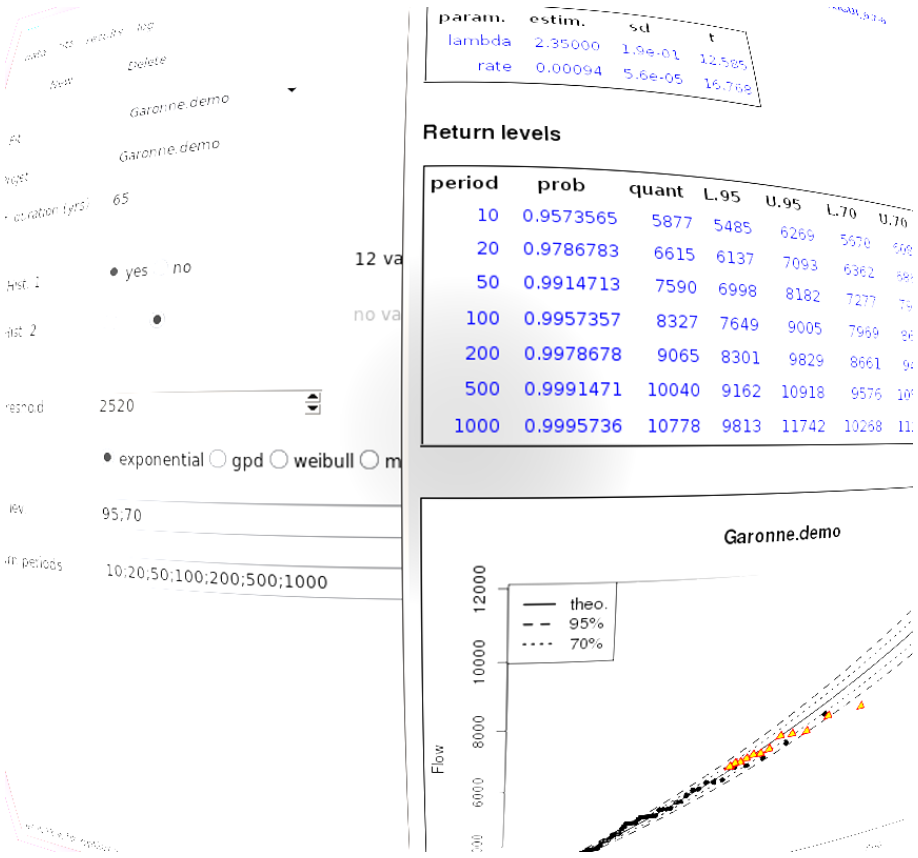
RenextGUI
installation and use

RenextGUI 1.0-0. March 12, 2012



Yves Deville
Lise Bardet
Claire-Marie Duluc

Contents

Chapter 1

Introduction

This document was written with **RenextGUI 1.0-0** and **Renext 2.0-0**

1.1 Goals

The **RenextGUI** package has been developed by the French *Institut de Radioprotection et de Sûreté Nucléaire* (IRSN). As its name might suggest, it provides a Graphical User Interface (GUI) for[1] the **Renext** package, which was also developed by the IRSN. These softwares are used with the R system which is part of the **R project** [10].

A recent version of R must be installed on the computer on which the packages are planned to be run.

The main goal is to allow the fitting of Peak Over Threshold (POT) models in an interactive fashion, with a possibly limited knowledge of the R language. Not all functions of **Renext** can be used with the GUI. The interface is mainly devoted to the *fitting* step in POT and provides no functionality for the analysis of the events in time.

The main functionalities are the following.

- *Import data* from csv[2] files and *specify auxiliary data*: duration, historical data.
- *Choose the fitting options*: the *threshold* and the *distribution* for the exceedances model, as well as some output options.
- Display the *parameters and the predictions* as *text*, *tables* and *return level plots* as shown in figure 1.1.
- *Export the results* or predictions as a HTML document if the package **R2HTML** has been installed.

The `RenextGUI` package has only one exported function, hence the package manual is reduced.

Note that the **extRemes** CRAN package [5] provides a GUI for some of the extreme values modeling functions described in Stuart Coles' book [1] and implemented in the **ismev** package [3]. The two GUIs' functionalities intersect with POT models having a GPD distribution and no historical data.

