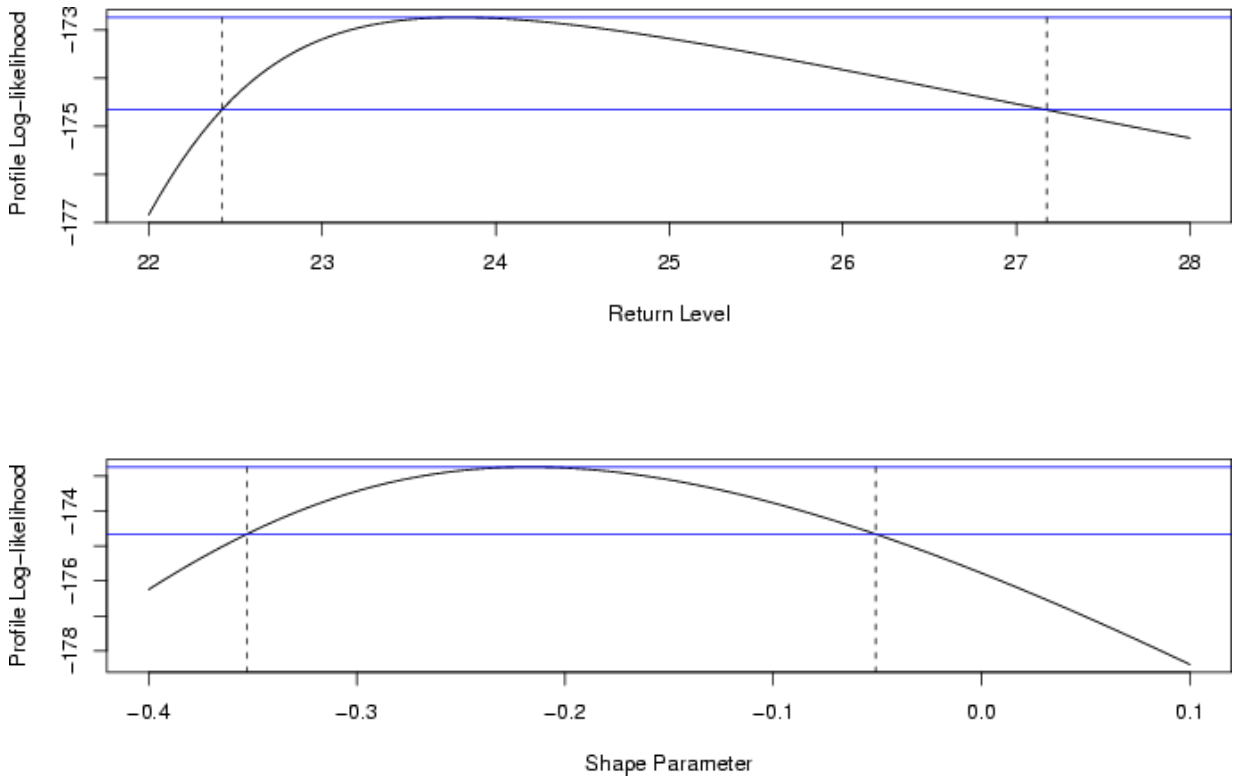


Figure 2.3: *Profile likelihood plots for the 100-year return level (degrees centigrade) and shape parameter (ξ) of the GEV distribution fit to the Port Jervis dataset.*

Weibull type). 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 likelihoods (Fig. 2.3) can be produced by clicking on the check checkbutton for this feature. In this case, our estimates are good because the dashed vertical lines intersect the likelihood at the same point as the lower horizontal line in both cases.

2.0.7 Fitting data to a GEV distribution with a covariate

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

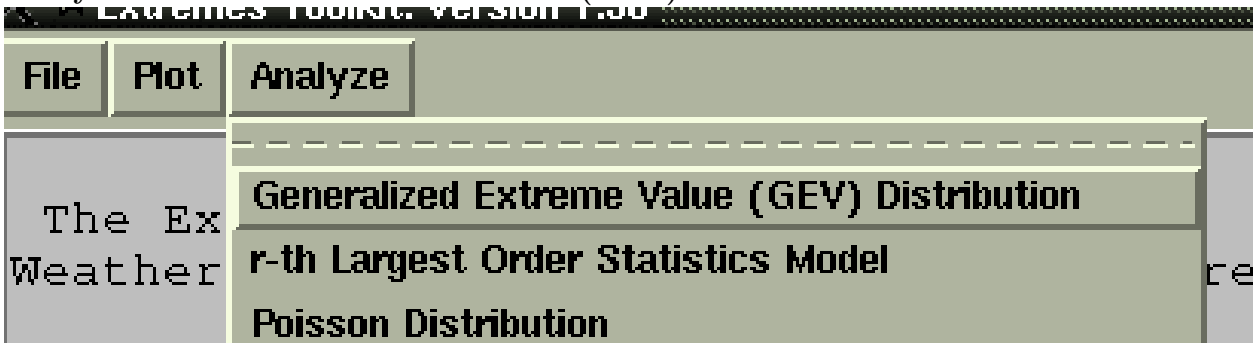
- **Analyze** > **Generalized Extreme Value (GEV) Distribution** > *New window appears*
- *Select data object from* **Data Object** *listbox. Column names appear in other listboxes.*

- *Choose a response variable from the* **Response** *listbox. Response variable is removed as an option from other listboxes.*
- *Select covariate variable(s) from* **Location parameter (mu), Scale parameter (sigma)** *and/or* **Shape parameter (xi)** *listboxes*
- *select which link function to use for each of these choices* > **OK**
- A GEV distribution will be fitted to the chosen response variable and stored in the same list object as the data used.

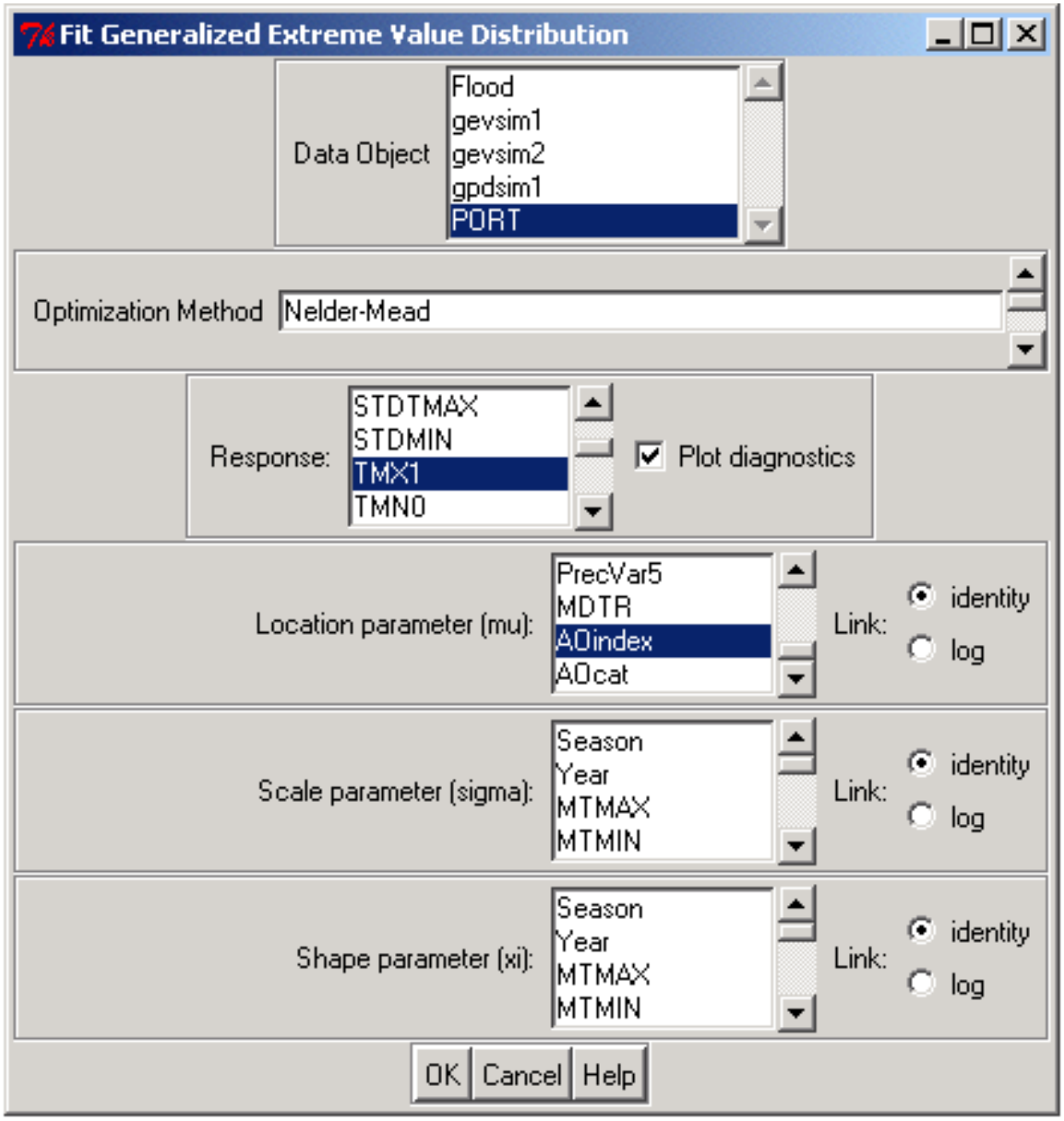
Example 2: Port Jervis data with a covariate

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 [18] for more information on this index.

Analyze > **Generalized Extreme Value (GEV) Distribution**.



- *Select* **PORT** *from* **Data Object** *listbox. Variables now listed in some other listboxes.*
- *Select* **TMX1** *from the* **Response** *listbox.* **TMX1** *removed from other listboxes.*
- *Optionally check the* **Plot diagnostics** *checkbox*
- *Select* **AOindex** *from* **Location parameter (mu)** *list* (*keep* **Link** *as* **identity**) > **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,

$$\hat{\mu} \approx 15.25 + 1.15 \cdot \textbf{AOindex}$$

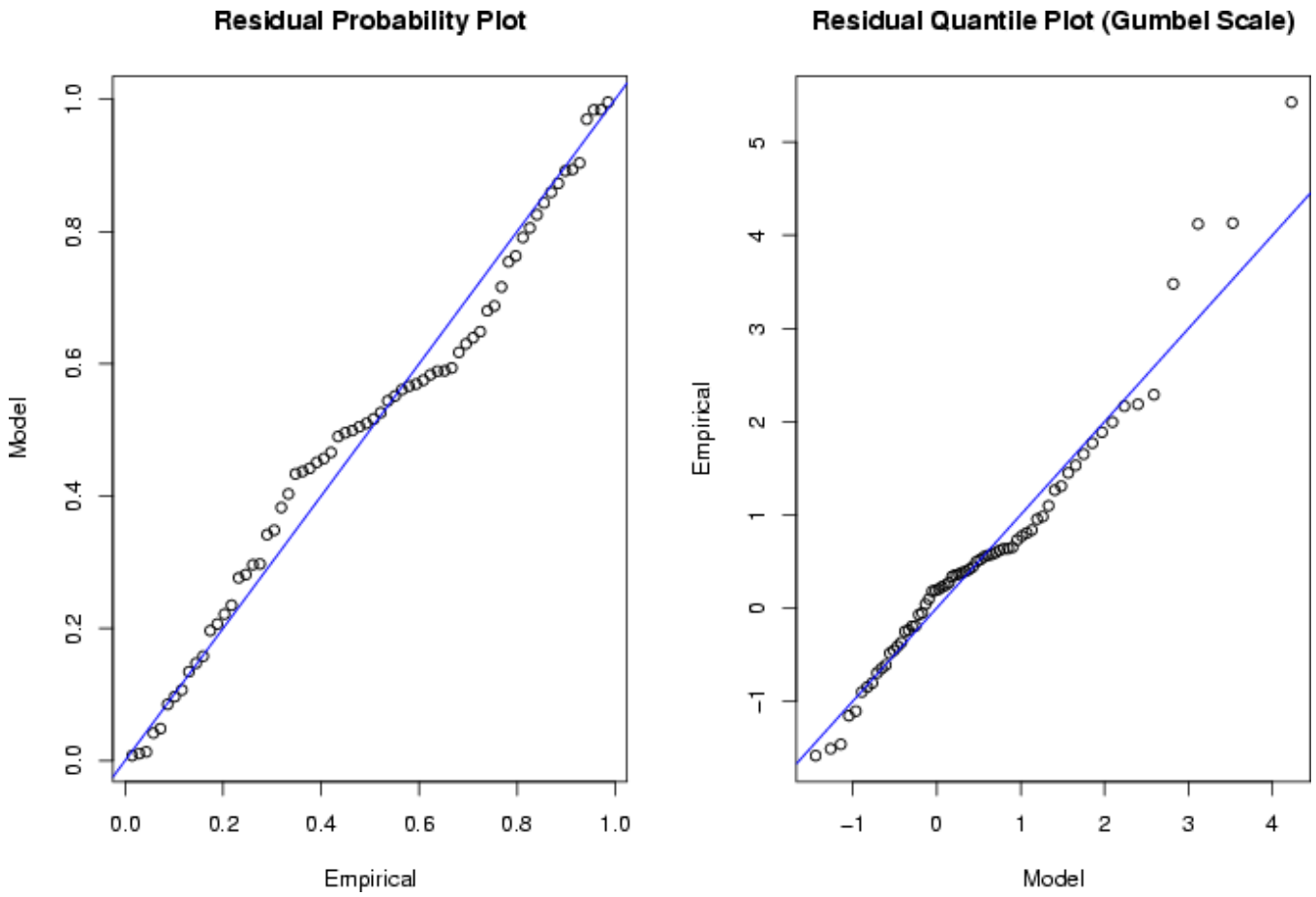


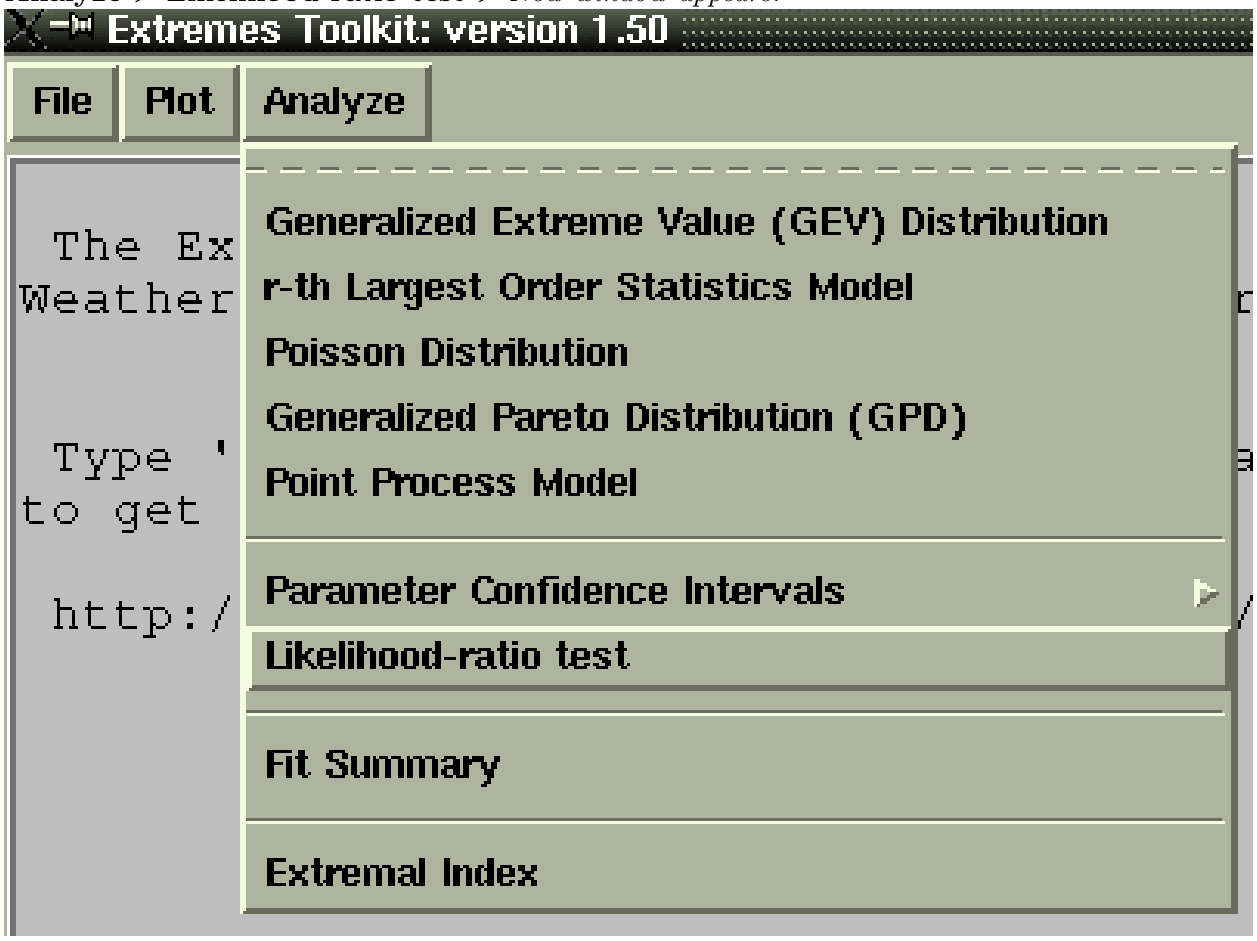
Figure 2.4: *GEV fit diagnostics for Port Jervis winter maximum temperature dataset with* **AOindex** *as a covariate. Both plots are generated using transformed variables and therefore the units are not readily interpretable. See appendix section C.0.29 for more details.*

Fig. 2.4 shows the diagnostic plots for this fit. Note that only the probability and quantile plots are displayed and that the quantile plot is in the Gumbel scale. See the appendix section C.0.29 for more details.

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. Specifically, the test compares the likelihood-ratio, $2 \cdot \log(\frac{l_1}{l_0})$, where l_0 and l_1 are the likelihoods for each of the two models (l_0 must be nested in l_1), to a χ^2_ν quantile, where ν 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 $\mu = b_0 + b_1 \cdot \mathbf{AOindex}$ giving us the new parameters: b_0, b_1, σ and ξ. So, for this example, $\nu = 4 - 3 = 1$. See

Coles [3] section 6.2 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:

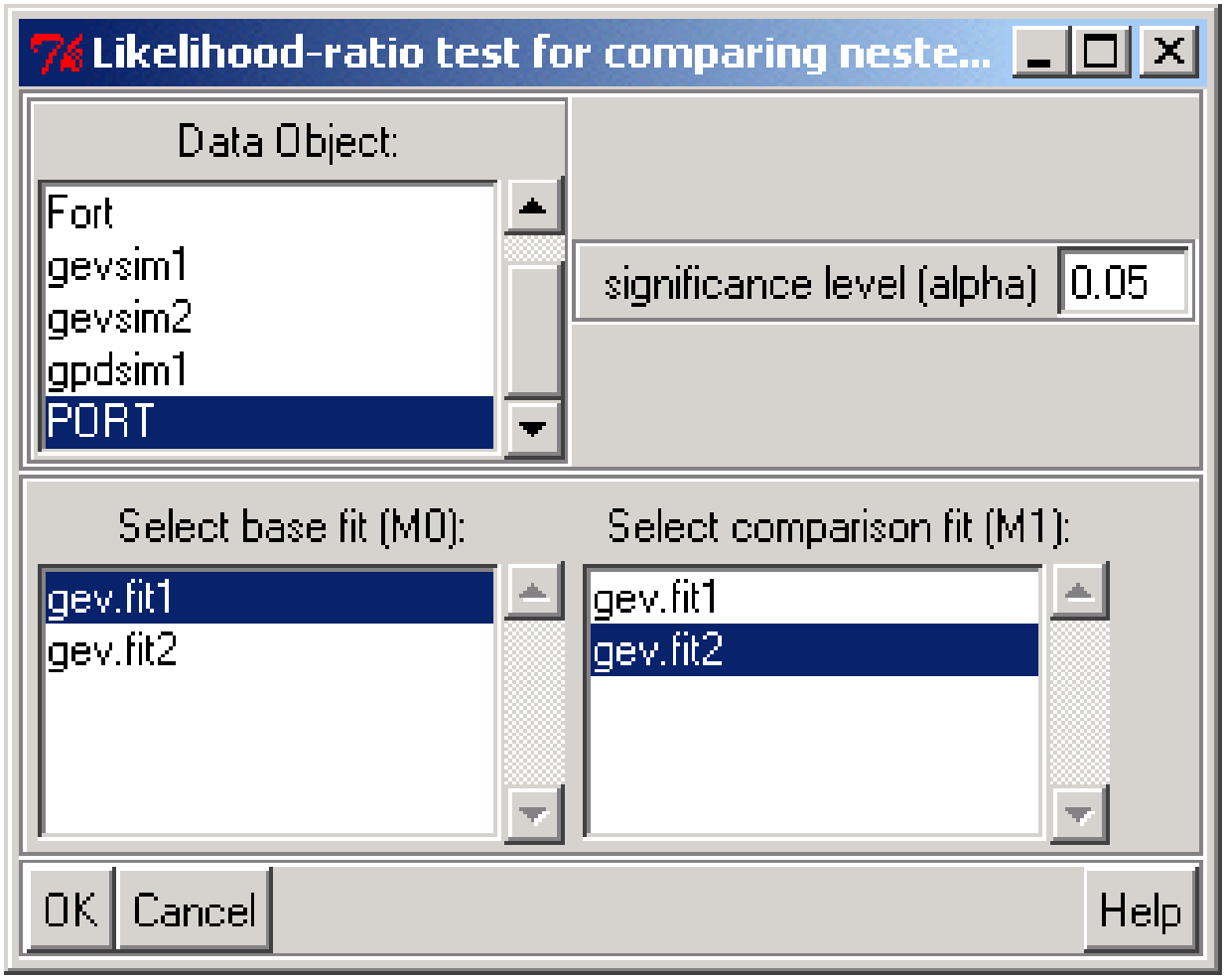
- **Analyze** > **Likelihood-ratio test** > *New window appears.*



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

- *Select fits to compare. In this case,* **gev.fit1** *from* **Select base fit (M0)** [6] *listbox and* **gev.fit2** *from* **Select comparison fit (M1)** [6] *listbox* > **OK**.

[6] If fit from **M0** has more components than that of **M1**, `extRemes` will assume **M1** is nested in **M0**, and computes the likelihood-ratio accordingly.



- 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 χ^2_1 distribution of 3.8415, suggesting that the covariate **AOindex** model is a significant improvement over the model without a covariate. The small p-value of 0.000565 further supports this claim.

In addition to specifying the covariate for a given parameter, the user has the ability to indicate what type of link function should relate that covariate to the parameter. The two available link functions (*identity* and *log*) are indicated by the radiobuttons to the right of the covariate list boxes. This example used the *identity* link function (note that the *log* link is labeled *exponential* in Stuart Coles' software (`ismev`)). For example, to model the scale parameter (σ) with the log-link and one covariate, say x, gives $\sigma = \exp(\beta_0 + \beta_1 x)$ or $\ln \sigma = \beta_0 + \beta_1 x$.

Chapter 3

Frequency of Extremes

Often it is of interest to look at the frequency of extreme event occurences. As the event becomes more rare, the occurence of events approaches a Poisson process, so that the relative frequency of event occurence approaches a Poisson distribution. See appendix section B.0.26 for more details.

3.0.8 Fitting data to a Poisson distribution

The Extremes Toolkit also provides for fitting data to the Poisson distribution, although not in the detail available for the GEV distribution. The Poisson distribution is also useful for data that involves random sums of rare events. For example, a dataset containing the numbers of hurricanes per year and total monetary damage is included with this toolkit named **Rsum.R**.

Analyze > **Poisson Distribution**.

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, λ, is currently estimated; in this case, the MLE for λ is simply the mean of the data. If a covariate is given, the generalized linear model fit is used from the R[14] function `glm` (see the help file for `glm` for more information). Currently, `extRemes` provides only for fitting data to Poissons with the "log" link function.
EXAMPLE: HURRICANE COUNT DATA

Load the Extremes Toolkit dataset **Rsum.R** as per section 1.2.1 and save it (in R) as **Rsum**. That is,

- **File** > **Read Data**
- *Browse for* **Rsum.R** *(in* `extRemes` *data folder)* > **OK**

- *Check* **R source** *radiobutton* > *Type* **Rsum** *in* **Save As (in R)** *field.* > **OK**

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 [13] or Katz [7]. A simple fit without a trend in the data is performed in the following way.

- **Analyze** > **Poisson Distribution** > *New window appears.*
- *Select* **Rsum** *from* **Data Object** *listbox.*
- *Select* **Ct** *from* **Response** *listbox* > **OK**.
- MLE for rate parameter (lambda) along with the variance and χ^2 test for equality of the mean and variance is displayed in the main toolkit window.

For these data $\hat{\lambda} \approx 1.817$, 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, λ. As per Katz [7], the estimated variance is shown to be 1.752, which is only slightly less than that of the mean (1.817). The χ^2_{70} statistic 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 2.0.5, 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.

3.0.9 Fitting data to a Poisson distribution with a covariate

The procedure for fitting data to a Poisson with a trend (using the **Rsum** dataset from section 3.0.8 with ENSO state as a covariate) is as follows.

- **Analyze** > **Poisson Distribution** > *New window appears.*
- **Select Rsum** *from* **Data Object** *listbox.*
- *Select* **Ct** *from* **Response** *listbox.*
- *Select* **EN** *from* **Trend variable** *listbox* > **OK**.
- Fitted rate coefficients and other information are displayed in main toolkit window.