[1] Some functions of.
[2] Comma Separated Variables

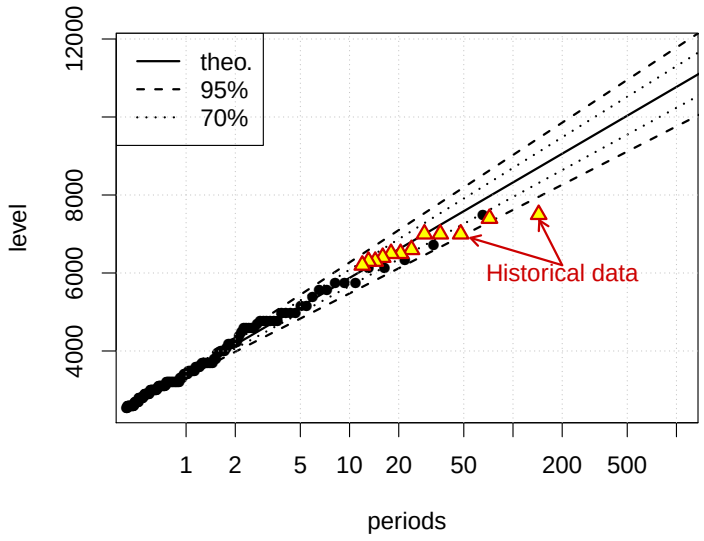


Figure 1.1: A *return level plot* with historical data. The x-axis uses a log-scale and ticks are in years.

1.2 Required packages

Packages

The **RenextGUI** package requires some other packages to work. Some are required to build the interface, others to fit the models or to export the results in HTML format.

R packages can be installed following different methods described in the *R Installation and Administration* manual available at CRAN sites. Provided that the user has open access to CRAN sites, any CRAN package can be installed from the R command line using the `install.packages` function. For instance the CRAN package named `Rcmdr` can be installed from the R command line by

```
R> install.packages(pkgs = "Rcmdr", dependencies = TRUE)
```

The optional argument `dependencies` will install all needed or useful[3] packages.

GUI packages

The present version is based on the **gWidgets** package by John Verzani [9], which provides a common interface to several toolkits. Besides the **gWidgets** package itself, one of the "**gWidgets**xxx" packages are required, which in turn requires some other packages

gWidgetsxxx	based on	requires
gWidgetstcltk	tcl-tk	**tcltk**, **tcltk2**, **digest**
gWidgetsRGtk2	Gtk2	**RGtk2**

[3]Those declared as depend, suggest and import in the package description.

see [12], [7] and [6] for the powerful **RGtk2** package by Michael Lawrence and Duncan Temple Lang. Other toolkits exist, but they have not been tested and should not be expected to work directly. The GUI will have a similar yet different look in the two cases. Some slight differences also exist in the widget behavior, e.g. for validation. Most of the screenshots appearing in this document have been made while running the `tcltk` version.

Depending on the installed GUI packages, the interface can be launched with an optional argument specifying the toolkit to use, see 2.2 later. For instance

```
R> RenextGUI("RGtk2")
```

starts a GUI based on the `RGtk2` toolkit. The default choice for the argument with name `guiToolkit` is `"tcltk"` since the GUI packages are easier to install.

The `"RGtk2"` option based on the package **gWidgetsRGtk2** (and hence **RGtk2**) is better, and should be chosen by most Unix users. For Windows users, however, installing the package **RGTk2** can be tedious at the time.

Renext

The **Renext** package [2] can be obtained from the CRAN sites. However, not all versions of **Renext** are submitted to the CRAN, and it might be necessary to download a fresh version from the IRSN gforge site

`https://gforge.irsn.fr/gf/project/renext`

Renext contains the fitting and plotting functions, but it also uses some other packages from the CRAN: the **evd** package by Alec Stephenson [11] is required, as well as the **numDeriv** by Paul Gilbert [4].

It may be useful to have a look to the pdf document named *RenextGuide.pdf* accompanying **Renext** notably for questions about historical data.

R2HTML

The **R2HTML** package by Éric Lecoutre [8] is *suggested*. It will be needed to export data.frames in HTML format, if wanted. This package is available from the CRAN and is readily installed.

Chapter 2

Using RenextGUI

2.1 General concepts

A typical use of the GUI consists in interactively fitting a POT model through the following steps.

1. Define a **project**: specify the location of a file where to read the main sample X_i, give the effective duration, enter historical data, ...
2. Choose the options for the **fit**. These are: the threshold u, the distribution for the exceedances $Y_i := X_i - u$, ...
3. Inspect the **results**: estimated parameters, predictions, return level plot. Go back to step 2. if needed.
4. Copy/paste the results. When needed, the results can be exported in HTML. The data and fit options can also be sealed within a R program file or "R dump" that can be used later to reproduce the results.

As explained later, there is no dynamic linking between the project, the fit and the results: changing a project will not update existing results for that project.

2.2 Starting the GUI

Once **RenextGUI** and the required packages have been installed, the GUI is launched from a R command line. This is done by loading the package using `library`, and then simply calling the eponymous function. However

the `warn` option should be given a positive value

otherwise, problems met during the fitting step (e.g. failed convergence) will remain concealed. Thus, launching the GUI can be done as follows

```
R> libary(RenextGUI)
R> options(warn = 1)
R> RenextGUI()
```

A window with title `RenextGUI` then opens, as at the left of figure 2.1. This window will be referred to as the *main window*. At the very bottom, a *status bar* is used to display some information or a message.

The required packages will be loaded. Note that the optional package **R2HTML** will be loaded when needed, i.e. when exporting is required. If this packages is not available, a message at the R console will inform the user that the export is not possible.

For Windows users, the R command line is RGUI. It might be convenient to minimize the GUI window in order to keep the **RenextGUI** windows in foreground position.

By default, the toolkit used is `"tcltk"`. If the `"RGtk2"` toolkit is to be used, it must be given as an argument of the function, see 1.2 above.

2.3 Control frames

RenextGUI's main window appears as a notebook with four tabs (see left of figure 2.1).

- **data** is dedicated to the import and control of data or *projects*.
- **fits** chooses the fitted model and the fitting threshold.
- **results** displays the estimated parameters and predictions.
- **log** contains nothing for the moment, but could in the future provide some info about fitting or error.

Within each of the notebook tabs, buttons are found with straightforward meaning. For instance, the Delete button will obviously be used to remove data, projects, fits or results. If data has been imported from a csv file, using the Delete button of the `data` tab will of course not delete the data file but only remove the read dataset from the list of those available for fits.

2.4 Caution

The GUI makes use of global variables, i.e. of R objects set or modified in the global environment. Among these are the lists objects named `dataList`, `fitsList` and `resList`. They contain named elements with name corresponding to those appearing in the interface, e.g. `Garonne.demo`. Of course, these objects must not be modified programmatically while running the GUI. It is safe to remove all existing objects before starting the GUI, or to start a new R session.

Do not use two GUIs in the same R session!

Note that there is no dynamic linking between a *project* and the subsequent fits and results using it. Thus

Modifying a project will NOT change the fits or results that could exist before!

Deleting the project/fit and results might be preferable whenever any doubt exists about the data really used.

2.5 Example: using the `Garonne.demo` project

Data and project

As a first example, we make use of an existing project with name `Garonne.demo`. The project is built with the dataset `Garonne` which is shipped with the **Renext** package and concerns measurements of *La Garonne* river flow. This dataset is also used as a demo example in **RenextGUI**.

Choose/edit an existing project

The following steps are illustrated on figure 2.1.

1. Activate the **data** tab of the main window, and choose `Garonne.demo` in the Project combobox. Some of the fields are now filled with values attached to the project. See the right panel of figure 2.1.
2. Use the Plot button to get a simple plot of the data. The main OT (*Over Threshold*) sample data is shown using vertical "needles" at events as shown in figure 2.2.
3. Change the **Main title** of the plot using the relevant text entry, and click again on Plot.

Other fields concern the following data.

- The *effective duration* (here 65 years) is an information which should normally be available for such data. It is the duration (in years) of the whole period, ignoring the possible gaps in the measurement. See the **Renext** package's vignette.
- *Historical data* represent maximum levels recorded during an *historical period* and providing an information independent from that carried in the OT sample. The duration of the historical period *must* be provided. Here, we have an historical period of about 143 years (non overlaping the OT period) and we know the 12 largest flows: 7500 m^3/s, 7400 m^3/s, and so on: see right panel of figure 2.1.

The effective duration and historical data fields are *editable*: they could be used to modify the existing project. Note the format used here for the vector: numeric values separated by the semi-colon `;`. A similar representation is used elsewhere for similar numeric vectors. The OK button would then be clicked to validate the choices, and the edited (new) values would then become attached to the project name.

Fitting a model

Projects that have been defined become available for a POT fit. We now describe the process of fitting. The steps are illustrated on figure 2.3.