EN for this dataset represents the ENSO state (i.e., **EN** is -1 for La Niña events, 1 for for El Niño events, and 0 otherwise). A plot of the residuals is created if the **plot diagnostics** checkbutton is engaged. The fitted model is found to be:

$$\log(\hat{\lambda}) = 0.575 - 0.25 \cdot \mathbf{EN}$$

For fitting a Poisson regression model to data, a likelihood-ratio statistic is given in the main toolkit dialog, where the ratio is the null model (of no trend in the data) to the model with a trend (in this case, ENSO). Here the addition of ENSO as a covariate is significant at the 5% level (p-value ≈ 0.03) indicating that the inclusion of the ENSO term as a covariate is reasonable.

Chapter 4

r-th Largest Order Statistic Model

It is also possible to extend the block maxima methods to other order statistics. The simplest case is to look at minima, where one needs only take the negative of the data and then use the regular maximum methods (see, for example, section 6.0.16 Example 3). It is also possible to model other order statistics more generally. One such method is referred to as the r-th largest order statistic model. This model has essentially been replaced by the threshold exceedance methods (see chapters 5 and 6) in practice, but `extRemes` 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].

Although limited in scope, it is possible to perform an r-th largest order statistics model fit using `extRemes`. The (*common* format) dataset **Ozone4H.dat** is included in the **data** directory. Data for fitting this model must be in a much different form than data used for all the other model fits with `extRemes`. Instead of one response column, there needs to be as many columns as r. That is, if interest is in the fourth highest value, then there must be at least four columns of data giving the maxima, second-, third- and fourth-highest values, respectively; missing values are allowed. In the case of **Ozone4H.dat**, there are five columns: the first (**obs**) is simply an index from 1 to 513, the second (**r1**) are maxima, followed by **r2**, **r3** and **r4**. Here, all of the data come from 1997, but from 513 different monitoring stations in the eastern United States. The order statistics represent the maximum, second-, third- and fourth-highest daily maximum 8-hour average ozone for 1997 (see Fuentes [5] or Gilleland and Nychka [6] for more about these data). After loading **Ozone4H.dat**, saved in R as **Ozone4H**, the r-th largest order statistic model can be applied in the following manner.

- **Analyze** > **r-th Largest Order Statistics Model**
- *Select* **Ozone4H** *from the* **Data Object** *listbox.*
- *Select* **r1**, **r2**, **r3** and **r4** *from the* **Response** *listbox.*

- *Check the* **Plot diagnostics** *checkbutton (if desired)*[7] > **OK**.

[7]Multiple panels of plots will be plotted. The user must hit return at the R session window to view each plot. This may interrupt seeing fit results until all plots are viewed. See Coles [3] for an explanation of these plots.

Chapter 5

Generalized Pareto Distribution (GPD)

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. `extRemes` provides for fitting data to GPD models as well as some tools for threshold selection. For more information on the GPD see appendix section B.0.25.

5.0.10 Fitting Data to a GPD

The general procedure for fitting data to a GPD using `extRemes` is:

- **Analyze** > **Generalized Pareto Distribution (GPD)** > *New window appears*
- *Select a data object from* **Data Object** *listbox. Covariates appear in various listboxes.*
- *Select a response variable from* **Response** *listbox. Selected response is removed from other listboxes.*
- *Enter a threshold (only values above this threshold will be fitted to the GPD)* > *other options* > **OK**
- A GPD will be fitted and results will appear in the main toolkit window.

Example 1: Hurricane damage

For this example, load the `extRemes` dataset, **damage.R** and save it (in R) as **damage**. That is,

- **File** > **Read Data**

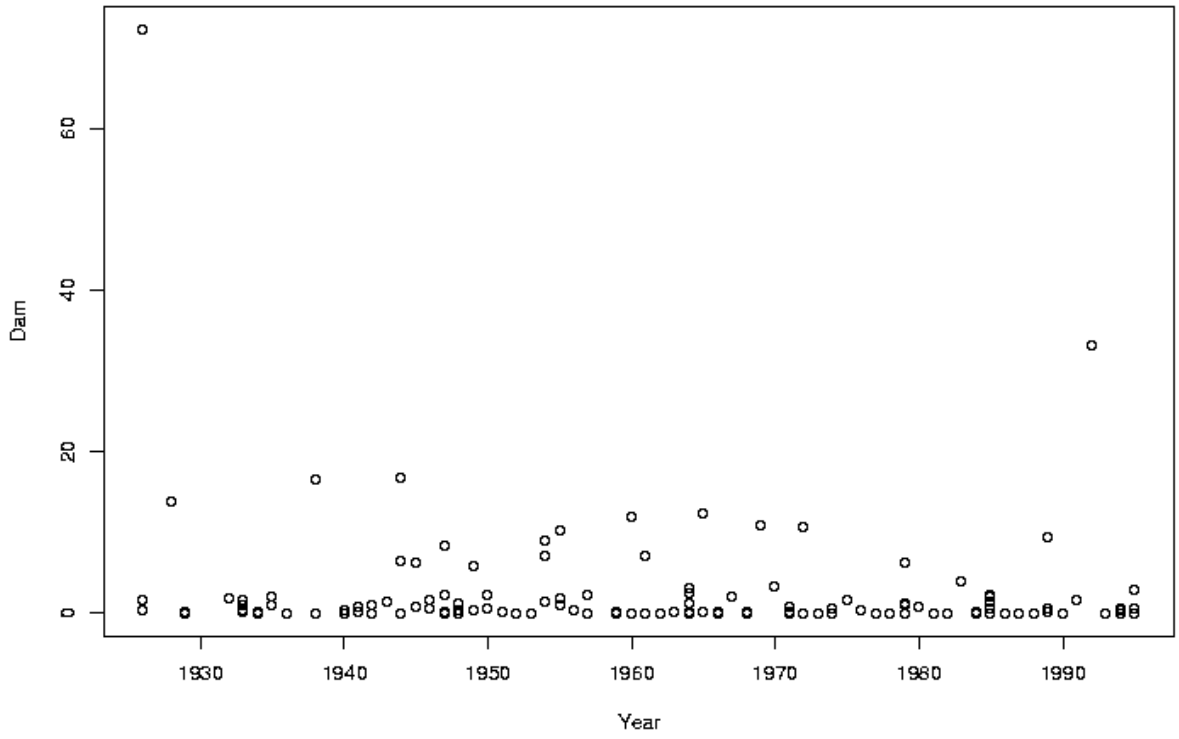
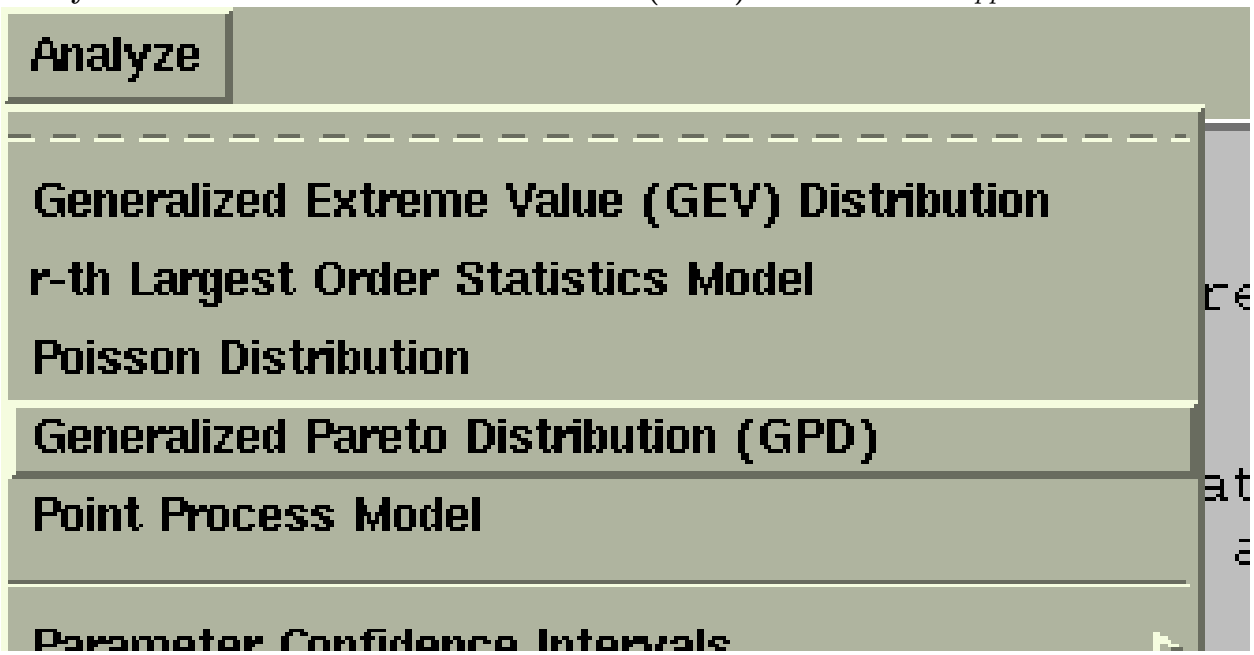


Figure 5.1: *Scatter plot of U.S. hurricane damage (in billions $ U.S.).*

- *Browse for* **damage.R** *in* `extRemes` *library data folder* $>$ **OK**
- *Check the* **R source** *radiobutton.*
- *Type* **damage** *in the* **Save As (in R)** *field* $>$ **OK**

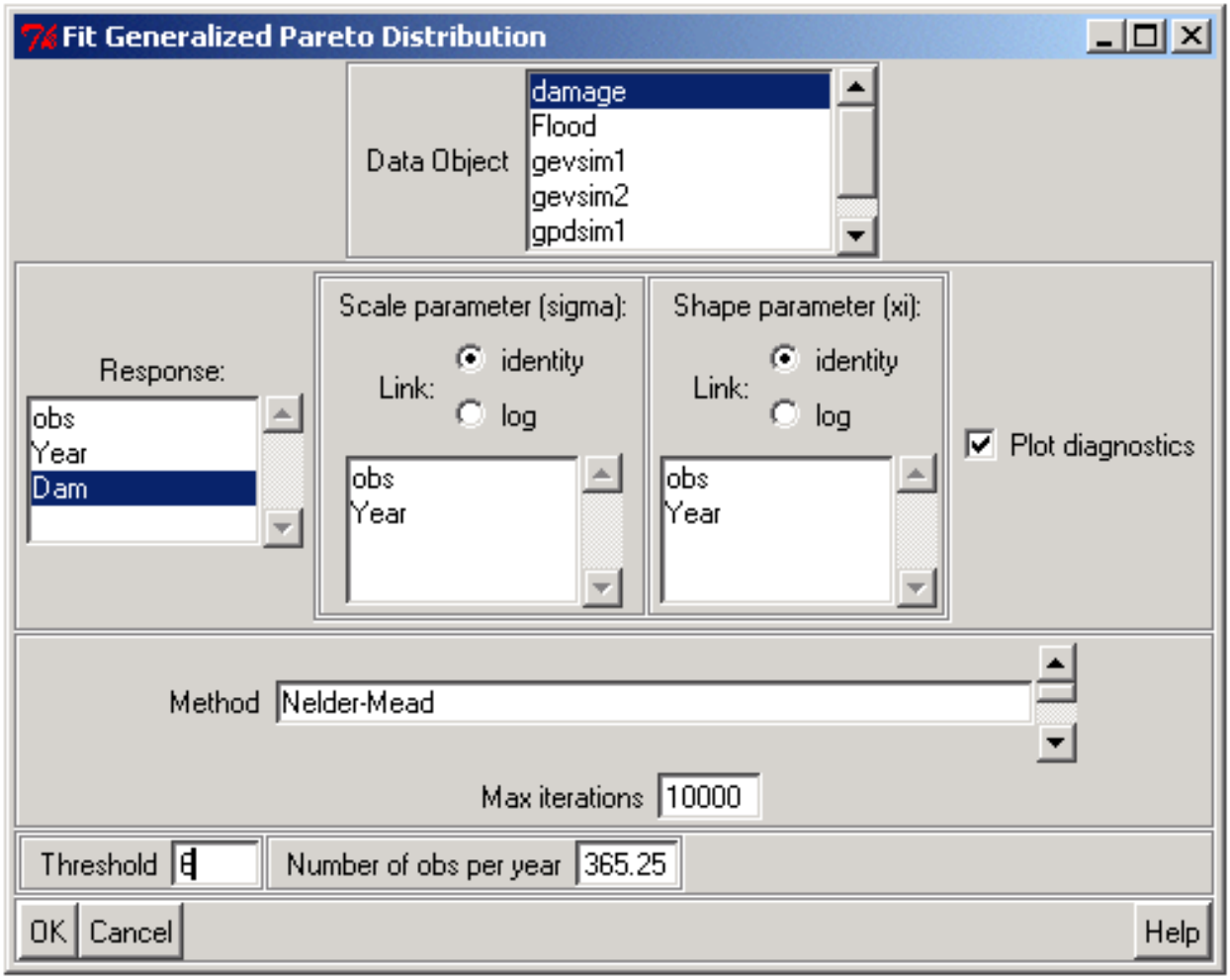
Fig. 5.1 shows the scatter plot of these data from 1925 to 1995. The data are economic damage of individual hurricanes in billions of U.S. dollars. These data correspond to the count data discussed in section 3.0.8. To learn more about these data, please see Pielke and Landsea [13] or Katz [7]. The time series shows that there was a particularly large assessment of economic damage early on (in 1926) of over 70 billion dollars. After this time, assessments are much smaller than this value.

- **Analyze** > **Generalized Pareto Distribution (GPD)** > *New window appears*



- *Select* **damage** *from the* **Data Object** *listbox. Covariates appear in various listboxes.*
- *Select* **Dam** *from* **Response** *listbox. Selected response is removed from other listboxes.*
- *Enter* **6** *in the* **Threshold** *field.*

- *optionally check* **Plot diagnostics** > **OK**

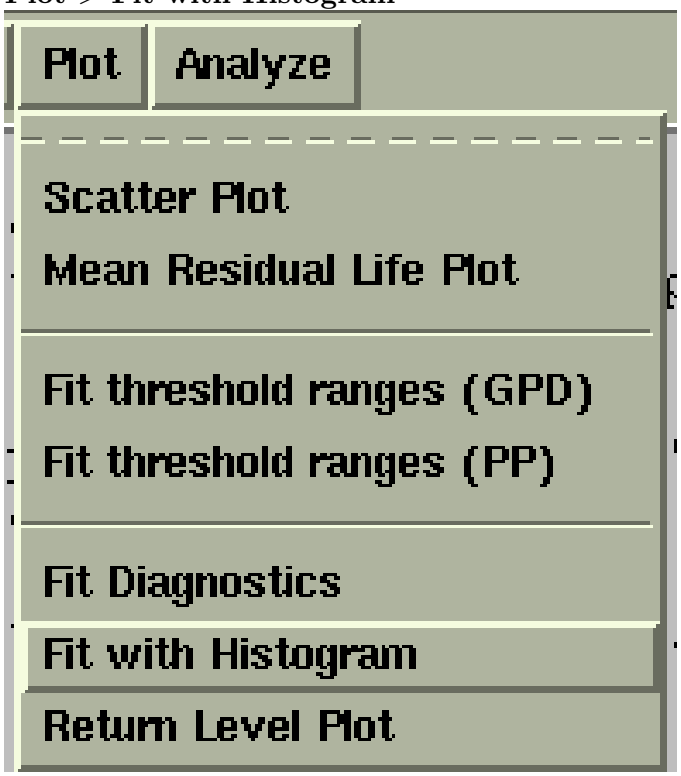


- A GPD will be fitted and results will appear in main toolkit window.
- Note that the **Number of obs per year** is not relevant for this type of dataset.

Diagnostic plots for the GPD fit for these data with economic damage, **Dam**, as the response variable and a threshold of 6 billion dollars are shown in Fig. 5.2. The fit looks pretty good considering the one rather large outlier from 1926 and only 18 values over the threshold.

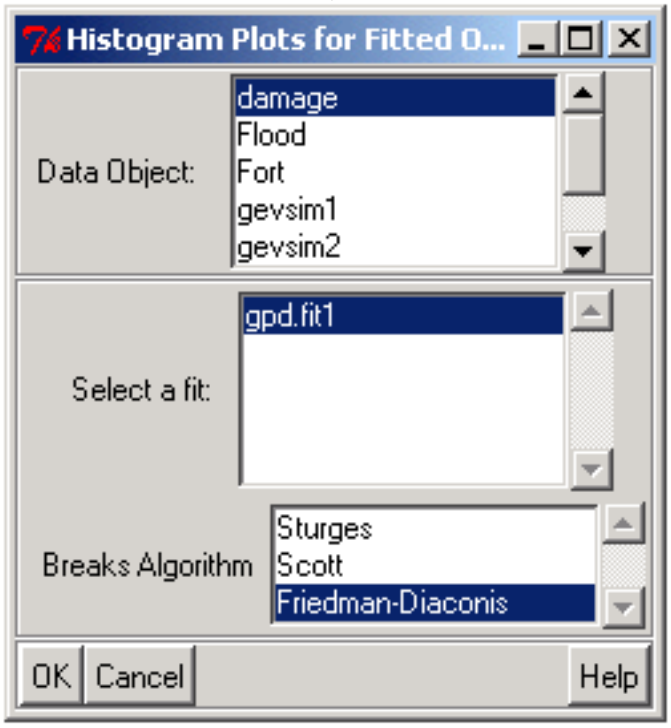
The histogram in Fig. 5.2 *appears* to include all of the data, and not just data above the threshold. However, this is simply a result of the binning algorithm used; in this case the default **Sturges** algorithm. The same histogram can be plotted, with this or a choice of two other algorithms: **Scott** or **Friedman-Diaconis** in the following manner.

- **Plot > Fit with Histogram**



- *Select* **damage** *from the* **Response** *listbox.*
- *Select* **gpd.fit1** *from the* **Select a fit** *listbox.*

- *Select a breaks algorithm (here Friedman-Diaconis is selected) and click* **OK**.



The histogram shown in Fig. 5.3 used the Friedman-Diaconis algorithm. Each choice of breaks algorithm is simply a different algorithm for binning the data for the histogram. The histogram of Fig. 5.3 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 Fig. 5.4 looks better. It is currently not possible to produce this histogram directly from `extRemes`. This histogram was produced in the following manner. From the R prompt:

```
> max( damage$models$gpd.fit1$dat)
[1] 72.303
> brks <- seq(6, 72.303, ,15)
> hist( damage$models$gpd.fit1, breaks=brks)
```

See the help file for the R function `hist` for more details about plotting histograms in R. That is, from the R prompt type:

```
> help( hist)
```

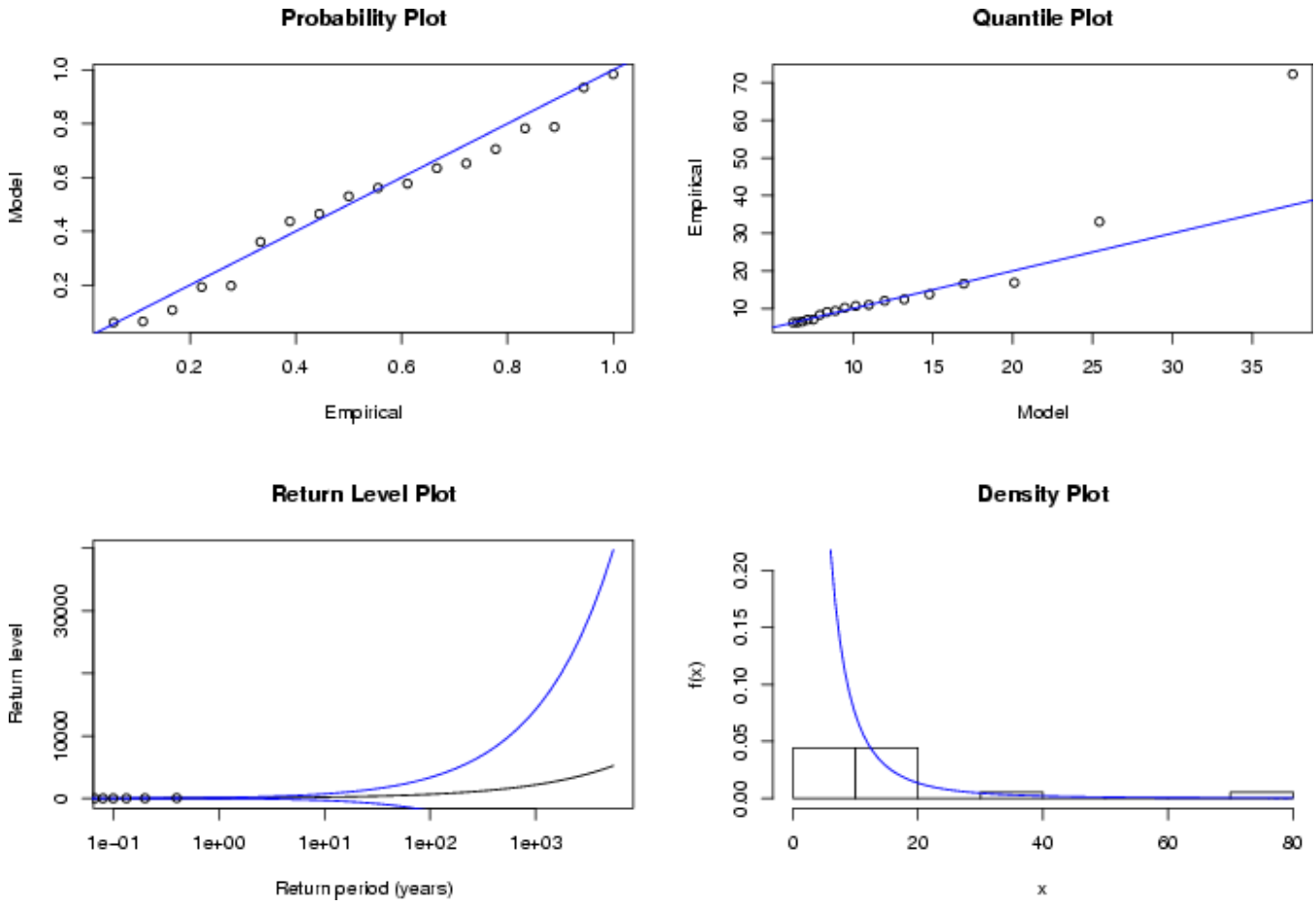


Figure 5.2: *GPD fit for hurricane damage data using a threshold of 6 billion dollars.*

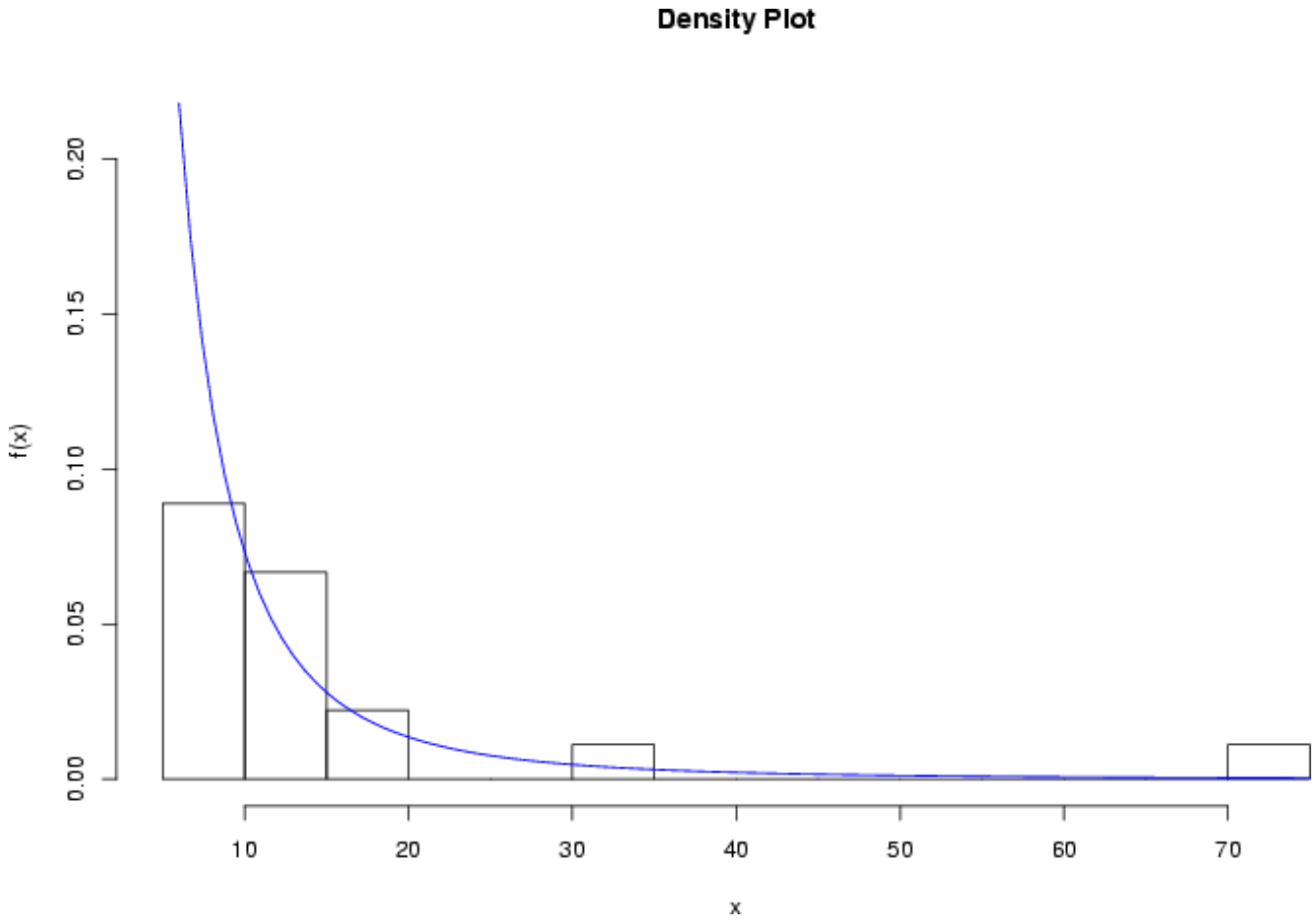


Figure 5.3: *Histogram for GPD fit for hurricane damage data using a threshold of 6 billion dollars and the Friedman-Diaconis algorithm for bin breaks.*

For these data, $\hat{\sigma} \approx 4.6$ billion dollars (1.82 billion dollars) and $\hat{\xi} \approx 0.5$ (0.340). 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.R** 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 relatively large dataset.

- **File** > **Read Data** > *New window appears*
- *Browse to* `extRemes` *data directory and select* **FtCoPrec.R** *New window appears*
- *Select* **common** *from the* **Data Type** *field* >

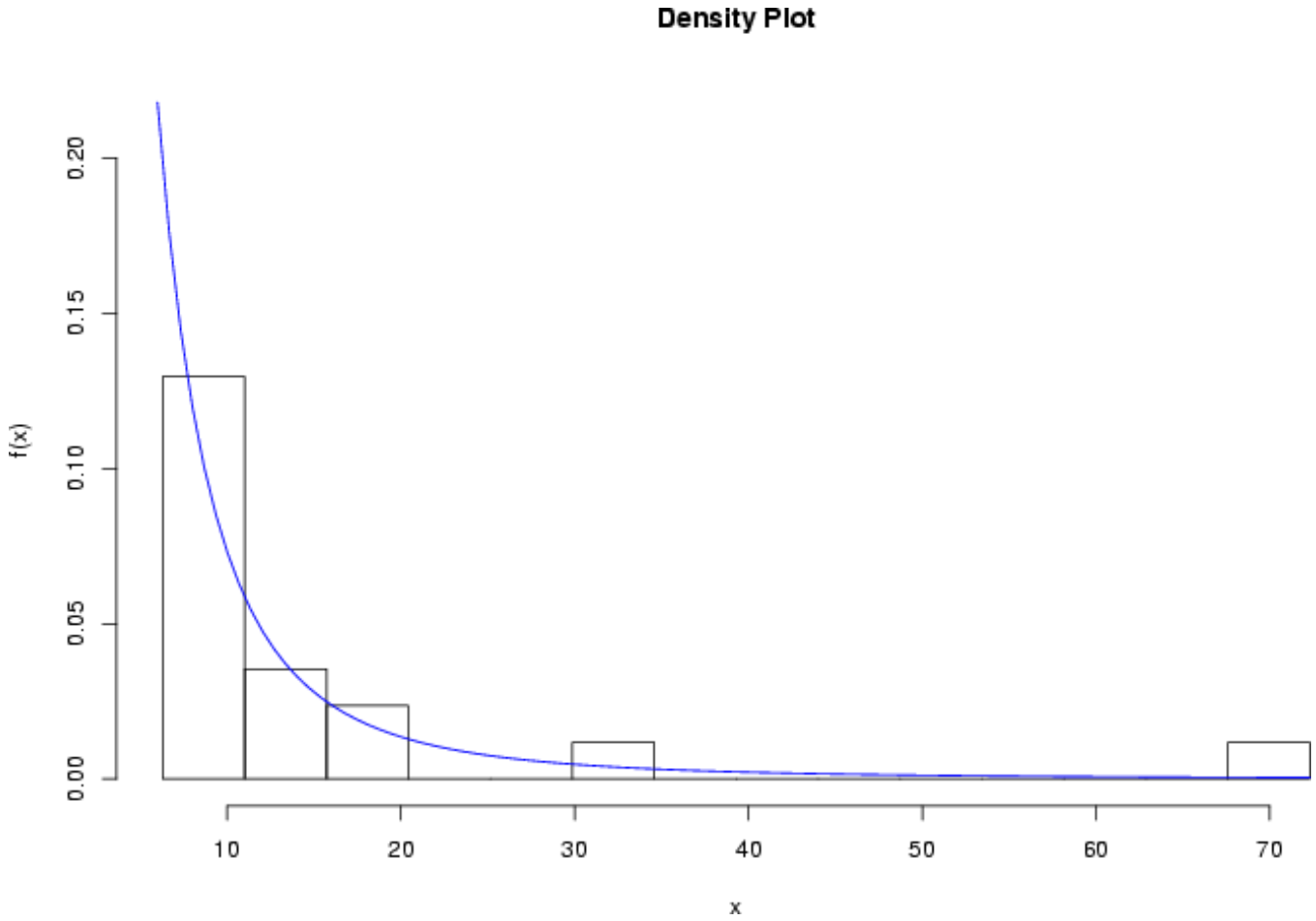


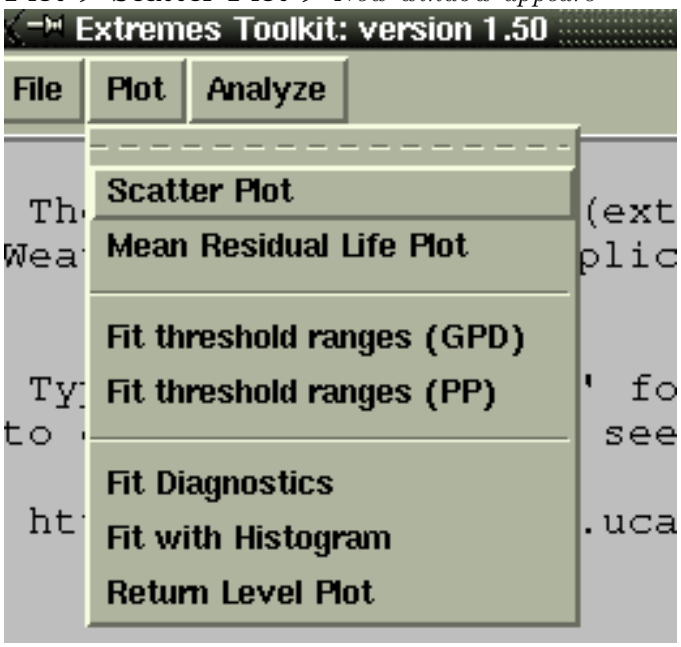
Figure 5.4: *Histogram for GPD fit for hurricane damage data using a threshold of 6 billion dollars and a specialized vector for the breaks. See text for more details.*

- *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**.

This dataset has precipitation data for a single location in Fort Collins, C.O., USA for the time period 1900-1999. These data are of special interest because of a flood that occurred there on July 28, 1997. See Katz *et al.* [9] for more information on these data.

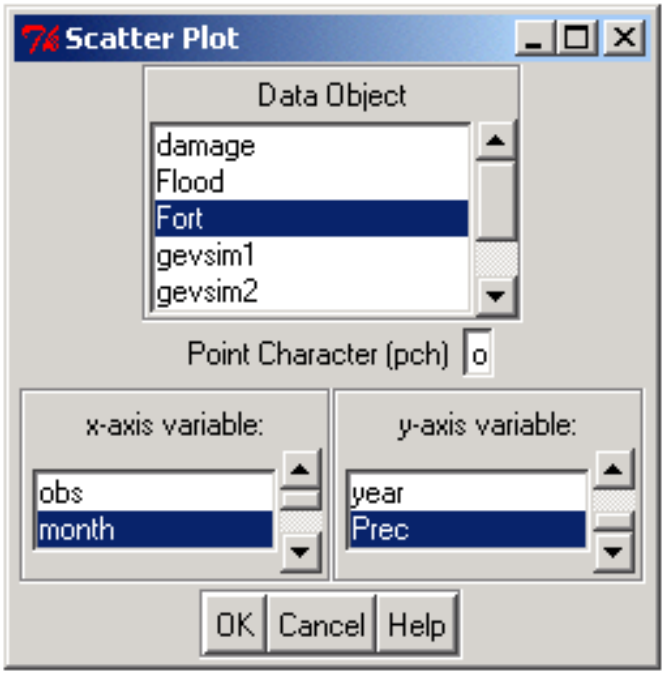
Fig. 5.5 shows a scatter plot of the daily precipitation (by month) at this location. Using `extRemes`:

- **Plot** > **Scatter Plot** > *New window appears*



- *Select* **Fort** *from* **Data Object** *listbox. Covariates appear in other listboxes.*

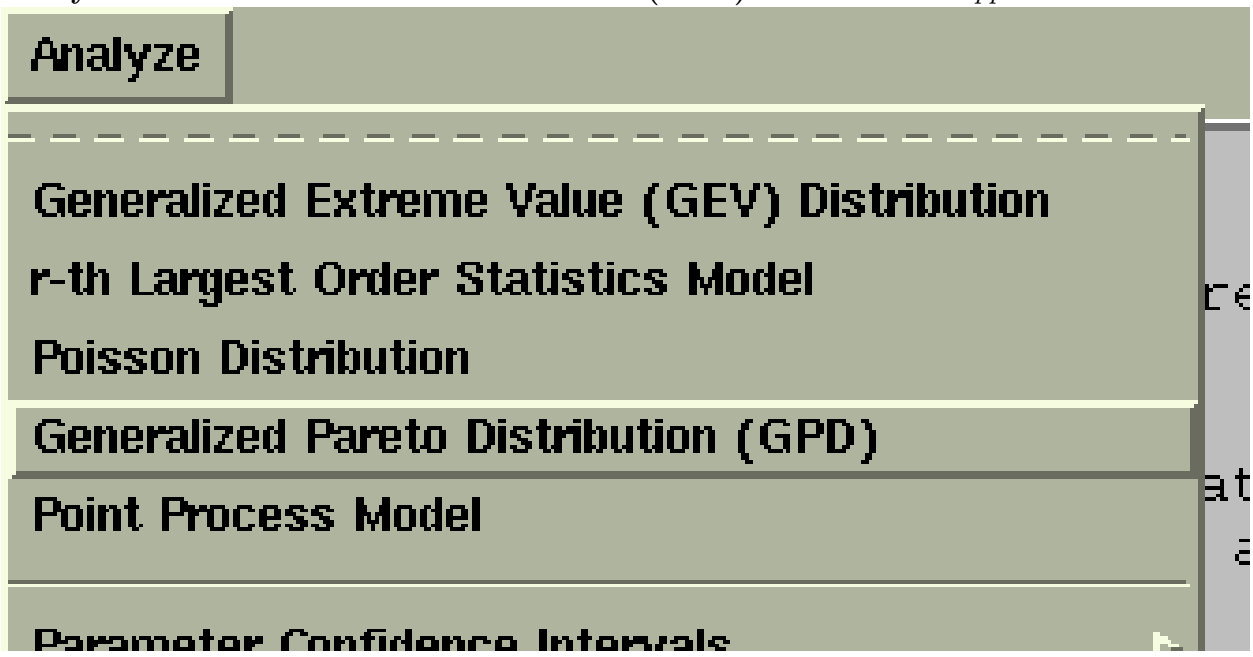
- *Select* **month** *from* **x-axis** *listbox and* **Prec** *from* **y-axis** *listbox* > **OK**



- Plot in Fig. 5.5 should appear.

To fit a GPD model using the toolkit do the following.

- **Analyze** > **Generalized Pareto Distribution (GPD)** > *New window appears*



- *Select* **Fort** *from* **Data Object** *listbox. Covariates appear in other listboxes.*

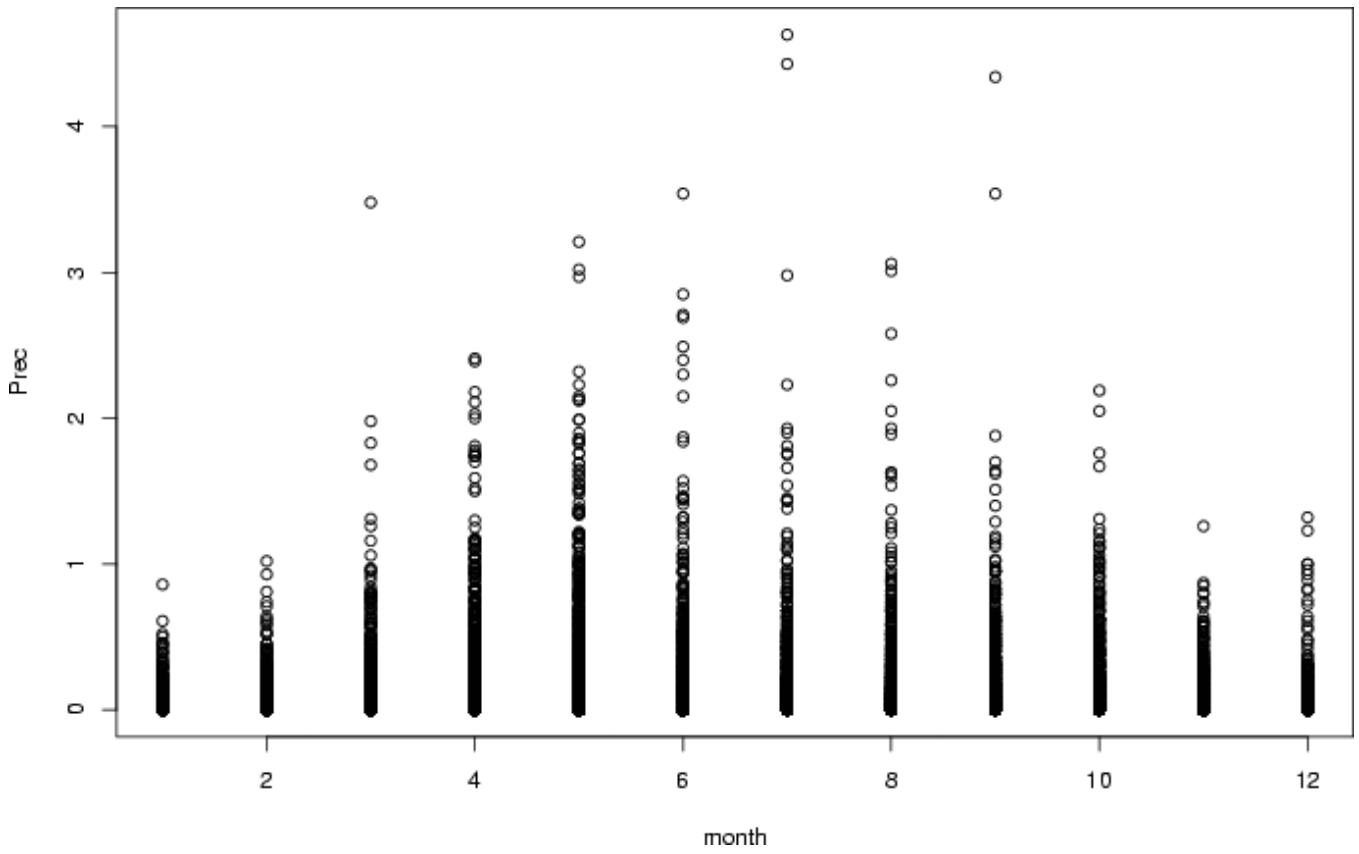
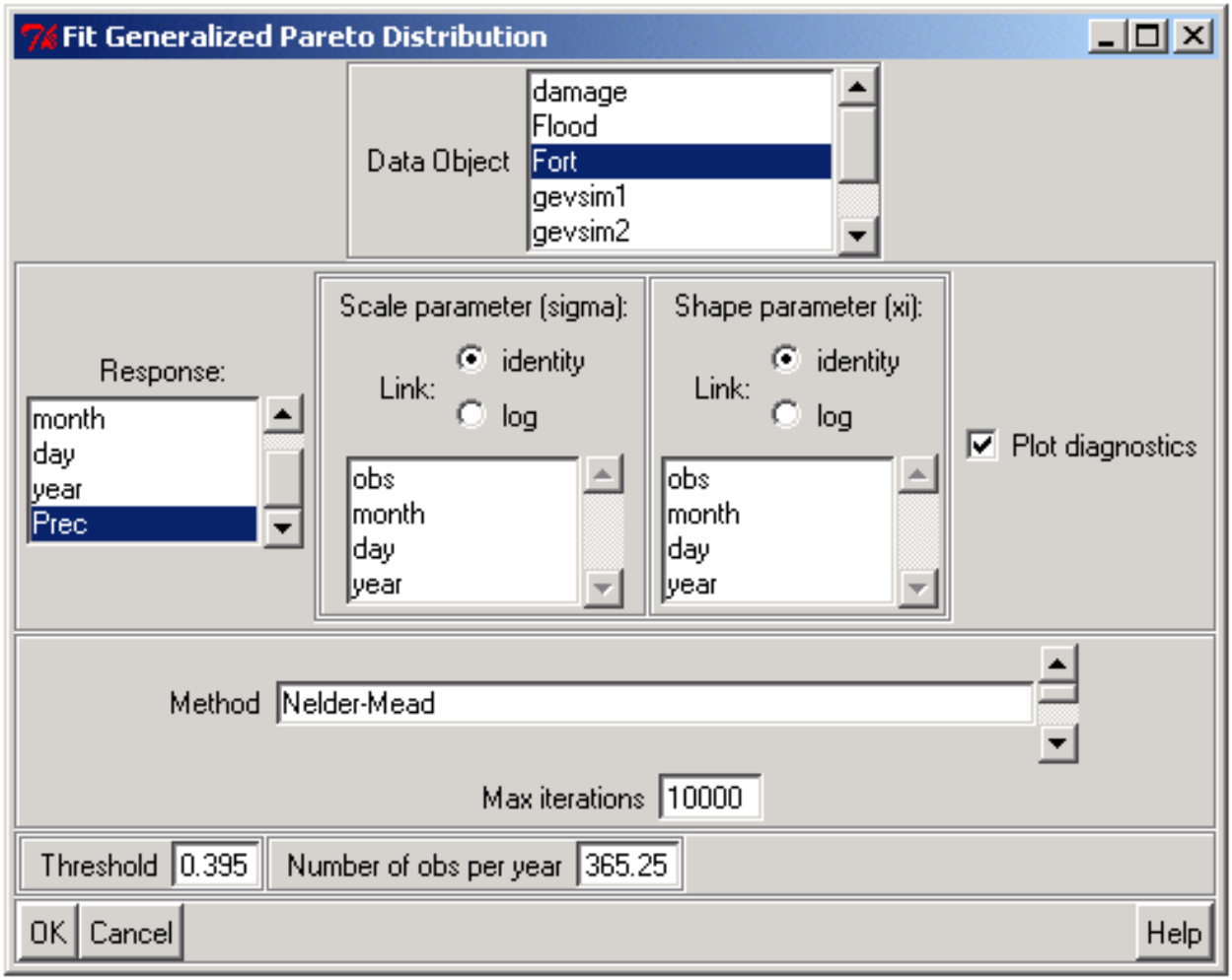


Figure 5.5: *Scatter plot of observed daily precipitation (inches) values by month for a Fort Collins, C.O. rain gauge.*

- *Select* **Prec** *from the* **Response** *listbox.* **Prec** *is removed from other listboxes.*
- *Check* **Plot diagnostics** *checkbutton.*
- *Enter* **0.395** *in the* **Threshold** *field* > **OK**



- Note that unlike the hurricane damage dataset, the **Number of obs per year** field is appropriate in this case because data are collected on a daily basis throughout the year.

The threshold of 0.395 inches is used as in Katz *et al.* [9].

A plot similar to that of Fig. 5.6 should appear along with summary statistics for the GPD fit in the main toolkit window. This fit yields MLE's of $\hat{\sigma} \approx 0.32$ inches (0.016 inches), $\hat{\xi} \approx 0.21$ (0.038), and a negative log-likelihood of about 85. Note that we are ignoring, for now, the annual cycle that is evident in Fig. 5.5.

Fig. 5.6 can be reproduced at any time in the following way.

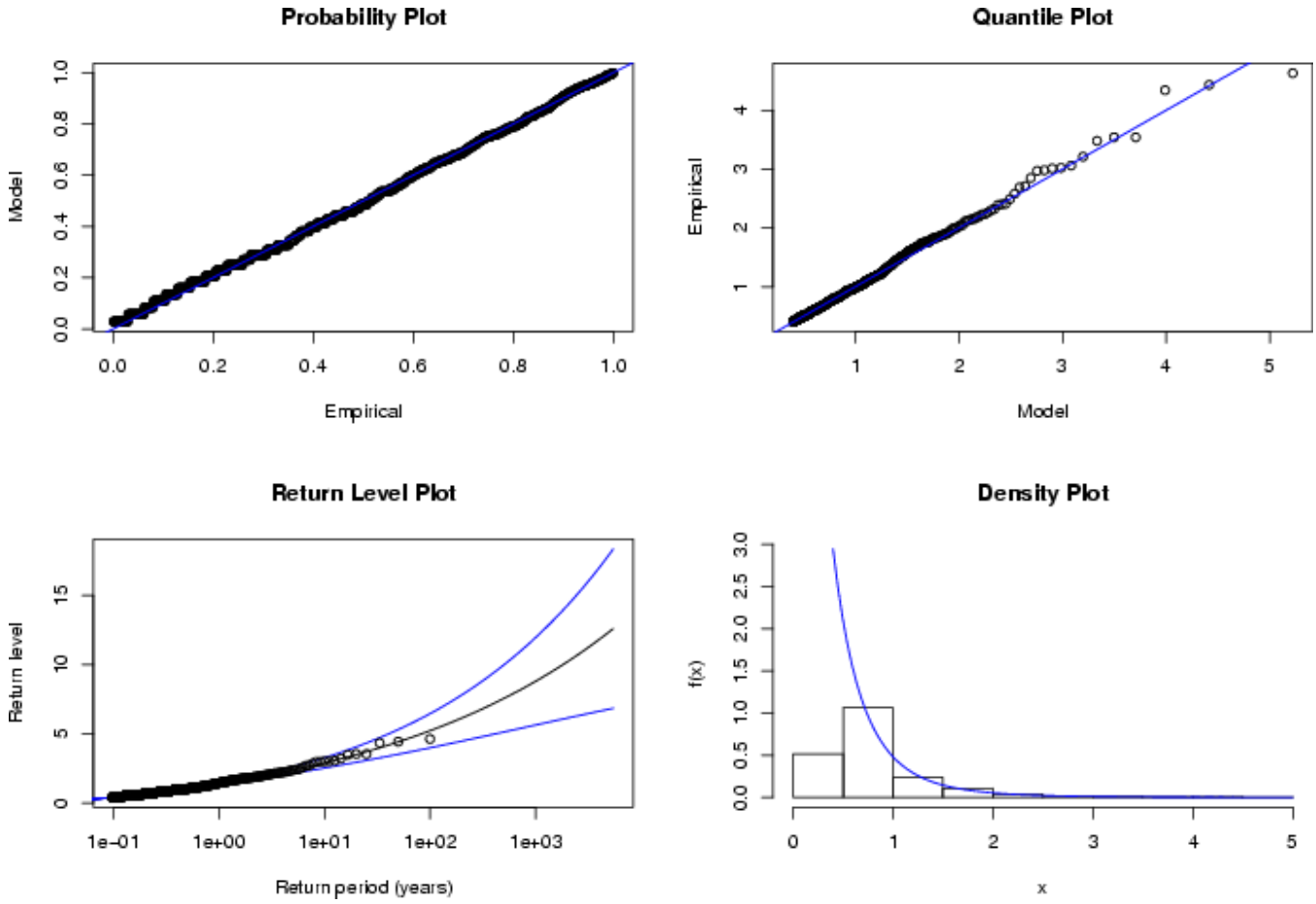


Figure 5.6: *Diagnostic plots for the GPD fit of the Fort Collins, C.O. Precipitation data using a threshold of 0.395 in.*

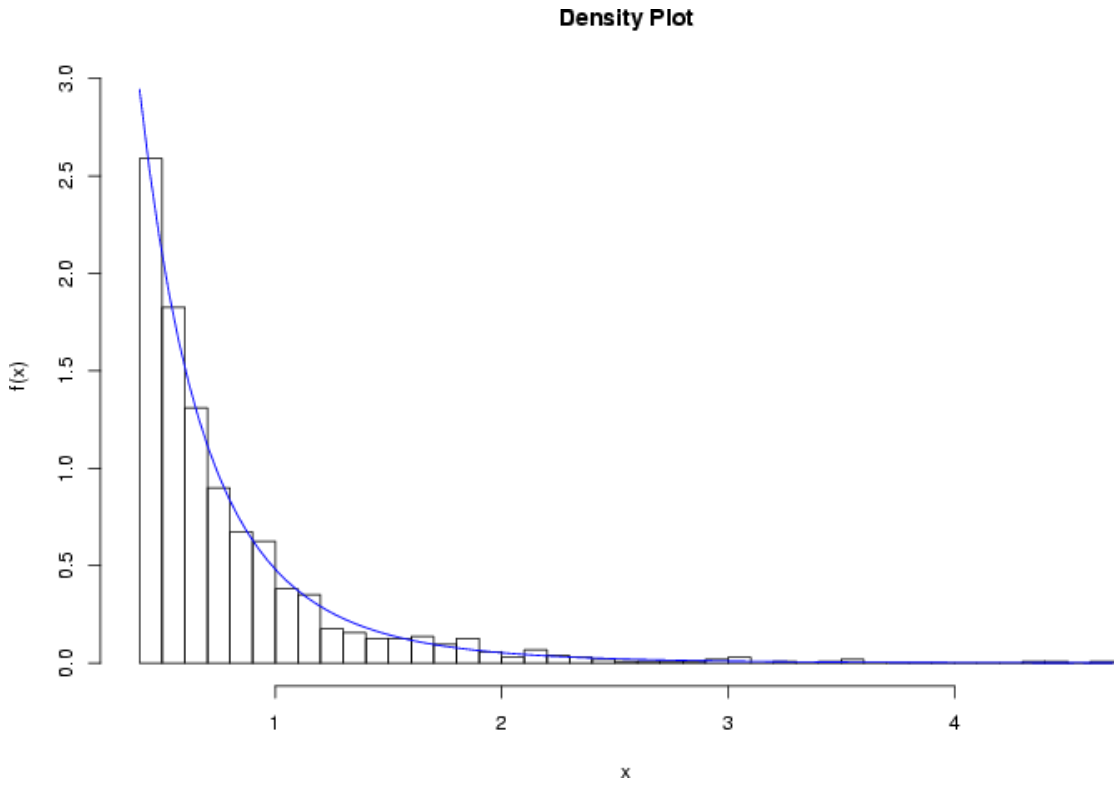


Figure 5.7: Histogram of GPD fit to Fort Collins precipitation (inches) data using the Friedman-Diaconis algorithm for determining the number of breakpoints.