1. In the main window, activate the **fits** tab. See fit 1 at the left of figure.
2. Click on the button labelled New. A dialog box appears with title *New fit*. See fit 2a at top right of figure.
3. In the dialog box, choose the `Garonne.demo` project, see fit 2b. You can change the effective duration (without impacting that of the dataset) and change the name of the fit.

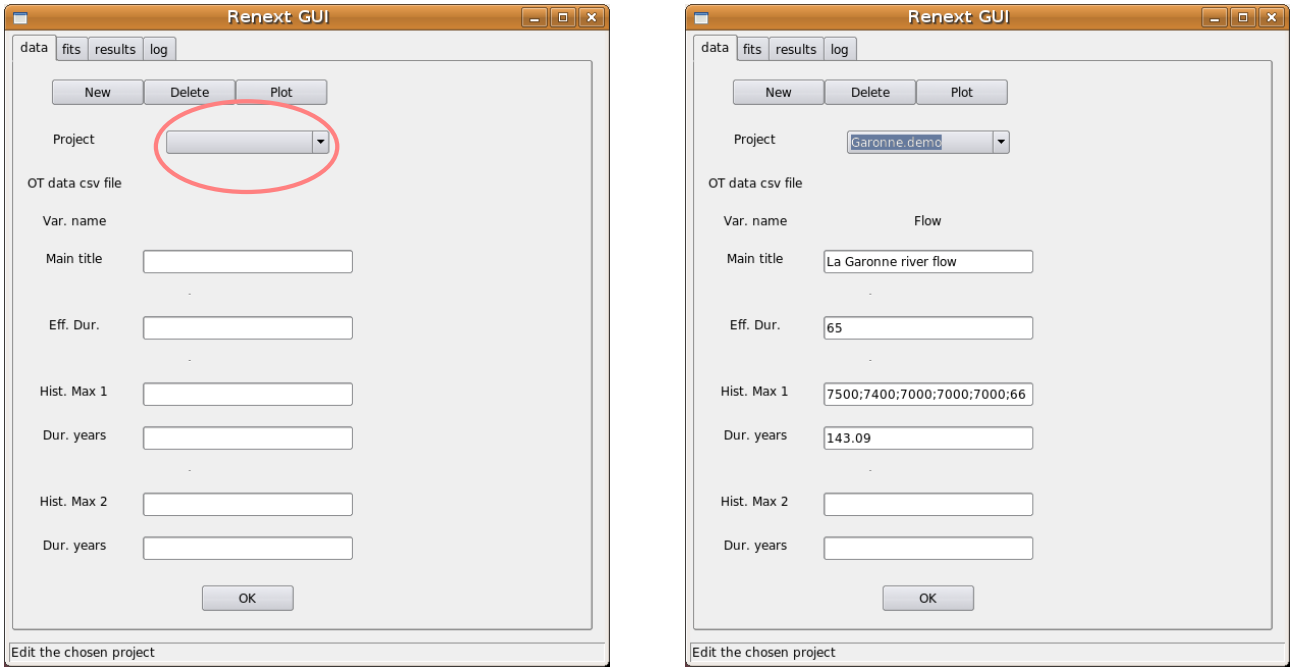


Figure 2.1: Choosing the existing project `Garonne.demo`. Once the project is selected with the Project combo box (left panel), some fields are filled (right panel).

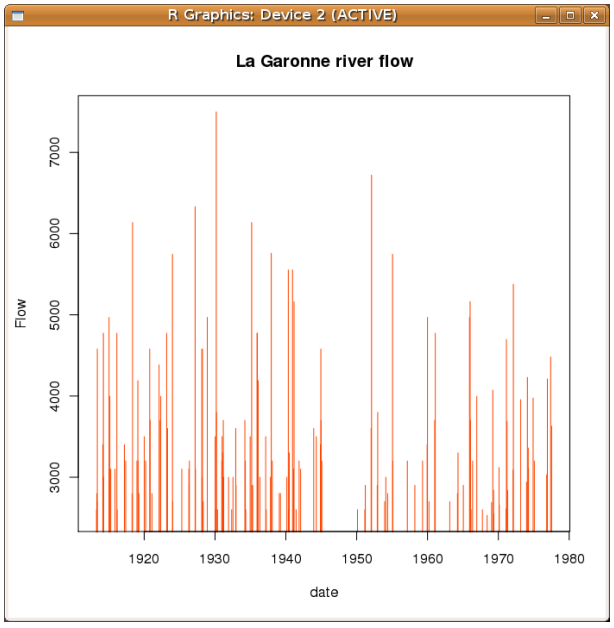


Figure 2.2: Using the Plot button of the `data` tab produces a time plot in a "R Graphics" window.