- **Plot > Fit Diagnostics**
- *Select* **Fort** *from the* **Data Object** *listbox.*
- *Select* **gpd.fit1** *from the* **Select a fit** *listbox* **> OK**.

Fig. 5.7 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[14] for more details).

The general procedure for plotting a histogram of a fitted GPD function using `extRemes` is (identical to that of the GEV):

- **Plot > Fit with Histogram >** *New window appears*
- *Select an object from the* **Data Object** *listbox* **>**
- *Select the desired fit object from the* **Select a fit** *listbox.*
- *Select an algorithm from the* **Breaks Algorithm** *listbox and click* **OK**
- Histogram is plotted.

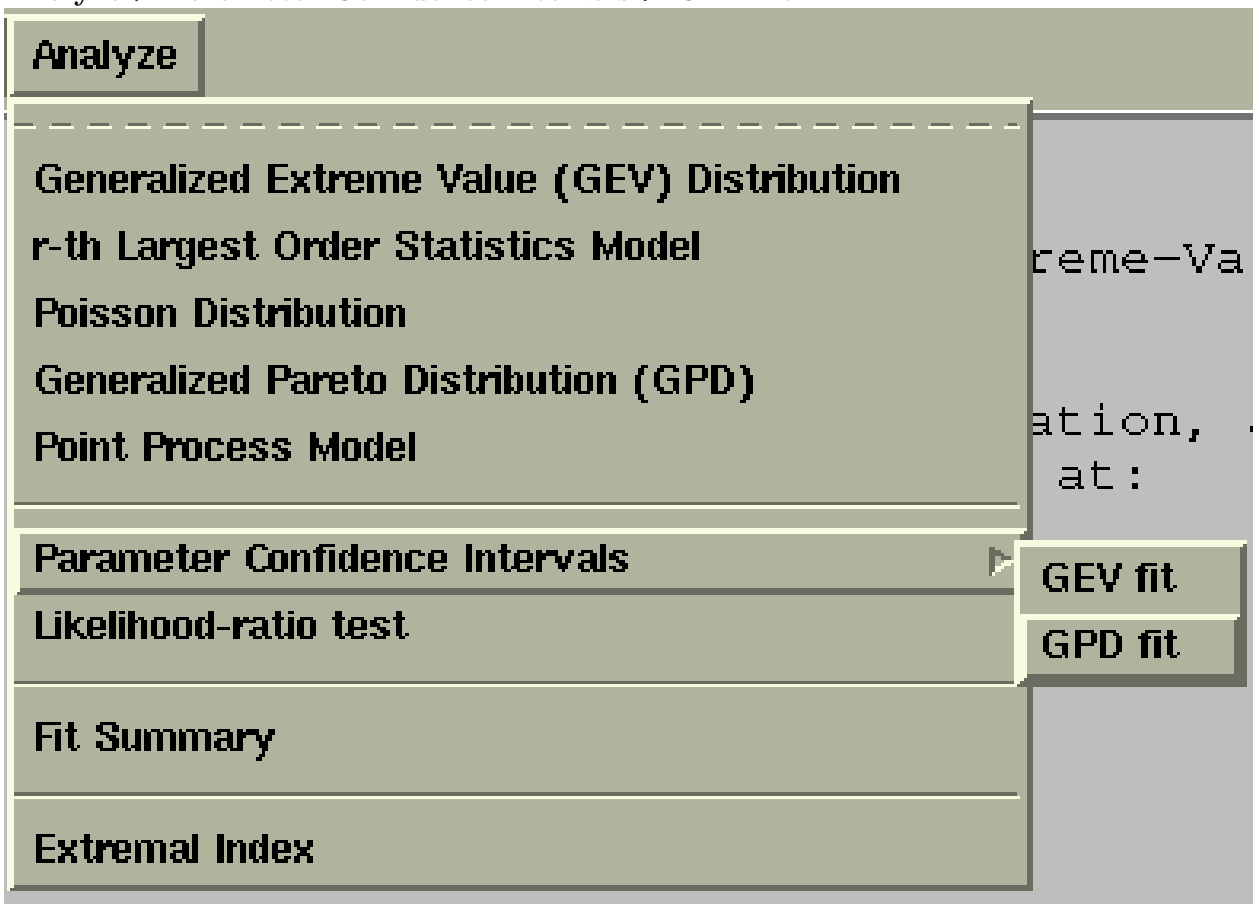
5.0.11 Return level and shape parameter (ξ) $(1-\alpha)$% confidence bounds

Confidence intervals may be estimated using the toolkit for both the return level and shape parameter (ξ) of both the GEV and GP distributions. See page 44 for more information on how the confidence intervals are obtained.

EXAMPLE: FORT COLLINS PRECIPITATION DATA

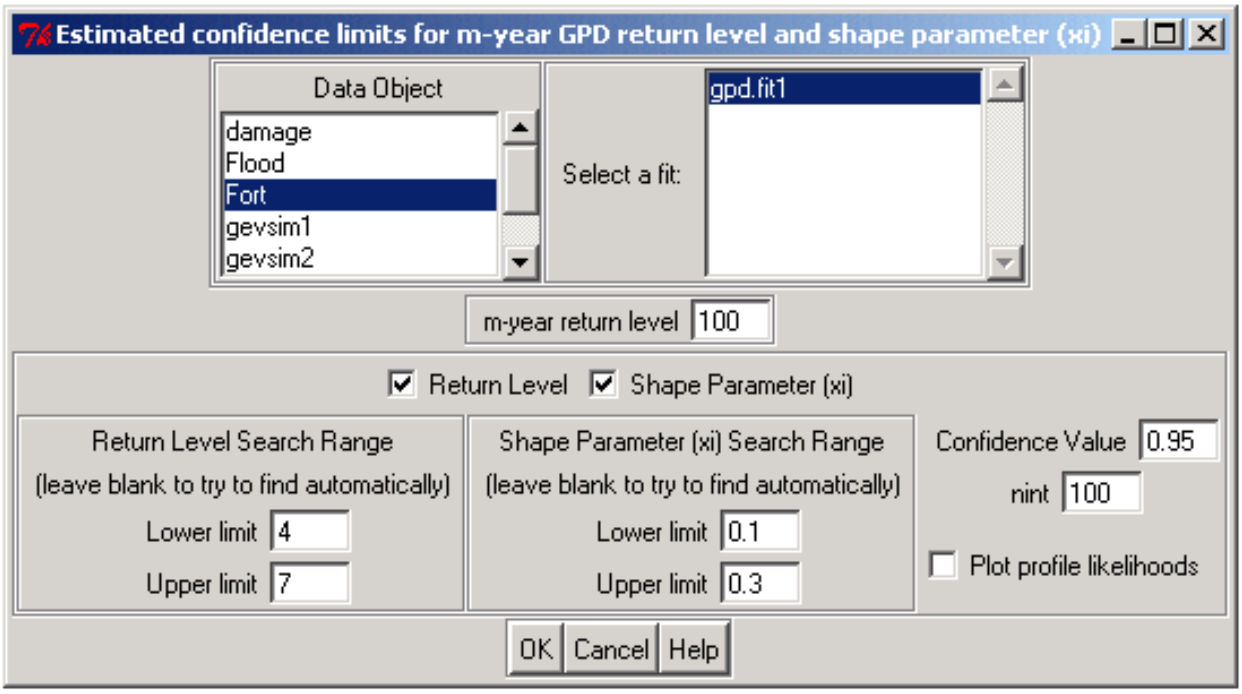
To estimate the confidence limits for the GPD shape parameter using `extRemes`:

- **Analyze > Parameter Confidence Intervals > GPD fit**



- *Select* **Fort** *from* **Data Object** *listbox.*
- *Select* **gpd.fit1** *from* **Select a fit** *listbox.*
- *Leave the default value of* 100 *in the* **m-year return level** *field.*
- *enter* 4 *in the* **Lower limit** *field of the* **Return Level Search Range**[8] *and* 7 *in the* **Upper limit** *field.*
- *enter* 0.1 *in* **Lower limit** *field of the* **Shape Parameter (xi) Search Range**[8]

and enter 0.3 *in the* **Upper limit** *field* > **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, consistent with the shape parameter being greater than zero. Visual inspection of the dashed vertical lines in Fig. 5.8 act as a guide to the accuracy of the displayed confidence limits; here the estimates shown appear to be accurate because 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.

5.0.12 Threshold Selection

Threshold selection is an important topic, and still an area of active research. It is desired to find a threshold that is high enough that the underlying theoretical development is valid,

[8] For the Fort Collins, C.O. precipitation data the MLE for the 100-year return level is near 5 inches and $\hat{\xi} \approx 0.19$, so a good search range for the confidence limits would include 5 and be wide enough to capture the actual limits. If any of the search range fields are left blank, `extRemes` will try to find a reasonable search limit (for each field left blank) automatically. It is a good idea to check the **plot profile likelihoods** checkbutton when searching for ranges automatically. This way, the profile likelihoods with vertical dashed lines at estimated limits will be displayed; if dashed lines intersect profile at lower horizontal line, then the estimate is reasonably accurate. For this example, 4 to 7 inches are used for the 100-year return level and 0.1 to 0.3 for the shape parameter.

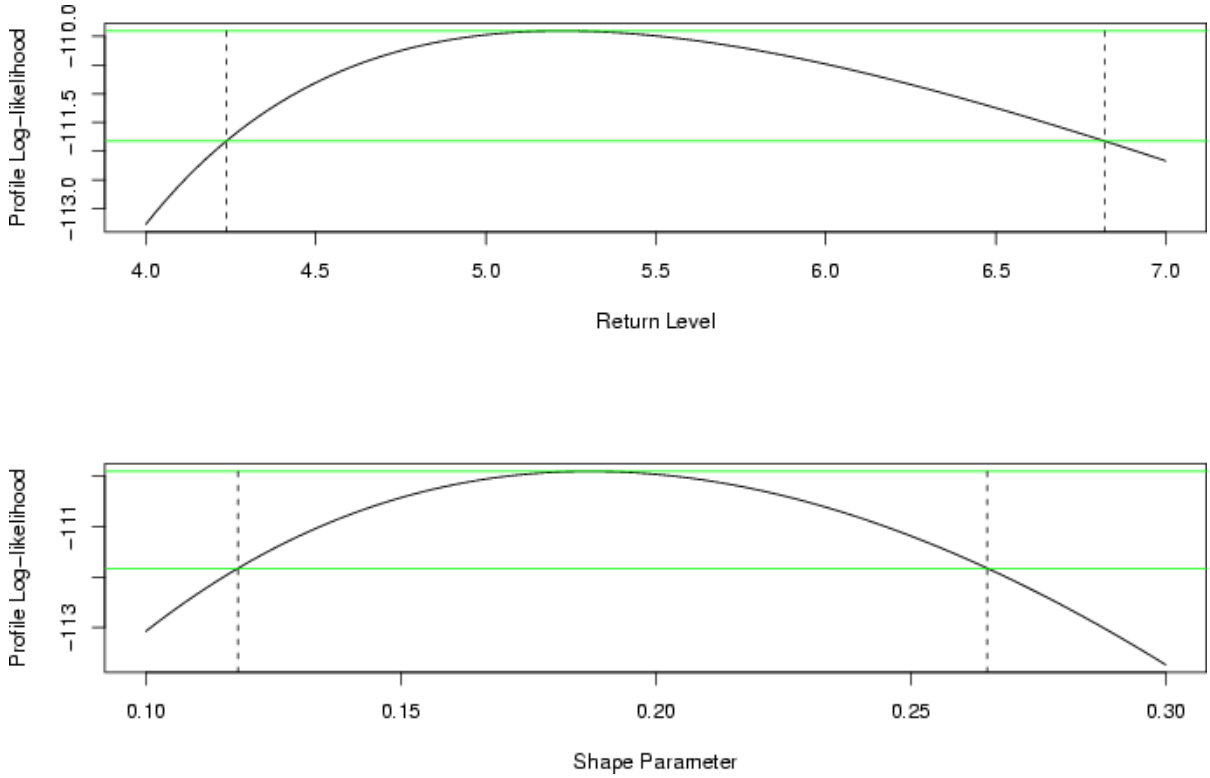


Figure 5.8: *Profile log-likelihood plots for GPD 100-year return level (inches) and shape parameter (ξ) for Fort Collins, C.O. precipitation data.*

but low enough that there is sufficient data with which to make an accurate fit. That is, selection of a threshold that is too low will give biased parameter estimates, but a threshold that is too high will result in large variance of the parameter estimates. Some useful *descriptive* tools for threshold selection are included with `extRemes`. Specifically, the mean excess, or mean residual life, plot and another method involving the fitting of data to a GPD several times using a range of different thresholds.

5.0.13 Threshold Selection: Mean Residual Life Plot

Mean residual life plots, also referred to as mean excess plots in statistical literature, can be plotted using `extRemes`. For more information on the mean residual life plot (and threshold selection) see appendix section B.0.27. The general procedure for plotting a mean residual life plot using `extRemes` is:

- **Plot** > **Mean Residual Life Plot** > *New window appears*
- *Select an object from* **Data Object** *listbox. Variables appear in* **Select Variable** *listbox. Select one.*
- *Choose other options* > **OK**.
- Mean residual life plot appears.

Example: Fort Collins precipitation

Fig. 5.9 shows the mean residual life plot for the Fort Collins, C.O. 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 because of the annual cycle (seasonality) that is being ignored here. Nevertheless, the plot appears roughly linear from about 0.3 to 2.5 inches and is erratic above 2.5 inches, so 0.395 inches is a plausible choice of threshold.

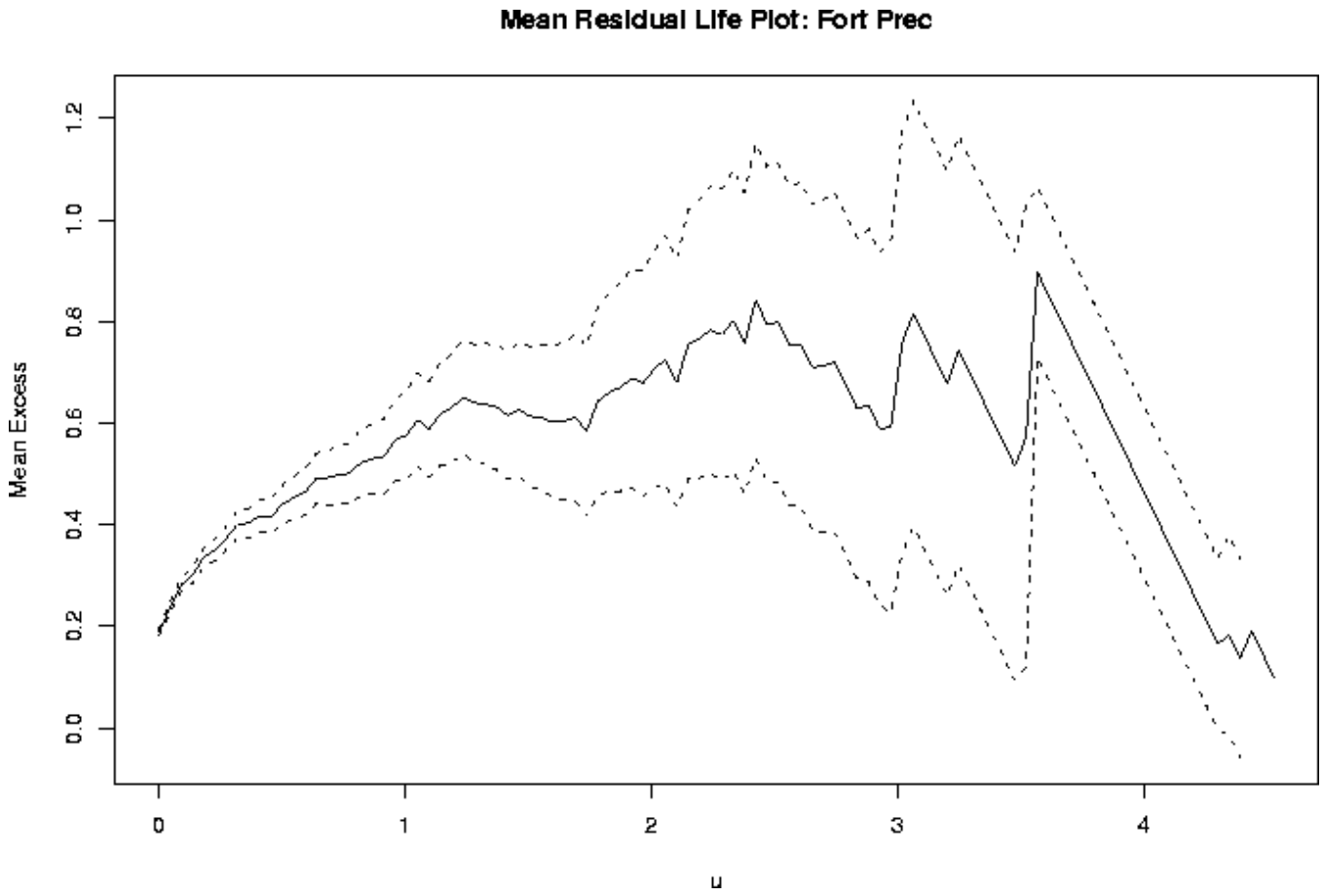
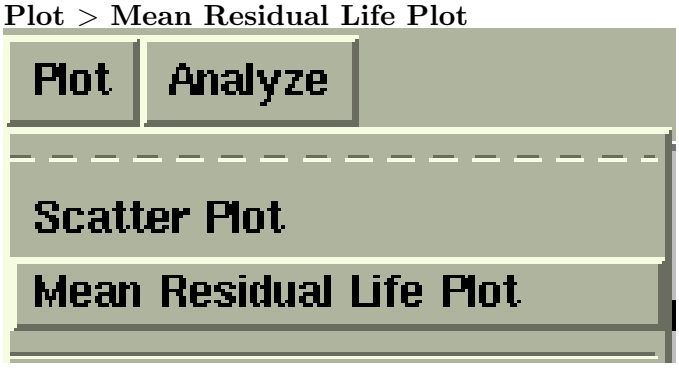


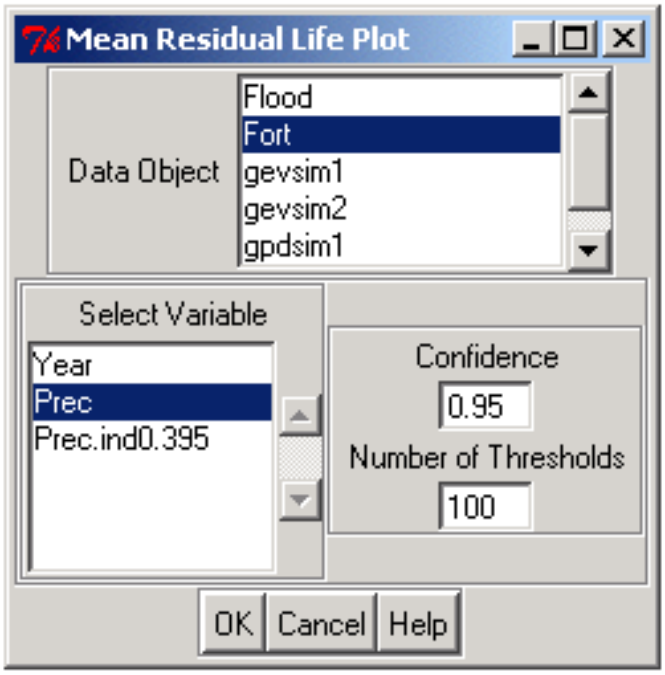
Figure 5.9: *Mean Residual Life Plot of Fort Collins precipitation data. Thresholds (u) vs Mean Excess precipitation (in inches).*

To plot Fig. 5.9 using `extRemes`:

- **Plot > Mean Residual Life Plot**



- *Select* **Fort** *from the* **Data Object** *listbox.*
- *Select* **Prec** *(the dependent variable) from the* **Select Variable** *listbox.* 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**.



5.0.14 Threshold Selection: Fitting data to a GPD Over a Range of Thresholds

The second method for trying to find a threshold requires fitting data to the GPD distribution several times, each time using a different threshold. The stability in the parameter

estimates can then be checked. The general procedure for fitting threshold ranges to a GPD is:

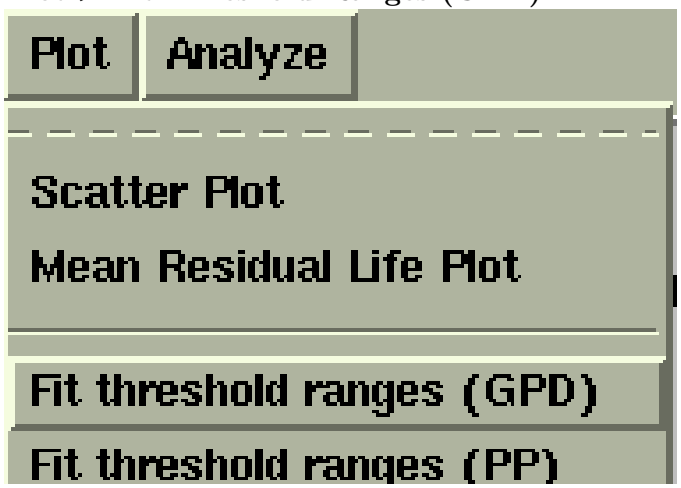
- **Plot** > **Fit Threshold Ranges (GPD)** > *New window appears*
- *Select a data object from* **Data Object** *listbox. Variables appear in* **Select Variable** *listbox. Select one*
- *Enter lower and upper limits and number of thresholds in remaining fields* > **OK**.
- If successfull, plot will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

Example: Fort Collins precipitation

Fig. 5.10 shows plots from having fit the GPD model for a range of 50 thresholds from 0.01 inches to 1 inch for the Fort Collins precipitation data (see section 5.0.10 for more information on these data). Fig. 5.10 suggests that, for the GPD model, a threshold of 0.395 inches is appropriate.

To create the plot from Fig. 5.10 using `extRemes`, do the following.