4. Change the name of the Fit in the dialog box, for example by typing `GaronneFit`, see fit 2c.

   Use only a name that could be used as an object name within R: no embedded white space, no accented characters, and so on.

5. Click on the OK button in the dialog box (see fit 2d), thus closing the dialog. A new fit has been initialized with name `GaronneFit`.

6. Using the Fit combobox in the main window and **fit** tab, select the newly created fit with name `GaronneFit`. Default values are displayed in the fields. Some are editable, others not.

7. Now click on the Fit button (bottom right of the main window). A graphics is drawn in a new window showing the *return level plot*.

   A (slightly different) plot can be redrawn from the results *page.*

As might be expected, it is possible to change the settings and then re-fit the model. For instance:

- The Fit threshold "spin button" can be used to set the POT threshold to one of the preselected values chosen in the OT data range. Another value can be given by typing within the editable field.

- Setting the radio box labelled Use Hist. 1 to `"no"` will refit the same model, but *without using the historical dataset 1*, which will no longer be shown on the return level plot.

Note that it can happen that the estimation fails, for instance if the model is not the well suited to the data.

Check the messages at the R console if necessary.

Use results

The results consist in a summary text and tables stored in a notebook, see figure 2.5. Each notebook tab has a label identical to the fit name. Thus refitting after a modification of the fit options changes the results displayed within the existing tab, but does not produce a new tab.

The content of one notebook tab is the following.

- A summary text recalling some characteristics of the dataset and with a few statistical results such as a Kolmogorov-Smirnov test.

- A table with the estimated values of the coefficients with their (approximated) standard deviation and the "t" values.

- A table with the "predictions", i.e. the return levels and their confidence limits. The return periods are those displayed in the fit tab of the main window, and can be set by the user.

Note that the rate of the Homogeneous Poisson Process is named `lambda`. This parameter has the physical dimension of an inverse time and is expressed in *events by year*. The parameters for the distribution of the exceedances have understandable names. Again, the user can refer to the