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



- *Select* **Fort** *from the* **Data Object** *listbox.*
- *Select* **Prec** *from the* **Select Variable** *listbox.*
- *Enter* 0.01 *in the* **Minimum Threshold** *field.*
- *Enter* 1 *in the* **Maximum Threshold** *field.*

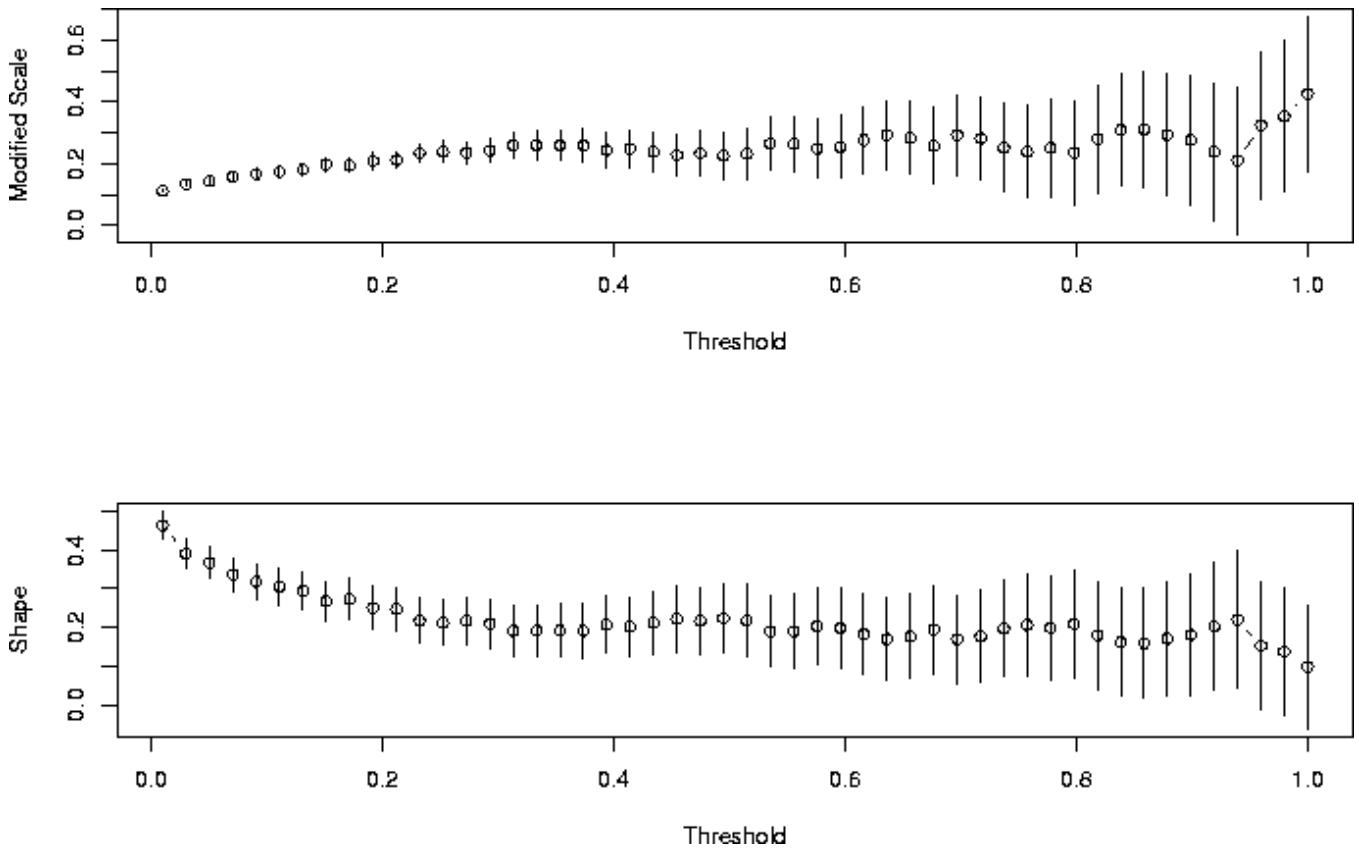
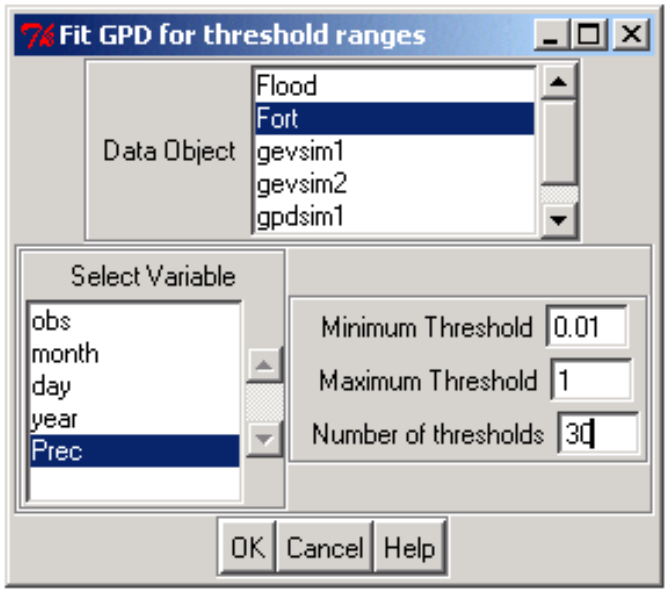


Figure 5.10: *GPD fits for a range of 50 thresholds from 0.01 inches to 1 inch for the Fort Collins precipitation dataset.*

- *Enter* 30 *in the* **Number of thresholds** *field* > **OK**.



Note that different values may be tried here as well, but the program will fail for certain choices. Keep trying different threshold ranges until it works.

Chapter 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 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 (see Coles [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 data to the GEV distribution that makes use of more information about the upper tail of the distribution than does the block maxima approach (Coles [3]).

6.0.15 Fitting data to a Point Process Model

Fig. 6.1 is not quite as easy to interpret as Fig. 5.10 for the GPD because of the fewer thresholds, but it seems that a threshold anywhere in the range of 0.30 to 0.40 inches would be appropriate.

To create the plot in Fig. 6.1 do the following.

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

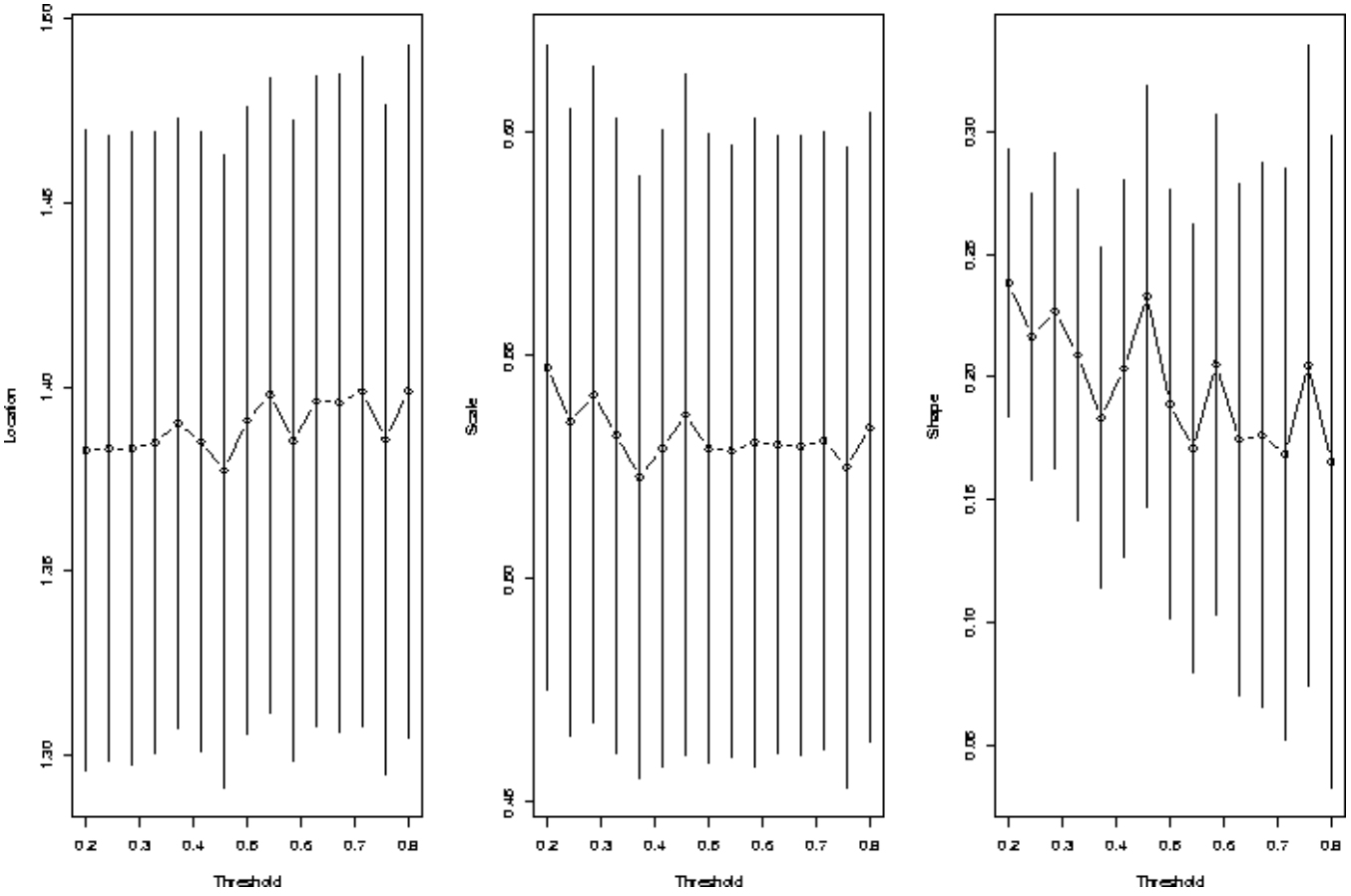
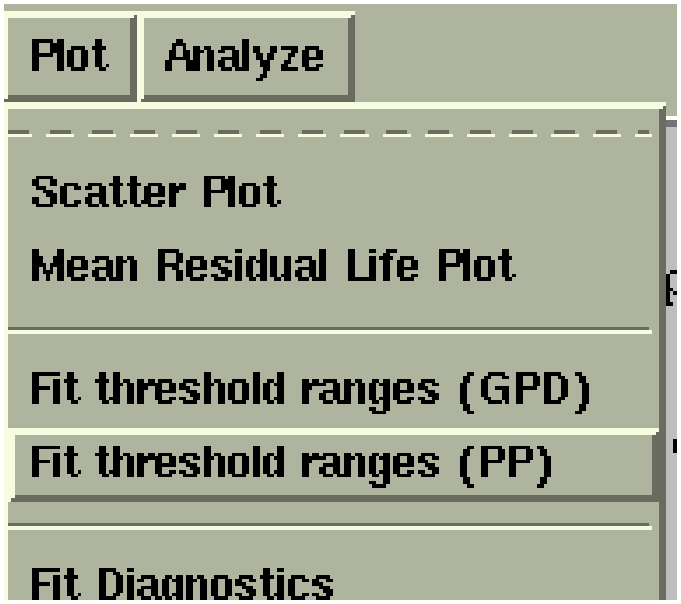
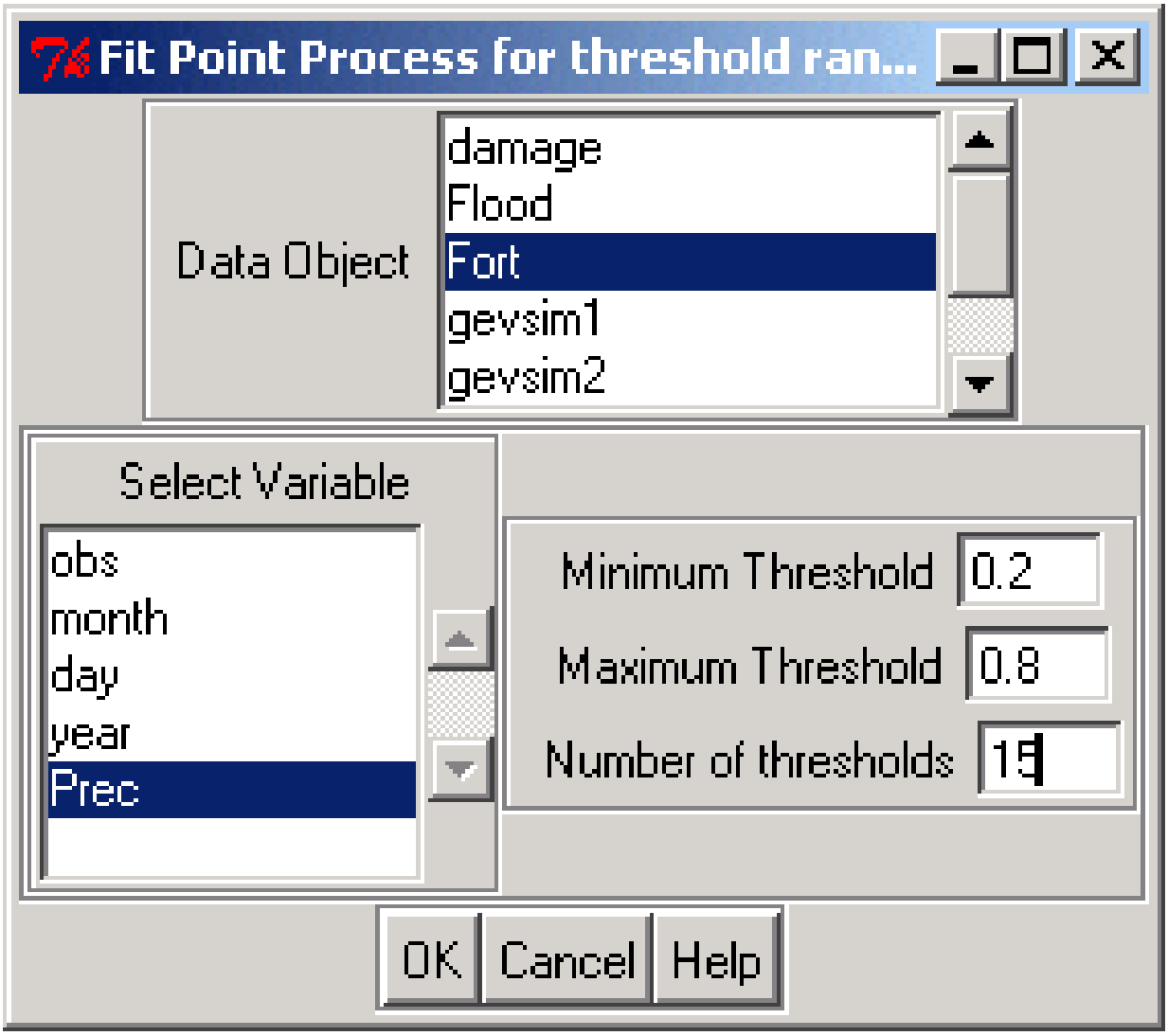


Figure 6.1: *Point process model fits for a range of 15 thresholds from 0.2 inches to 0.80 inches for the Fort Collins, C.O. precipitation dataset.*



- *Select* **Fort** *from the* **Data Object** *listbox.*
- *Select* **Prec** *from the* **Select Variable** *listbox.*
- *Enter* 0.2 *in the* **Minimum Threshold** *field*
- *Enter* 0.8 *in the* **Maximum Threshold** *field*
- *Change the* **Number of thresholds** *to* $15 >$ **OK**.



Once a threshold is selected, a point process model can be fitted. Fig. 6.2 shows diagnostic plots (probability and quantile plots) for such a fit.

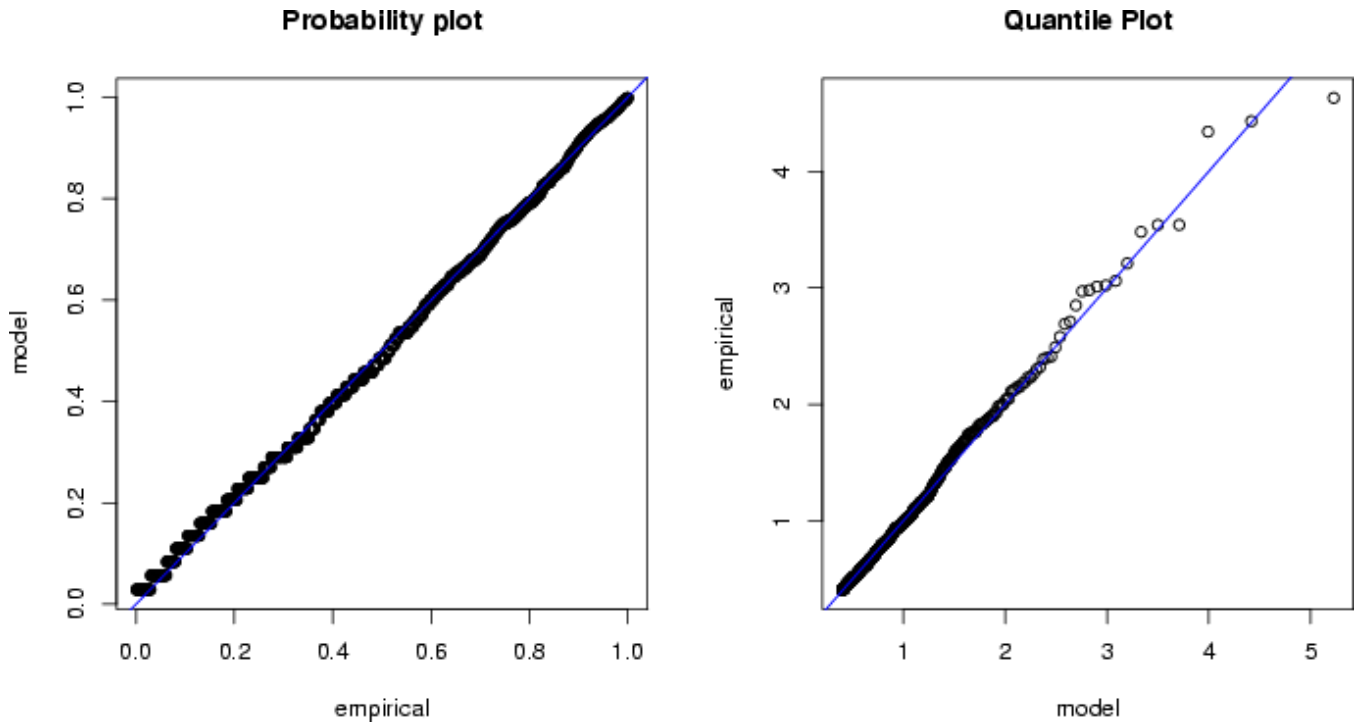
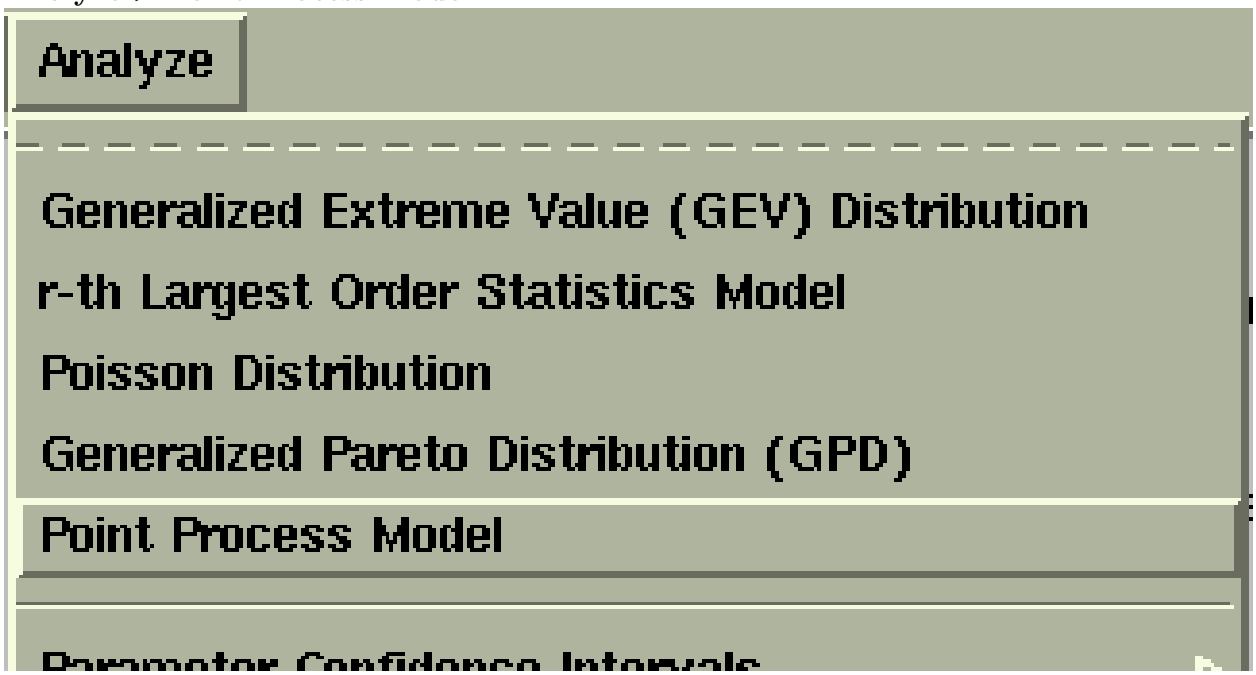


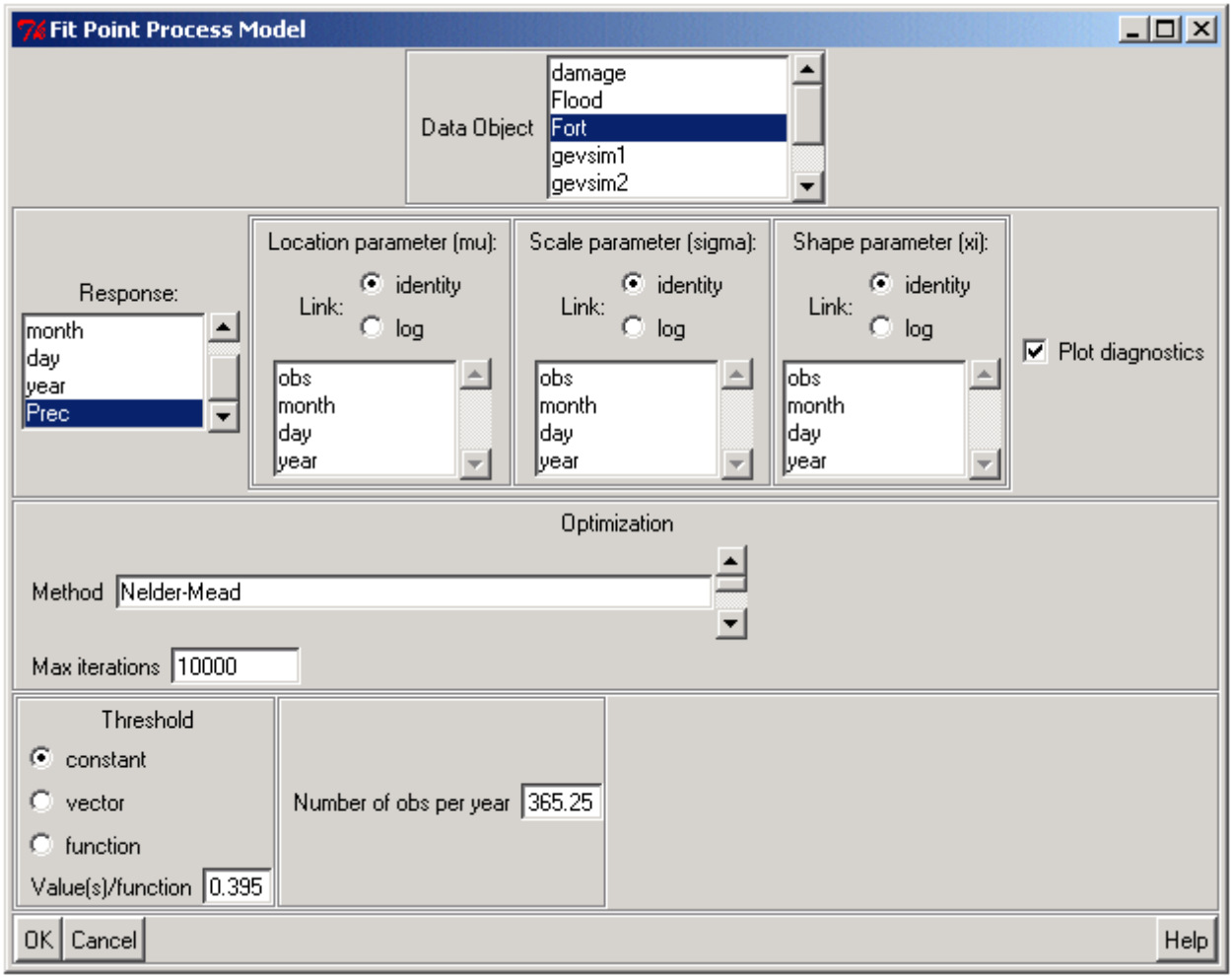
Figure 6.2: *Diagnostic plots for Fort Collins, C.O. precipitation (inches) data fit to a point process model.*

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** *listbox.*
- *Select* **Prec** *from the* **Response** *listbox.*
- *Check the* **Plot diagnostics** *checkbutton.*
- *Enter* 0.395 *in the* **Threshold value(s)/function** *field* > **OK**



MLE's found for this fit are: $\hat{\mu} \approx 1.38$ inches (0.043), $\hat{\sigma} \approx 0.53$ inches (0.037 inches) and $\hat{\xi} \approx 0.21$ (0.038) parameterized in terms of the GEV distribution for annual maxima, with negative log-likelihood of about -1359.82.

6.0.16 Relating the Point Process Model to the Poisson-GP

The parameters of the point process model can be expressed in terms of the parameters of the GEV distribution or, equivalently through transformations specified in appendix section B.0.28, in terms of the parameters of a Poisson process and of the GPD (i.e., a Poisson-GP model).

Example 1: Fort Collins Precipitation (no covariates)

When fitting the Fort Collins precipitation data to the point process model (using the BFGS optimization method) with a threshold of 0.395 and 365.25 observations per year, the following parameter estimates are obtained.

$\hat{\mu} \approx 1.38343$

$\hat{\sigma} \approx 0.53198$
$\hat{\xi} \approx 0.21199$
Parameters from fitting data to the GPD (using the **BFGS** optimization method) with a threshold of 0.395 and 365.25 observations per year are $\hat{\sigma}^* \approx 0.3225$ and $\hat{\xi} \approx 0.21191$– denoting the scale parameter of the GPD by σ^* to distinguish it from the scale parameter σ of the GEV distribution. Immediately, it can be seen that the value of $\hat{\xi}$ is very nearly identical to the estimate found for the point process approach. Indeed, the small difference can be attributed to differences in the numerical approximations. The other two parameters require a little more work to see that they correspond.

Specifically, because there are 1,061 observations exceeding the threshold of 0.395 inches out of a total of 36,524 observations, the (log) MLE for the Poisson rate parameter is $\log \hat{\lambda} = \log[365.25\frac{1061}{36524}] \approx 2.3618$ per year.

Plugging into Eqs. (B.3) and (B.4) (section B.0.28) gives

$$\log \hat{\sigma} = \ln(0.3225) + 0.2119(2.3618) \approx -0.63118 \Rightarrow \hat{\sigma} \approx \exp(-0.6311) \approx 0.53196$$

$$\hat{\mu} = 0.395 - \frac{0.53196}{0.2119}(10.61^{-0.2119} - 1) \approx 1.3835$$

both of which are very close to the respective MLEs of the point process model.

Example 2: Phoenix summer minimum daily temperature

The Phoenix minimum temperature data included with this toolkit represents a time series of minimum and maximum temperatures (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 [17] or Balling *et al.* [1]. Temperature is a good example of data that may have dependency issues because of the tendency of hot (or cold) days to follow other hot (or cold) days. However, we do not deal with this issue here (see chapter 7). For this example, load the **Tphap.R** dataset and save it (in R) as **Tphap**. The minimum temperatures (degrees Fahrenheit) are shown in Fig. 6.3. Note the increasing trend evident from the superimposed regression fit. Again, we will not consider this trend here, instead we defer this topic to chapter 7.

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 the extreme value distribution theory for maxima can be applied to minima. That is, $-\max(-X_1, \ldots, -X_n) = \min(X_1, \ldots, X_n)$ This transformation can be easily made using `extRemes`.

- **File > Transform Data > Negative**

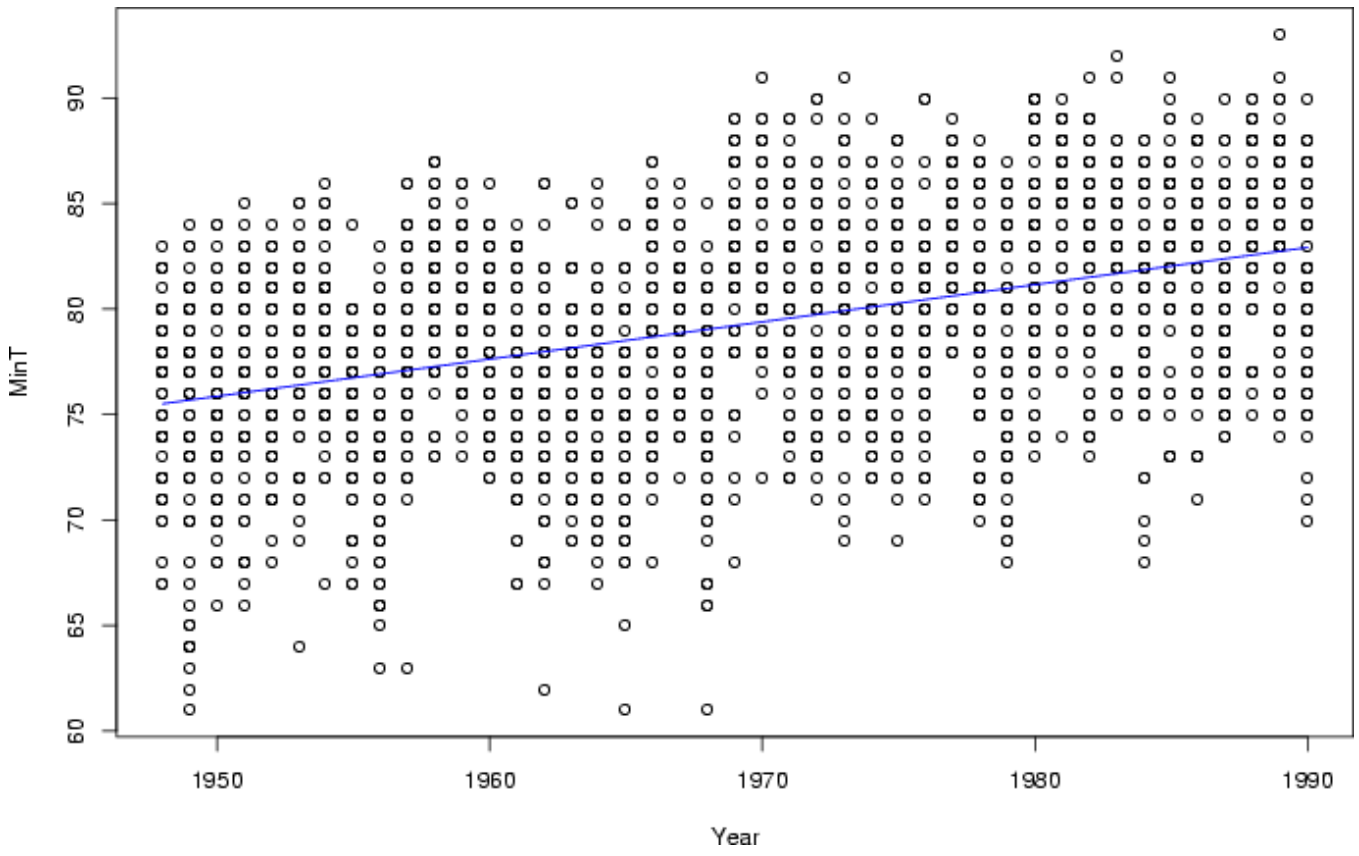
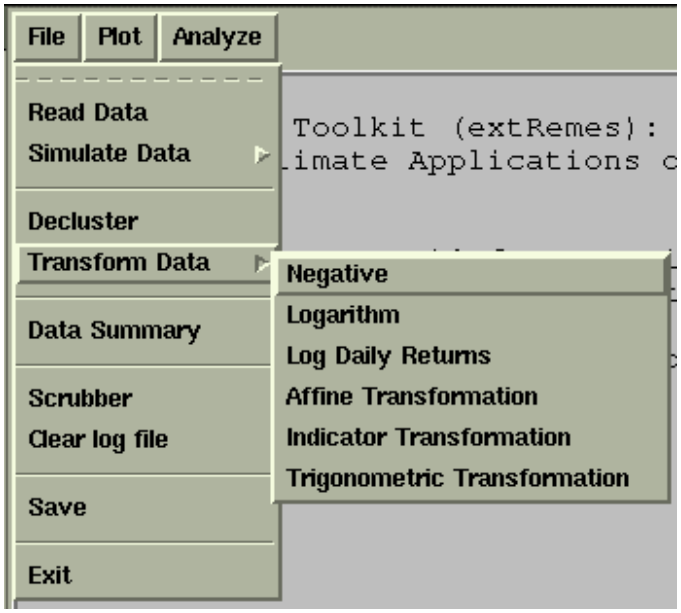
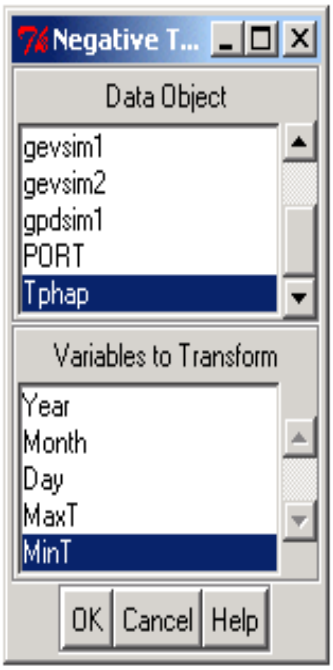


Figure 6.3: *Scatter plot of minimum temperature (degrees Fahrenheit), with regression line, for the summer months of July through August at Sky Harbor airport in Phoenix, A.Z.*



- *Select* **Tphap** *from the* **Data Object** *listbox.*
- *Select* **MinT** *from the* **Variables to Transform** *listbox* > **OK**.



For the Phoenix minimum temperature series, the Poisson log-rate parameter for a threshold of -73 degrees (using the negative of minimum temperature, **MinT.neg**) is $\log \hat{\lambda} = \log(62 \cdot \frac{262}{2666}) \approx 1.807144$ per year, where there are 62 days in each "year" or summer season (covers two months of 31 days each; see appendix section B.0.28) and 262 exceedances out

of 2,666 total data points. MLEs (using the BFGS method) from fitting data to the GPD are $\hat{\sigma}^* \approx 3.91$ degrees (0.303 degrees) and $\hat{\xi} \approx -0.25$ (0.049), and from fitting data to the point process model: $\hat{\mu} \approx -67.29$ degrees (0.323 degrees), $\hat{\sigma} \approx 2.51$ degrees (0.133 degrees) and $\hat{\xi} \approx -0.25$ (0.049). Clearly, the shape parameters of the two models match up. Using Eq. (B.3) of appendix section B.0.28, the derived scale parameter for the point process model is $\log \hat{\sigma} \approx 0.92$, or $\hat{\sigma} \approx 2.51$ degrees (the same as that of the point process estimate fitted directly). Using Eq. (B.4) gives $\hat{\mu} \approx -67.29$ degrees (also equivalent to the point process estimate fitted directly).

Clearly, the probability and quantile plots (Figs. 6.4 and 6.5) are identical, but the curvature in the plots indicates that the assumptions for the point process model may not be strictly valid–although, the plots are not too far from being straight.

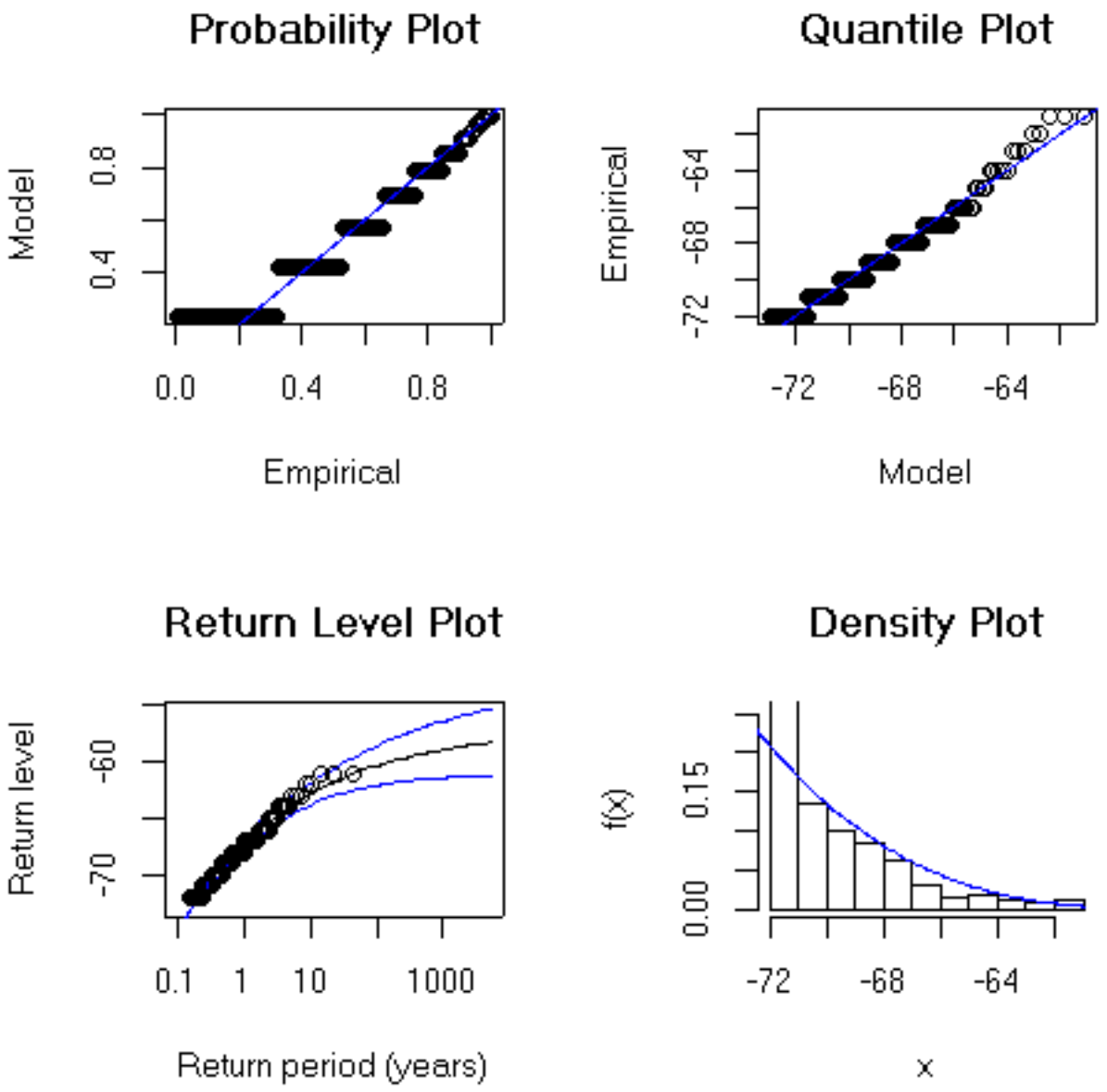


Figure 6.4: *Diagnostic plots of GPD fit for Phoenix Sky Harbor airport summer minimum temperature (degrees Fahrenheit) data (***Tphap***).*

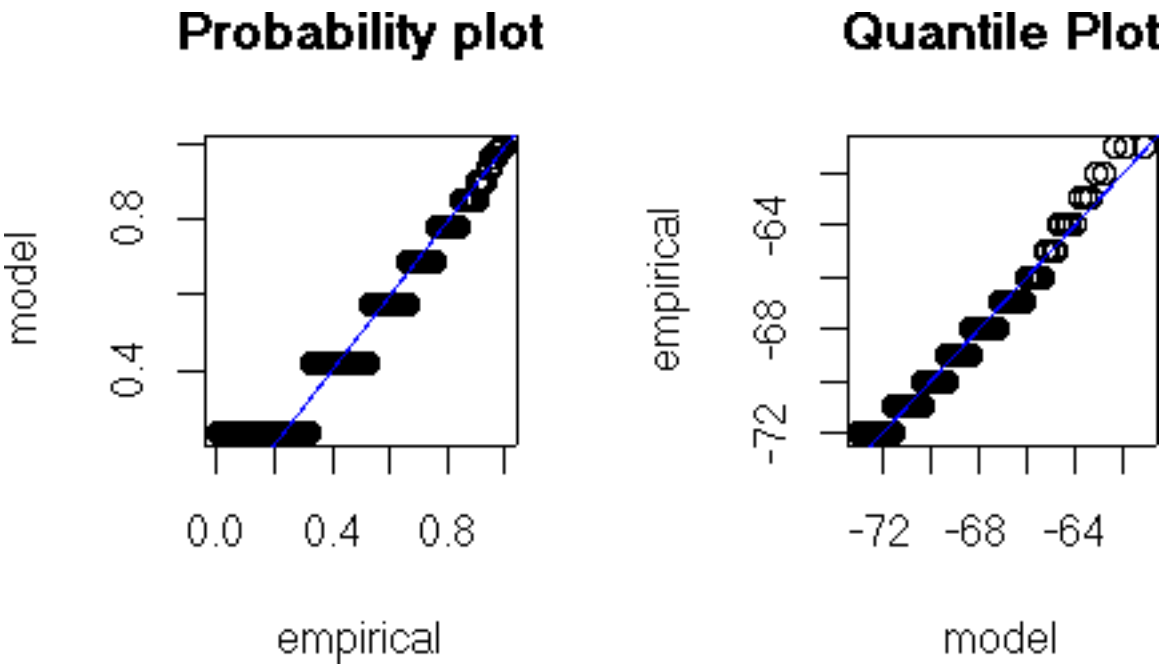


Figure 6.5: *Diagnostic plots of point process fit for Phoenix Sky Harbor airport summer minimum temperature (degrees Fahrenheit) data (***Tphap***).*

Chapter 7

Extremes of Dependent and/or Nonstationary Sequences

Much of the theory applied thus far assumes independence of the data, which may 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 independent random variables is to a stationary series, which is realistic for many physical processes. Here the variables may be mutually dependent, but the stochastic properties are homogeneous over time (see Coles [3] Ch. 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.

7.0.17 Parameter Variation

It is possible to allow parameters of the extreme value distributions to vary as a function of time or other covariates. In doing so, it is possible to account for some nonstationarity sequences. One could, for example, allow the location parameter, μ, of the GEV(μ, σ, ξ) distribution to vary cyclically with time by replacing μ by $\mu(t) = \mu_0 + \mu_1 \sin(\frac{2\pi t}{365.25}) + \mu_2 \cos(\frac{2\pi t}{365.25})$. When allowing the scale parameter to vary, it is important to ensure that $\sigma(t) > 0$, for all t. Often a link function that only yields positive output is employed. The *log* link function is available for this purpose as an option with `extRemes`. For example, the model $\sigma(x) = \exp(\beta_0 + \beta_1 x)$ can be employed using the default linear representation $\log \sigma(x) = \beta_0 + \beta_1 x$ by checking the appropriate **Link** button. While it is also possible to allow the shape parameter to vary, it is generally difficult to estimate this parameter with precision; so it is unrealistic to allow this parameter to vary as a smooth function. One alternative is to allow it to vary on a larger scale (e.g., fit a different distribution for each

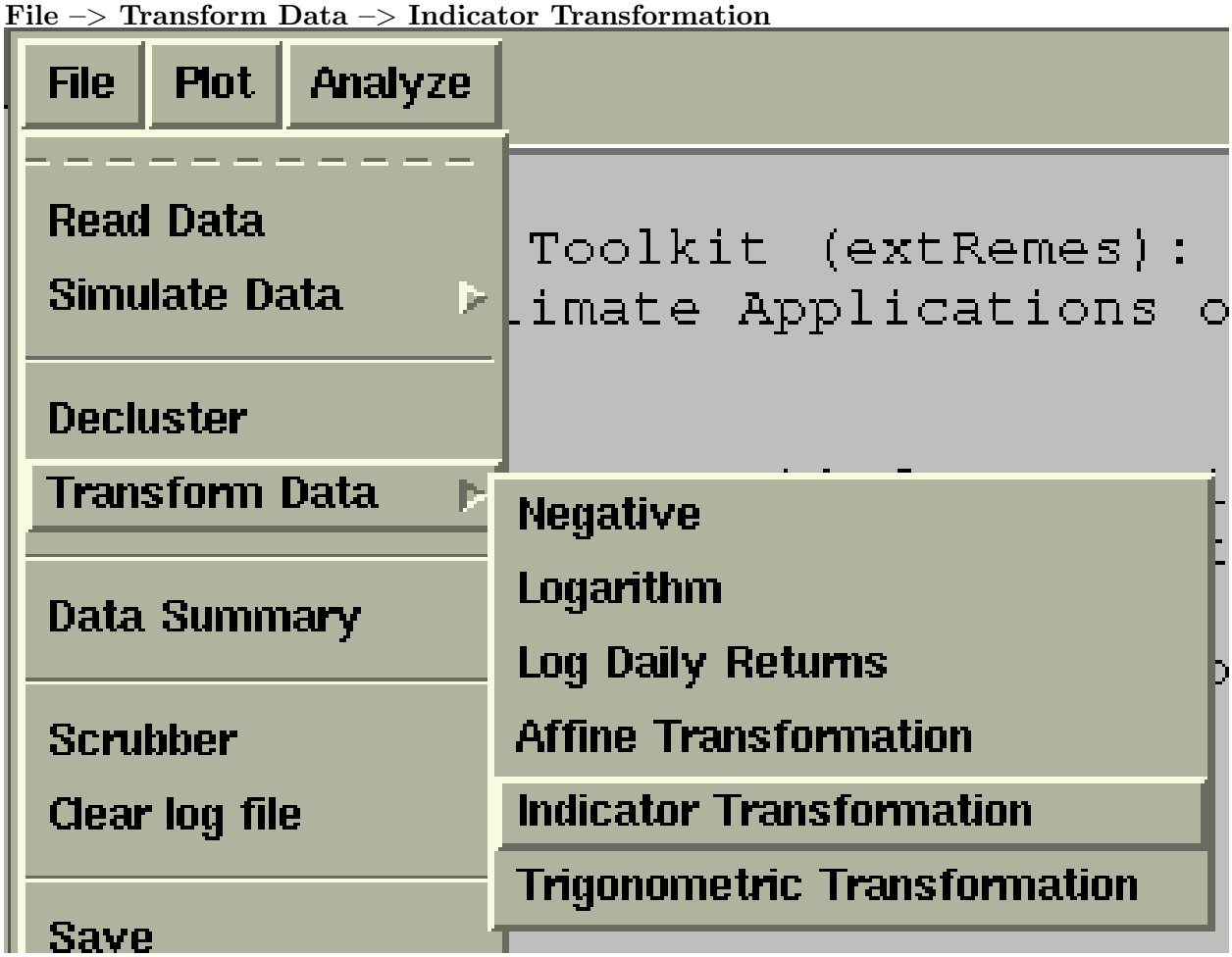
season) if enough data are available (see, for example, Coles [3] section 6.1).

EXAMPLE 2: FORT COLLINS PRECIPITATION (ANNUAL CYCLE)

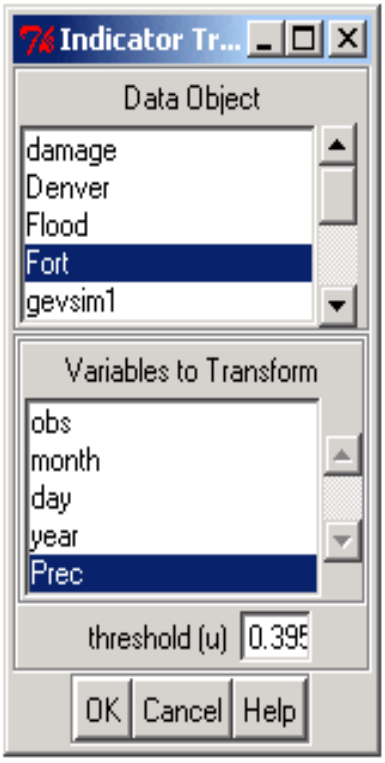
It is also possible to include a seasonal trend in the model; either within the model parameters or within the threshold. Here, we shall include an annual cycle in the scale parameter. To do this, we first need to create a few new columns in the data.

First, we require an indicator variable that is 1 whenever the precipitation exceeds 0.395 inches, and 0 otherwise. Using `extRemes`:

- **File –> Transform Data –> Indicator Transformation**



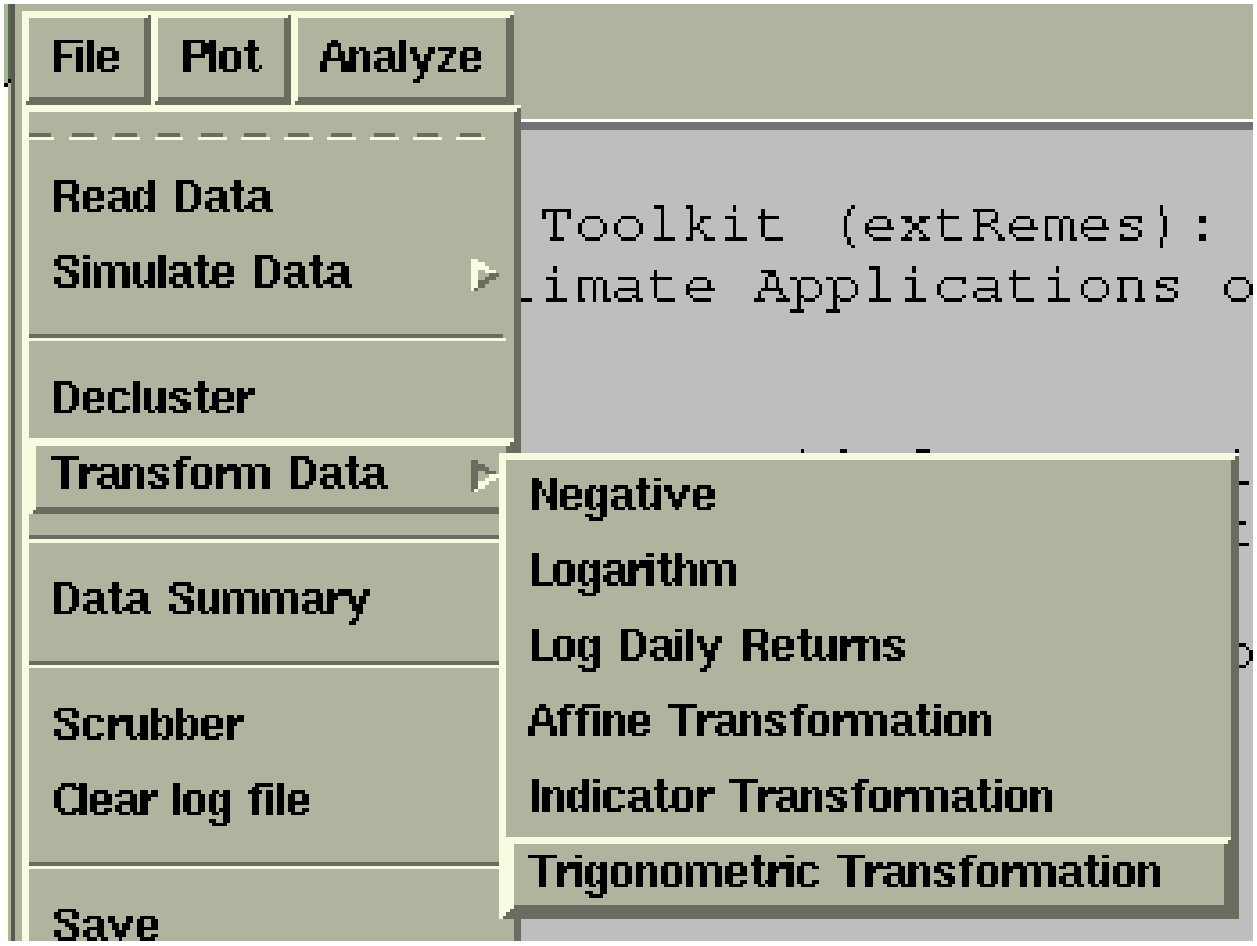
- *Select* **Fort** *from the* **Data Object** *listbox.*
- *Select* **Prec** *from the* **Variables to Transform** *listbox.*
- *Enter* 0.395 *in the* **threshold (u)** *field* –> **OK**.



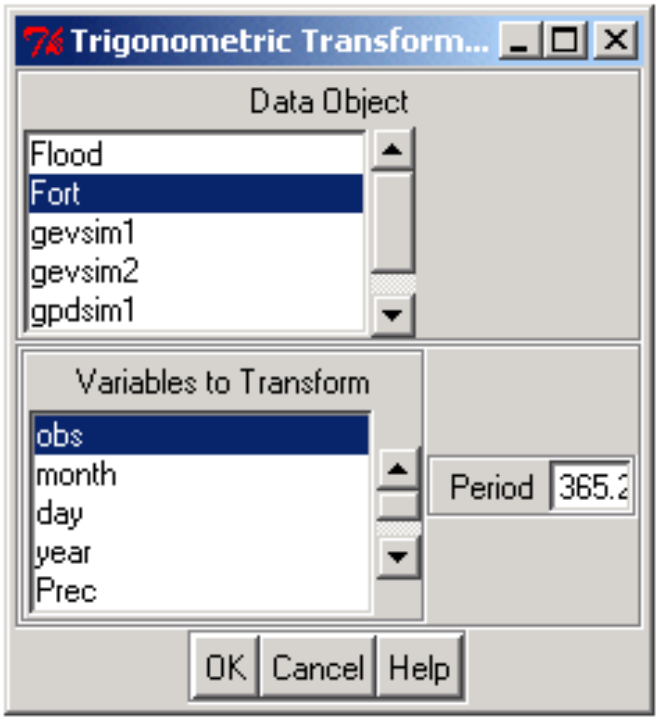
There should now be a new column called **Prec.ind0.395** in the Fort Collins precipitation data matrix, **Fort$data**.

Next, we need to add columns that will account for annual cycles. Specifically, we want to add columns that give $\sin(\frac{2\pi t}{365.25})$ and $\cos(\frac{2\pi t}{365.25})$, where t is simply the **obs** column found in **Fort$data** (i.e., $t = 1, \ldots, 36524$). Using `extRemes`:

- **File > Transform Data > Trigonometric Transformation**



- *Select* **Fort** *from the* **Data Object** *listbox.*
- *Select* **obs** *from the* **Variables to Transform** *listbox.*
- *Leave the value of* **Period** *at the default of* $365.25 >$ **OK**.

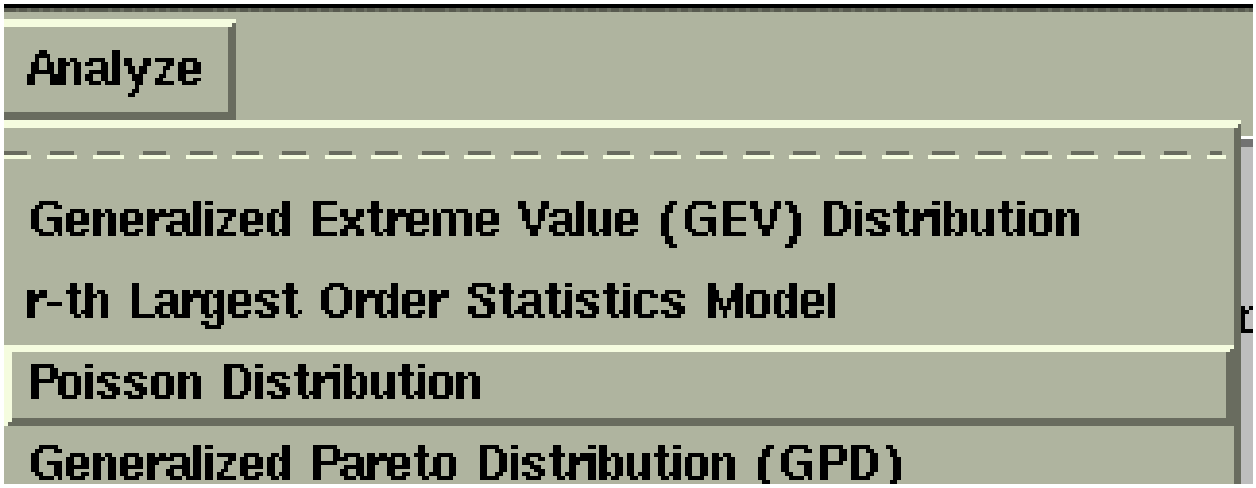


There should now be two new columns in **Fort$data** with the names **obs.sin365**[9] and **obs.cos365**[9]. Now, we are ready to incorporate a seasonal cycle into some of the parameters of the Poisson-GP model for the Fort Collins precipitation data. We begin by fitting the Poisson rate parameter (λ) as a function of time. Specifically, we want to find

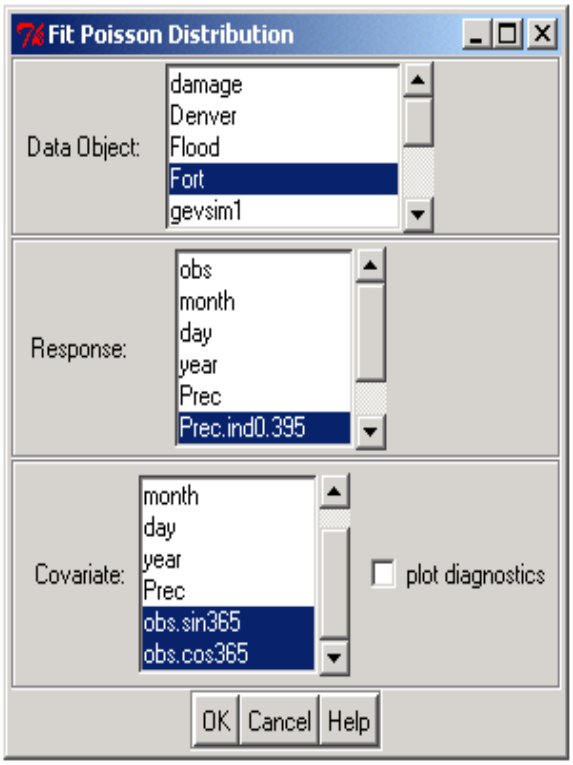
$$\log \lambda(t) = \beta_0 + \beta_1 \sin(\frac{2\pi t}{365.25}) + \beta_2 \cos(\frac{2\pi t}{365.25}) = \beta_0 + \beta_1 \cdot \textbf{obs.sin365} + \beta_2 \cdot \textbf{obs.cos365}. \tag{7.1}$$

- **Analyze –> Poisson Distribution**

[9]Note: because of the naming convention used by `extRemes` the trigonometric transformations with periods of 365 days cannot exist simultaneously with periods of, for example, 365.25 days. By default, and in order to prevent accidental deletion of data, `extRemes` will not allow a transformation if there is already a data column with the same name. In the present example, if a period of 365 is desired, the new names would also be obs.sin365 and obs.cos365; so both of these columns must be removed (e.g, using the Scrubber function under **File**) before invoking this transformation.



- *Select* **Fort** *from the* **Data Object** *listbox.*
- *Select* **Prec.ind0.395** *from the* **Response** *listbox.*
- *Select* **obs.sin365** *and* **obs.cos365** *from the* **Covariate** *listbox* > **OK**.

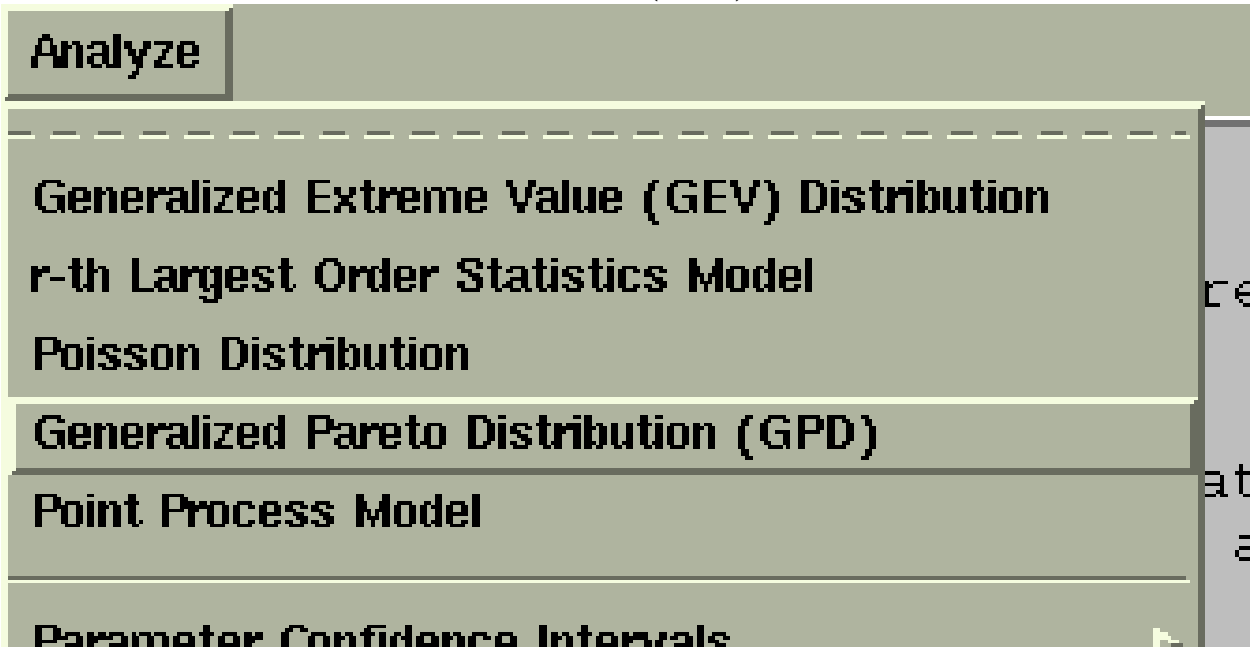


Results from fitting the Poisson rate parameter with an annual cycle (Eq. (7.1)) are $\hat{\beta}_0 \approx -3.72$ (0.037), $\hat{\beta}_1 \approx 0.22$ (0.046) and $\hat{\beta}_2 \approx -0.85$ (0.049). Note also that the likelihood-ratio against the null model (Example 1 above) is about 355 with associated p-value ≈ 0, which indicates that the addition of an annual cycle is significant.

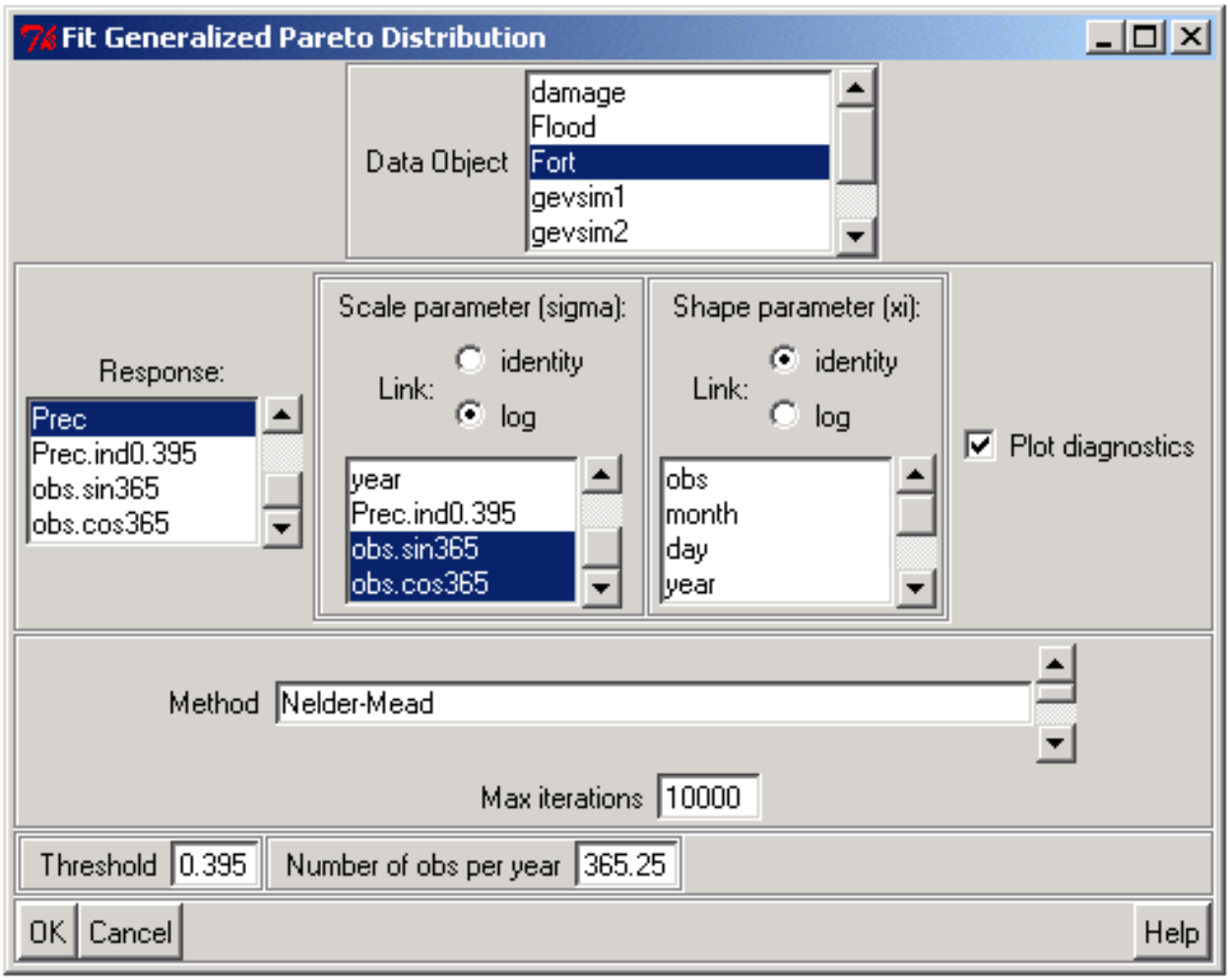
Next, we fit the GPD with the same annual cycle as a covariate in the scale parameter. That is, the scale parameter is modeled by

$$\log \sigma(t) = \sigma_0 + \sigma_1 \sin(\frac{2\pi t}{365.25}) + \sigma_2 \cos(\frac{2\pi t}{365.25}). \tag{7.2}$$

- **Analyze > Generalized Pareto Distribution (GPD) >**



- *Select* **Fort** *from the* **Data Object** *listbox.*
- *Select* **Prec** *from the* **Response** *listbox.*
- *Select* **obs.sin365** *and* **obs.cos365** *from the* **Scale parameter (sigma)** *listbox.*
- *Check the* **log** *radiobutton as the* **Link**.
- *Optionally check* **Plot diagnostics** *checkbutton.*
- *Enter* **0.395** *in the* **Threshold** *field* **> OK**



MLE parameter estimates for the scale parameter from Eq. (7.2) are $\hat{\sigma}_0 \approx -1.24$ (0.053), $\hat{\sigma}_1 \approx 0.09$ (0.048) and $\hat{\sigma}_2 \approx -0.30$ (0.069), and for the shape parameter $\hat{\xi} \approx 0.18$ (0.037). The negative log-likelihood value is about 73, and the likelihood-ratio test between this fit and that of section 5.0.10 Example 2 is about 24 (associated p-value nearly zero) indicating that inclusion of the annual cycle is significant.

7.0.18 Nonconstant Thresholds

In addition to varying parameters of the GPD to account for dependencies, it is also possible to vary the threshold. For some, such as engineers, interest may be only 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. Example: 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, similar to example 1, or one might use a trigonometric function to vary the threshold for each month. The latter will be employed here.

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

Fig. 7.1 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 R commands created the plot in Fig. 7.1.

```
> prec <- Fort$data[,"Prec"]
> plot( mths, Fort$data[,"Prec"], xlab="Month", ylab="precipitation (inches)",
xaxt="n")
> axis(1, labels=c("Jan", "Feb", "Mar", "Apr", "May", "Jun", "Jul", "Aug", "Sep",
"Oct", "Nov", "Dec"), at=1:12)
> abline( h=0.4)
> lines( mths[order(mths)], u.fortcollins[order(mths)], col="blue")
```

Fitting data to a point process model using `u.fortcollins` to fit a nonconstant (seasonal) threshold gives parameter estimates: $\hat{\mu} \approx 1.40$ 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. The ideal model would be based on a nonconstant threshold, but it is also possible to include annual cycles in the parameters; compare estimates to those found when including a seasonal cycle in the scale parameter from section 6.0.15. Inspection of the diagnostic plots (Fig. 7.2) 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, so the lower thresholds are necessary.

7.0.19 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, some empirical rule is used to define clusters of exceedances, maximums within each cluster are identified and cluster maxima are fit to the GPD; assuming independence among cluster maxima.

One simple way to determine clusters is commonly known as runs declustering. First, specify a threshold and define clusters to be wherever there are consecutive exceedances of this threshold. Once a certain number of observations, the run length, call it r, falls below

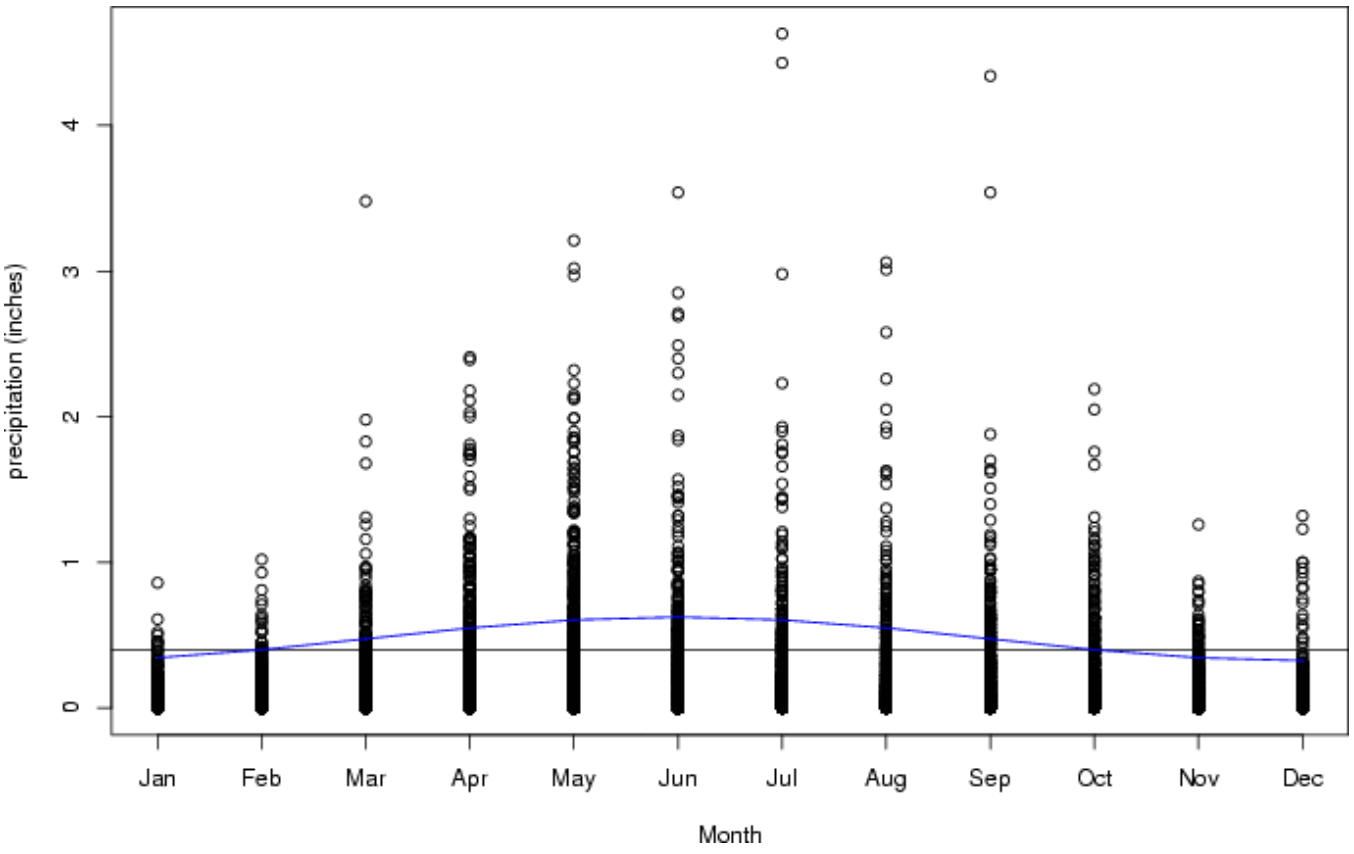


Figure 7.1: *Fort Collins, C.O. precipitation data with constant threshold of 0.4 inches (solid black line) and nonconstant (cyclic) threshold (solid blue line). Note that although the varying threshold appears to vary smoothly on a daily basis, the threshold used in the example is constant for each month.*

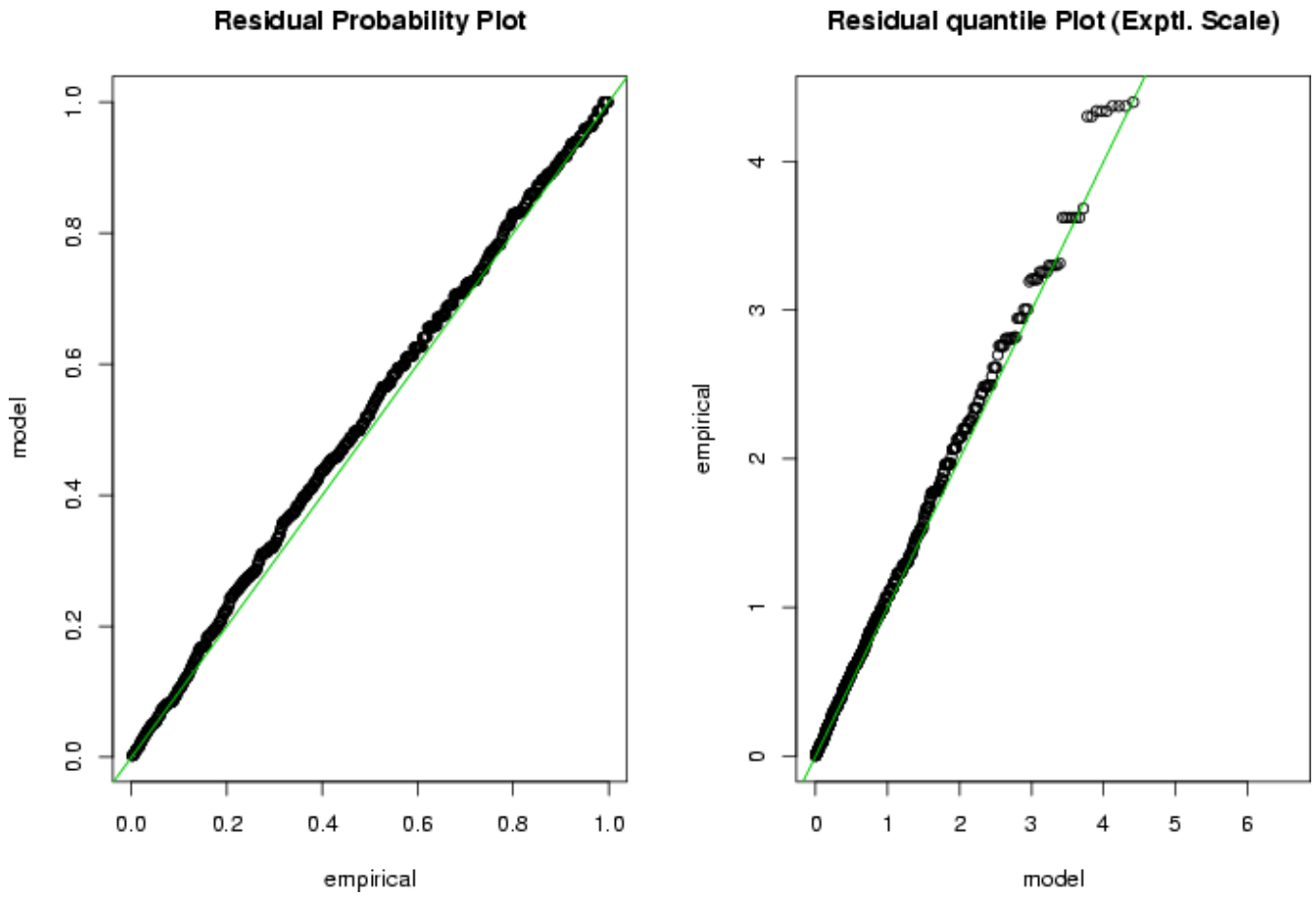


Figure 7.2: *Probability and quantile plots for fitting data to a point process model to the Fort Collins, C.O. precipitation (inches) data with a seasonal cycle incorporated into the threshold.*

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] Ch. 5 for more on this method and Ch. 9 for some alternatives to declustering.

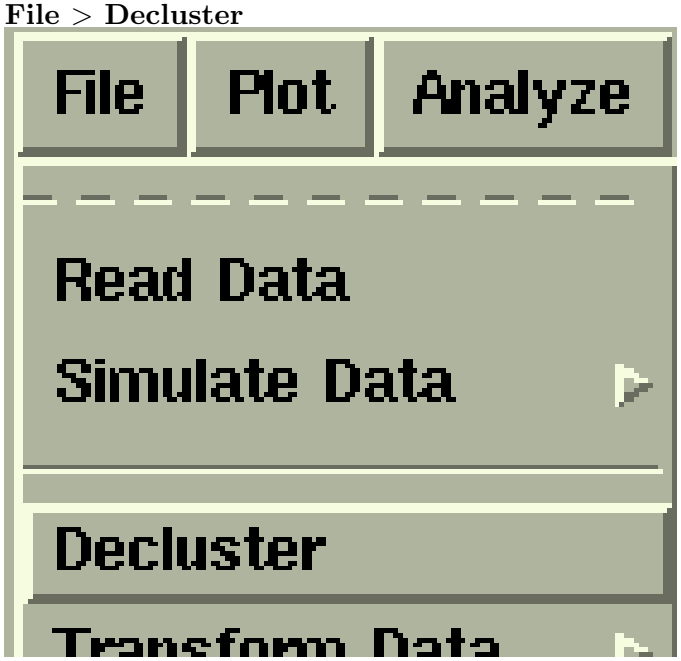
`extRemes` provides for declustering the data using runs declustering, but in practice declustering is a more involved process that should be executed by the user, and is not supported by `extRemes` itself. The general procedure for declustering data with the toolkit is as follows.

- **File > Decluster**
- *Select data from the* **Data Object** *listbox.*
- *Select the variable to decluster from the* **Variable to Decluster** *listbox.*
- *Optionally select the variable with which to "decluster by" from the* **Decluster by** *listbox.*
- *Enter desired threshold (or vector of thresholds) in the* **Threshold** *field.*
- *Enter a number for* **r** **> OK**.

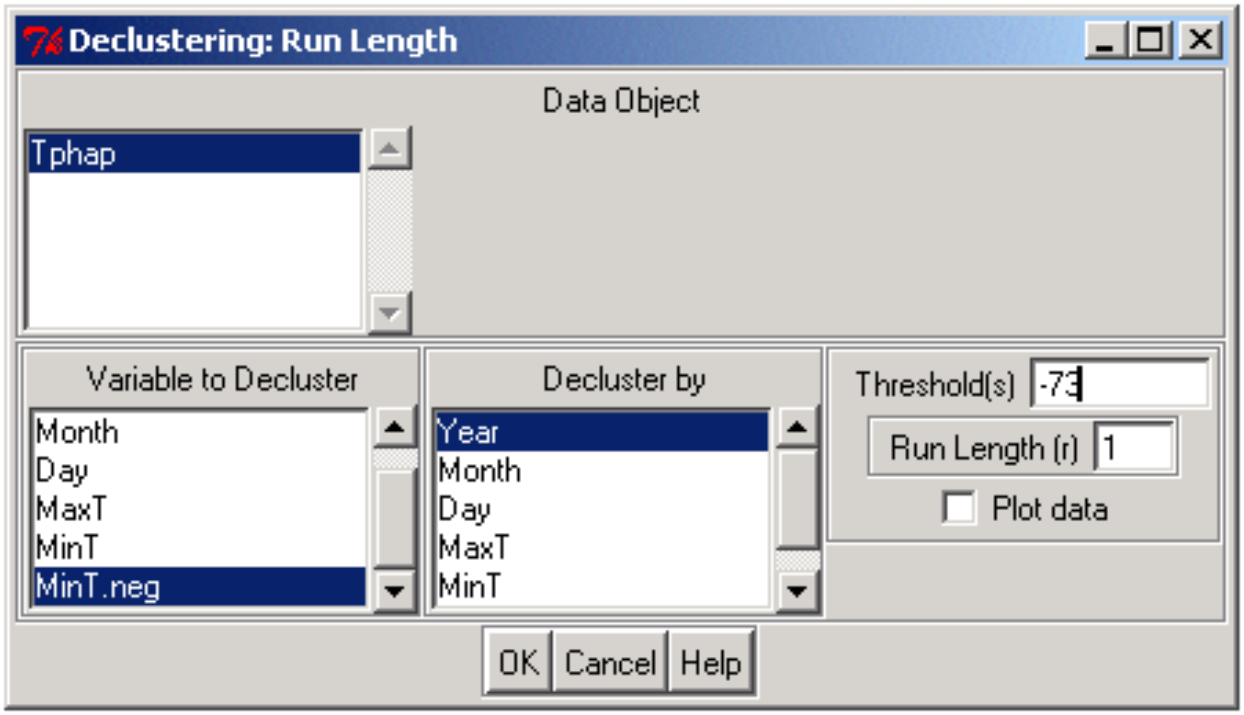
Example: Phoenix Minimum Temperature

To decluster the Phoenix minimum temperature (see section 6.0.16 Example 3) data using the toolkit (runs declustering), do the following.

- **File > Decluster**



- *Select* **Tphap** *from the* **Data Object** *listbox.*
- *Select* **MinT.neg** *from the* **Variable to Decluster** *listbox.*
- *Select* **Year** *from the* **Decluster by** *listbox.*
- *Enter* **-73** *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 at the cluster breaks by clicking on the **Plot data** checkbox. Here, however, (as is often the case) the amount of data and relatively large number of clusters creates a messy, illegible plot. Therefore, leave this box unchecked for this example. A message will be displayed on the main toolkit window that 84 clusters were found and that the declustered data were assigned to **MinT.neg.u-70r1dcbyYear**. This column has been added to the original data matrix using this name (where **u-70** corresponds to the threshold of -70 and **r1** corresponds to r being 1). Other information given includes two estimates of the extremal index. The first estimate is a simple estimate that is calculated after declustering is performed; referred to in the display as being estimated from runs declustering. Namely, the estimate is $\hat{\theta} = \frac{n_c}{N}$, where n_c is the estimated number of clusters and N is the total number of exceedances over the threshold, u. The second estimate is more complicated, but is made prior to declustering the data,

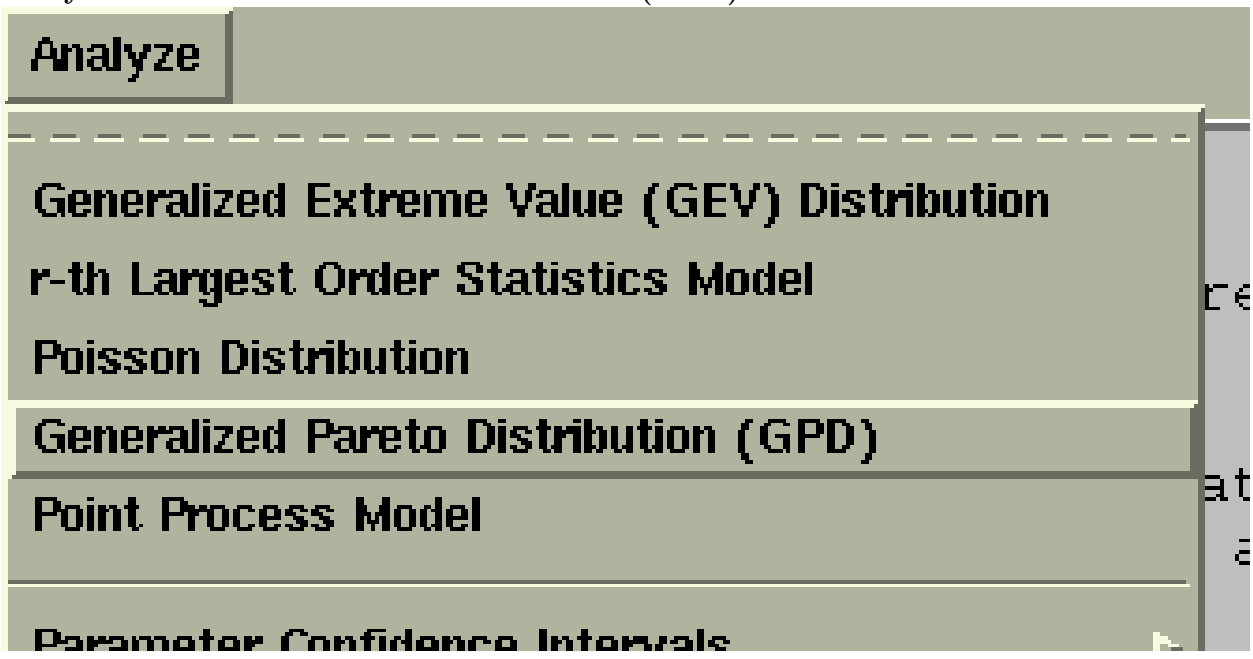
and is called the intervals estimator (Ferro and Segers [4]). Please see Appendix C for the definition of this estimate.

Other information given in the main toolkit dialog is a suggested run length based on the procedure of Ferro and Segers [4] of $r = 11$, but this number should be disregarded here because we are declustering by year. The procedure for determining the "best" run length employed with this software does not account for covariates when declustering. It is important to decluster by year here because we do not want values from August of one year to be clustered with values from July of the following year. If it were determined unnecessary to decluster by year, then $r = 11$ would still apply for declustering without taking into account the year.

Note that because this process reduces the number of data points, values below the threshold have been "filled in" so that the declustered data will have the correct dimensions in order to be added to the original data matrix. Specifically, every point not found to be a cluster maxima is converted to be the minimum of the data and the threshold–i.e., $\min(x, u)$. These *filled-in* values will not affect any POT analyses (using the same or higher threshold) because they are less than the threshold, and subsequently discarded. The original positions of the cluster maxima are preserved so that any covariates will not require further transformations. The optional use of the **Decluster by** feature ensures that, in this case, values from one year will not be clustered with values from another year.

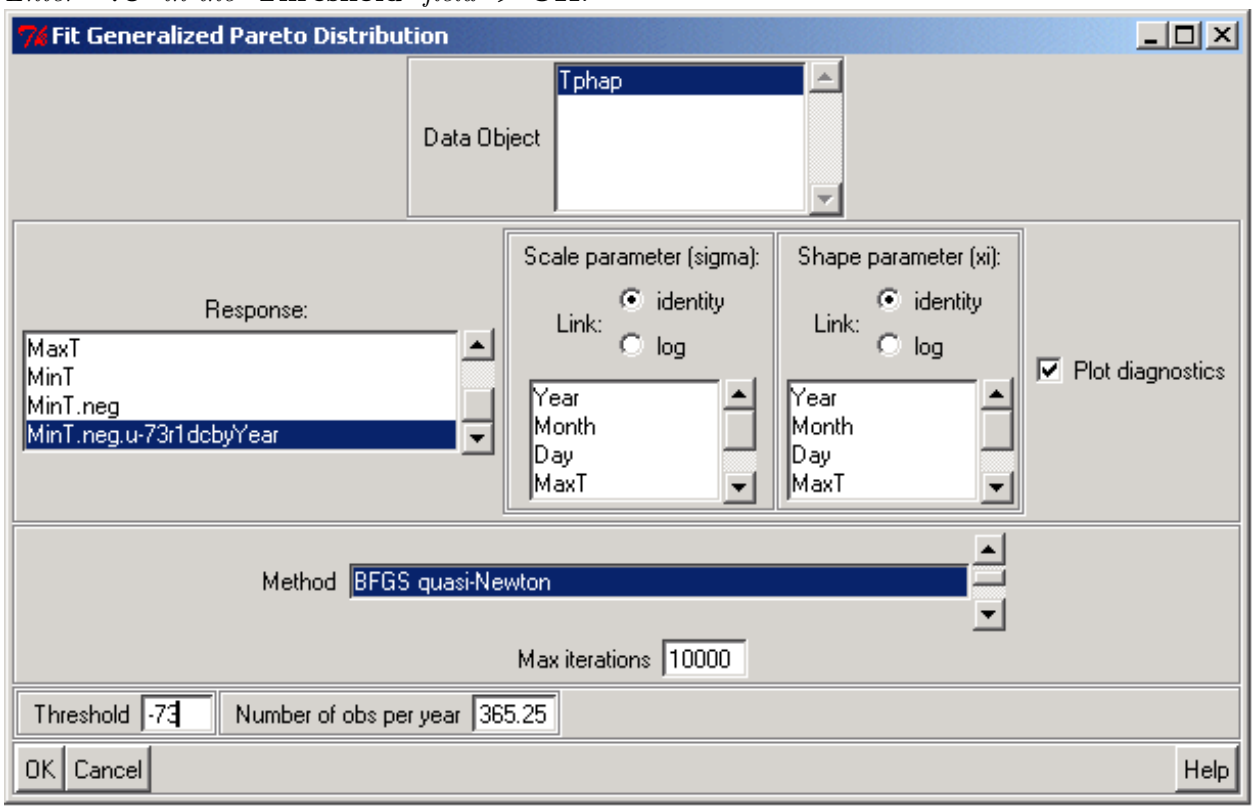
The next step is to fit the declustered data to a GPD.

- **Analyze > Generalized Pareto Distribution (GPD)**



- *Select* **Tphap** *from the* **Data Object** *listbox.*

- *Select* **MinT.neg.u-70r1dcbyYear** *from the* **Response** *listbox.*
- *Here, I optionally select* **BFGS quasi Newton** *from the* **Method** *listbox.*
- *Enter* **-73** *in the* **Threshold** *field* > **OK**.



One detail to be carefull about, in general, is that the number of points per year (**npy**) may be different once the data have 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. See Coles [3] Ch. 5 for an adjustment to the return level that accounts for the extremal index.

Results of fitting the GPD to these data are shown in Table 7.1. It is difficult to compare the models using the log-likelihoods here, but there does not appear to be much variability in parameter estimates from one model to the other suggesting that declustering is not important for these data. Particularly the 100-year return level estimates. Each estimate is within the 95% confidence bounds of every other estimate.

These minimum temperature data for Phoenix, A.Z. (Fig. 6.3), clearly have an upward trend over time, and possibly a varying standard deviation from one year to the next. The blue line in Fig. 6.3 is the least squares fit (using all data, and not just those points above a threshold) $\hat{Y}_t = \hat{\beta}_0 + \hat{\beta}_1 \cdot (t - 1948)$, which has a significant positive slope.

Declustering	None	r = 1	r = 2	r = 5
$\hat{\sigma}$	3.91 (0.303)	4.16 (0.501)	4.21 (0.540)	4.42 (0.620)
$\hat{\xi}$	-0.25 (0.049)	-0.24 (0.079)	-0.24 (0.086)	-0.25 (0.097)
100-yr r.l.	59.20	58.67	58.45	58.39
$\hat{\theta}$	1	0.57	0.53	0.44
$\tilde{\theta}$	-	0.20	0.20	0.20

Table 7.1: *Results of fitting data to the GPD to (negative) minimum temperature (degrees Fahrenheit) using a threshold of 73 degrees at Phoenix Sky Harbor airport with: no declustering, runs declustering with run length* $r = 1$ *(150 clusters), runs declustering with* $r = 2$ *(138 clusters) and runs declustering with* $r = 5$ *(115 clusters). Here,* $\hat{\theta}$ *is the extremal index estimated by* $\frac{n_c}{n}$*, and* $\tilde{\theta}$ *is the extremal index estimated as in Ferro and Segers [4].*

It is, therefore, of interest to also investigate incorporation of a trend into the GPD model. We will also make use of the Poisson-GP model of sections 6.0.16 and B.0.28, here. The fitted values for the Poisson rate parameter model $\log \lambda(t) = \lambda_0 + \lambda_1 t$, with $t = 1, \ldots, 43$ (**Year** $- 47$) are:

$$\hat{\lambda}_0 \approx -2.38 \ (0.160)$$

$$\hat{\lambda}_1 \approx -0.04 \ (0.008).$$

The likelihood ratio against the null model is approximately 26 with associated p-value near zero indicating that the inclusion of a temporal trend is statistically significant. The parameter estimates from fitting the GPD with $\log(\sigma) = \sigma_0 + \sigma_1 t$, t as above, are given by:

$$\hat{\sigma}_0 \approx 1.66 \ (0.161)$$

$$\hat{\sigma}_1 \approx -0.02 \ (0.008)$$

$$\hat{\xi} \approx -0.24 \ (0.069)$$

The likelihood ratio between this model and the model without any temporal trend is about 4.23, which is (slightly) greater than the $\chi^2_{1,0.95}$ critical value of 3.84, and the associated p-value of about 0.040; indicating that the inclusion of a temporal trend in the scale parameter is statistically significant, if only slightly. Of course, the apparent trend in Fig. 6.3 is relatively small, but nevertheless apparent so that it is reasonable to include such a trend in the Poisson-GP model.

Chapter 8

Details

8.0.20 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.

  ```
  > 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 8.0.22 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:

```
> 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:

```
> 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:

```
> install.packages( "ismev")
```

If this does not work, then it is likely 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( "ismev", lib="/home/[user]/src/library")
```

Once `ismev` is installed on your system, it needs to be loaded into R (see above).

8.0.21 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.")
   }
```

Note that many linux networks now use a different method for supplying R libraries, and the above `.libPaths` *may interfere and cause problems with this new paradigm.*

8.0.22 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:

`http://www.tcl.tk/doc/howto/compile.html`

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. Generally, Tcl/Tk will already be installed on unix/linux.

- 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.

  `> 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):
  `> 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 the next session. Better to upgrade to R version 1.7.0 (or greater) and not have to worry about it ever again.

8.0.23 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 verbatum 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.

8.0.24 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:

```
> foo$models$gpd.fit1 <- NULL
```

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 $\Pr\{M_n \leq u\} = \Pr\{X_i \leq u; \forall i = 1, \ldots, n\}$, 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, $F^n \longrightarrow 0$ as $n \longrightarrow \infty$. 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 $F^n(\frac{M_n - b_n}{a_n})$ leads to a nondegenerate distribution as $n \longrightarrow \infty$. Specifically, we seek $\{a_n > 0\}$ and $\{b_n\}$ such that $F^n(\frac{M_n - b_n}{a_n}) \longrightarrow G(z)$ 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.

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

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

II. Fréchet

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

III. Weibull

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

for parameters $\sigma > 0$, μ 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,

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

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.

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.0.25 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 μ, $\sigma > 0$ and ξ. Then for a large enough threshold, u, the distribution function of $(X - u)$, conditional on $X > u$, is approximately

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

defined on $\{y : y > 0 \text{ and } (1 + \frac{\xi y}{\tilde{\sigma}}) > 0\}$, 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.0.26 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* [15]. For more information on this approach, see Coles [3], Smith [16] and Smith [15].

B.0.27 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

$$\{(u, \tfrac{1}{n_u} \textstyle\sum_{i=1}^{n_u} (x_{(i)} - u)) : u < x_{\max}\}.$$

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

B.0.28 Poisson-GP Model

The parameters of the point process model can be expressed in terms of those of the GEV distribution or, equivalently through transformations specified below, in terms of the parameters of a Poisson process and the GPD (i.e., a Poisson-GP model). Specifically, given

μ, σ and ξ from the point process model, we have the following equations.

$$\log \lambda = -\frac{1}{\xi} \log[1 + \xi \frac{(u-\mu)}{\sigma}] \tag{B.1}$$

$$\sigma^* = \sigma + \xi(u - \mu) \tag{B.2}$$

where λ is the Poisson rate parameter, σ^* is the scale parameter of the GP and σ the scale of the point process model. Eqs. (B.1) and (B.2) can be solved simultaneously for σ and μ to obtain the parameters of the associated point process model (Katz *et al* [9]). Specifically, solving Eqs. (B.1) and (B.2) for μ and σ gives the following.

$$\frac{\sigma^*}{\sigma} = \frac{\sigma + \xi(u-\mu)}{\sigma} = 1 + \xi(\frac{u-\mu}{\sigma}) \Rightarrow$$

$$\log \lambda = -\frac{1}{\xi} \log(\frac{\sigma^*}{\sigma}) \Rightarrow$$

$$\log \sigma = \log \sigma^* + \xi \log \lambda \tag{B.3}$$

$$\mu = u - \frac{\sigma}{\xi}(\lambda^{-\xi} - 1) \tag{B.4}$$

The block maxima and POT approaches can involve a difference in time scales, h. For example, if observations are daily ($h \approx 1/365$) and annual maxima are modelled, then it is possible to convert the parameters of the GEV distribution for time scale h to the corresponding GEV parameters for time scale h' (see Katz *et al.* [10]) by converting the rate parameter, λ, to reflect the new time scale.

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

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.

When data are independent and identically distributed we have that $\Pr\{M_n \leq u_n\} = F(u_n)^n$, but if there is dependence, then we still have that $\Pr\{M_n \leq u_n\} = F(u_n)^{\theta n}$, where $\theta \in (0, 1]$ is called the extremal index (see O'Brien [11]). If the data are independent, then $\theta = 1$; but the converse is not true (see, for example, pg. 97 of Coles [3]). Similarly, as $\theta \longrightarrow 0$ the data are said to be perfectly dependent.

Ferro and Segers [4] present several estimates for the extremal index. The one they suggest as the "best" (and is used by this toolkit) is defined as

$$\tilde{\theta} = \begin{cases} \min\{1, \frac{2(\sum^{N-1} i{=}1 T_i)^2}{(N-1)\sum_{i=1}^{N-1} T_i^2}\}, & \text{if } \max\{T_i : 1 \leq i \leq N-1\} \leq 2 \\ \min\{1, \frac{2(\sum_{i=1}^{N-1}(T_i-1))^2}{(N-1)\sum_{i=1}^{N-1}(T_i-1)(T_i-2)}\}, & \text{if } \max\{T_i : 1 \leq i \leq N-1\} > 2, \end{cases} \tag{C.1}$$

where T_i are the interexceedance times (the length between exceedances).

C.0.29 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 $Z_t \sim$GEV$(\hat{\mu}(t), \hat{\sigma}(t), \hat{\xi}(t))$ then the standardized variables,

$$\tilde{Z}_t = \frac{1}{\hat{\xi}(t)} \log\{1 + \hat{\xi}(t)(\frac{Z_t - \hat{\mu}(t)}{\hat{\sigma}(t)})\}, \tag{C.2}$$

each have the standard Gumbel distribution with probability distribution function

$$P\{\tilde{Z}_t \leq z\} = \exp\{-e^{-z}\}, z \in \Re. \tag{C.3}$$

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

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

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

$$\tilde{Y}_t = \frac{1}{\hat{\xi}(t)} \log\{1 + \hat{\xi}(t)(\frac{Y_t - u(t)}{\hat{\sigma}(t)})\} \tag{C.4}$$

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

$$\tilde{Y}_t = 1 + \hat{\xi}(t)(\frac{Y_t - u(t)}{\hat{\sigma}(t) + \hat{\xi}(t)(u - \hat{\mu}(t))})^{-1/\hat{\xi}(t)} \tag{C.5}$$

is employed and the probability plot consists of the pairs

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

and the quantile plot consists of the pairs

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

Index

Bibliography

[1] Balling, R.C., Jr., Skindlov, J.A. and Phillips, D.H. The impact of increasing summer mean temperatures on extreme maximum and minimum temperatures in phoenix, arizona. *Climate*, 3:1491–1494, 1990.

[2] Coles, Stuart. S-plus functions for extreme value modeling: An accompaniment to the book *an introduction to statistical modeling of extreme values*. http://www.stats.bris.ac.uk/ masgc/ismev/uses.ps.

[3] Coles, Stuart. *An Introduction to Statistical Modeling of Extreme Values*. Springer-Verlag, London, 2001.

[4] Ferro, C.A.T. and Segers, J. Inference for clusters of extreme values. *J.R. Statist. Soc. B*, 65(2):545–556, 2003.

[5] Fuentes, Montserrat Statistical assessment of geographic areas of compliance with air quality. *Journal of Geographic Research–Atmosphere*, 2003, 108(D24).

[6] Gilleland, Eric and Nychka, Doug Statistical models for monitoring and regulating ground-level ozone. *Special Issue: Environmetrics*, 2005 (in press).

[7] Katz, Richard W. Stochastic modeling of hurricane damage. *Journal of Applied Meteorology*, 41:754–762, 2002.

[8] Katz, Richard W. and Parlange, Marc B. Generalizations of chain-dependent processes: Application to hourly precipitation. *Water Resources Research*, 31(5):1331–1341, 1995.

[9] Katz, Richard W., Parlange, Marc B. and Naveau, Philippe. Statistics of extremes in hydrology. *Advances in Water Resources*, 25:1287–1304, 2002.

[10] Katz, Richard W., Brush, Grace S. and Parlange, Marc B.. Statistics of extremes: modeling ecological disturbances. *Ecology*, 2005 (in press).

[11] O'Brien, George L.. Extreme values for stationary and Markov sequences. *The Annals of Probability*, 15 (1):281–291, 1987.

[12] Pielke, Roger A. and Downton, Mary W. Precipitation and damaging floods: Trends in the united states, 1932-97. *Journal of Climate*, 13(20):3625–3637, 2000.

[13] Pielke, Roger A. and Landsea, CW. Normalized hurricane damages in the United States: 1925-95. *Weather and Forecasting*, 13(3):621–631, 1998.

[14] R Development Core Team, R: A language and environment for statistical computing. http://www.R-project.org, *R Foundation for Statistical Computing*, ISBN 3-900051-00-3, Vienna, Austria, 2003.

[15] Smith, Richard L. Statistics of extremes with applications in environment, insurance and finance. http://www.stat.unc.edu/postscript/rs/semstatrls.ps, 2002.

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

[17] Tarleton, Lesley F. and Katz, Richard W. Statistical explanation for trends in extreme summer temperatures at Phoenix, A.Z. *Journal of Climate*, 8(6):1704–1708, 1995.

[18] Wettstein, Justin J. and Mearns, Linda O. The influence of the north atlantic-arctic oscillation on mean, variance and extremes of temperature in the northeastern United States and Canada. *Journal of Climate*, 15:3586–3600, 2002.