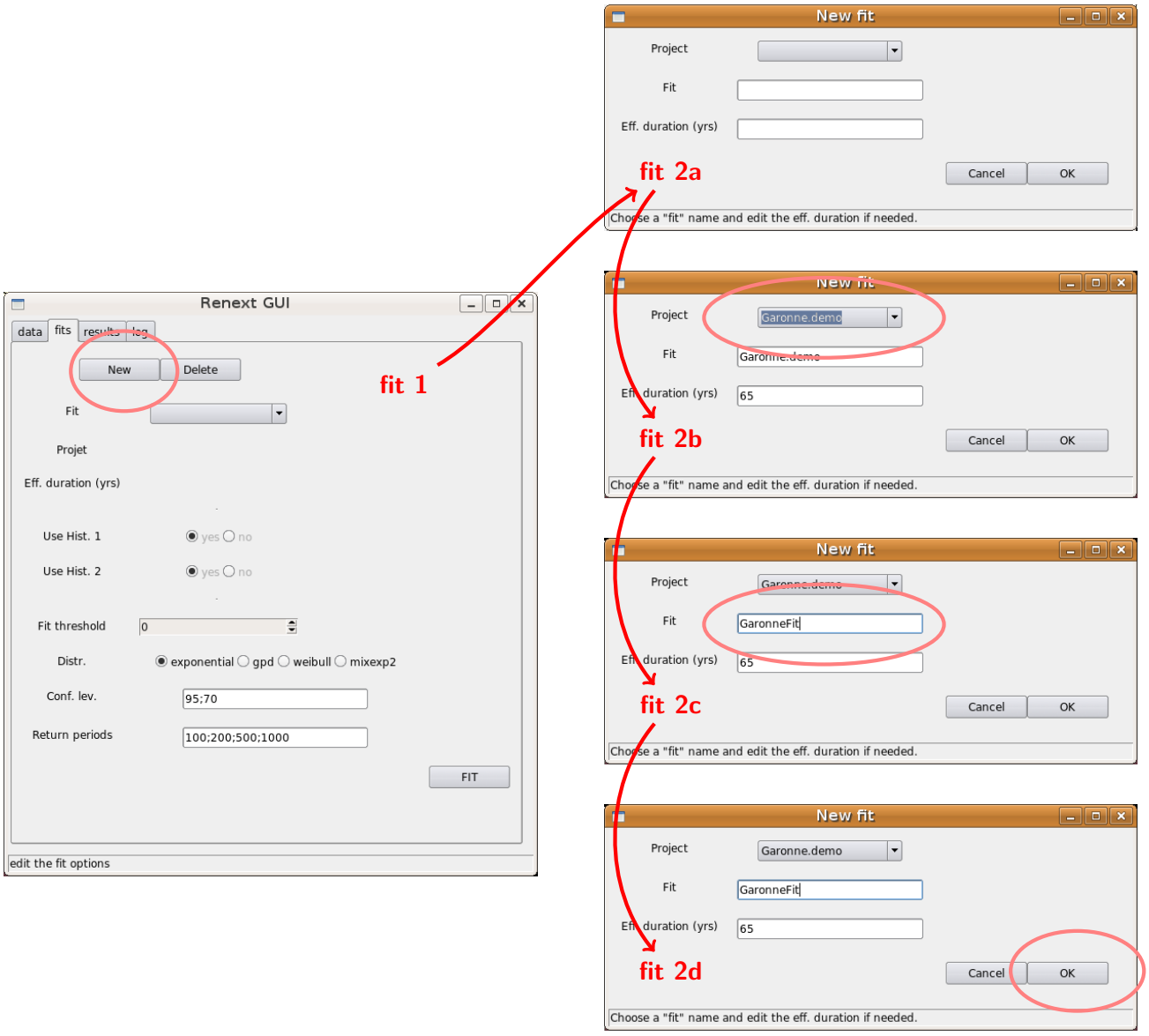


Figure 2.3: Defining a new fit using the `Garonne.demo` project, and with fit name `GaronneFit`. In the main window (left panel), the New button opens the dialog box **New fit** at the right (fit 2a). In this dialog box, the project is selected (fit 2b). Then the fit name can be edited (fit 2c), before clicking on OK (fit 2d) to close the dialog.

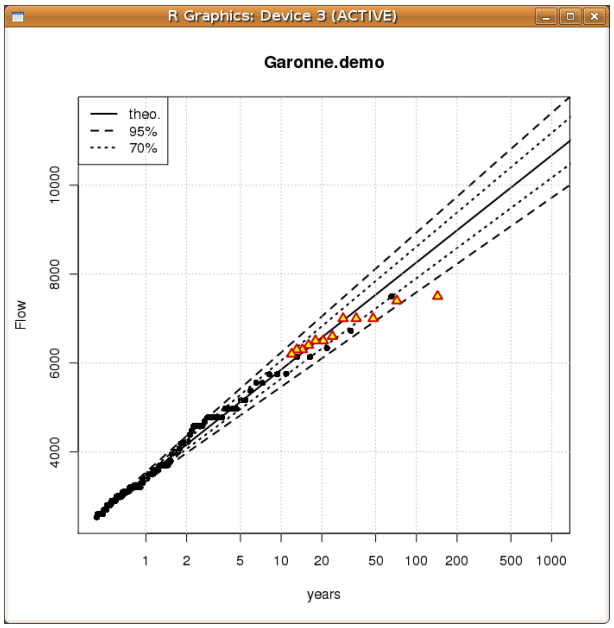


Figure 2.4: Using the Plot button of the `results` tab produces a return level plot in a "R Graphics" window.

vignette of the **Renext** package to learn more about the fitted POT models or about the estimation or inference methods used.

A return level plot can be drawn using the Plot button as shown on figure 2.4. The plot title, axes labels and axes limits can be set with a dialog open with the Options button (see later).

If a fit fails (divergence in the optimisation), it might be necessary to remove the (empty) results.

2.6 Reading from file

Csv files

Of course, the POT fitting with **RenextGUI** is not limited to the demo datasets which come with the package. In the general case, a new project uses data read in a csv file.

A dialog allows the user to select the file to read. The default directory is the *working directory* of R. Therefore, it may be convenient to set this to a place where data files and reports can be stored in subdirectories. This is done using the `setwd` function as in

```
R> setwd("/home/me/potfits")
```

The working directory must be set *before* launching the GUI.

The minimal requirement for a csv file is that the observations X_i are found in a column in the file. The column delimiter defaults to a semi-column but can be changed if needed. Similarly, the decimal separator defaults to the point `"."` but can also be a comma `","`. The file can optionally contain a header, that is a line located just before the first data line and containing column names separated by the column delimiter. Also optionally, there can be lines that must be skipped, e.g.

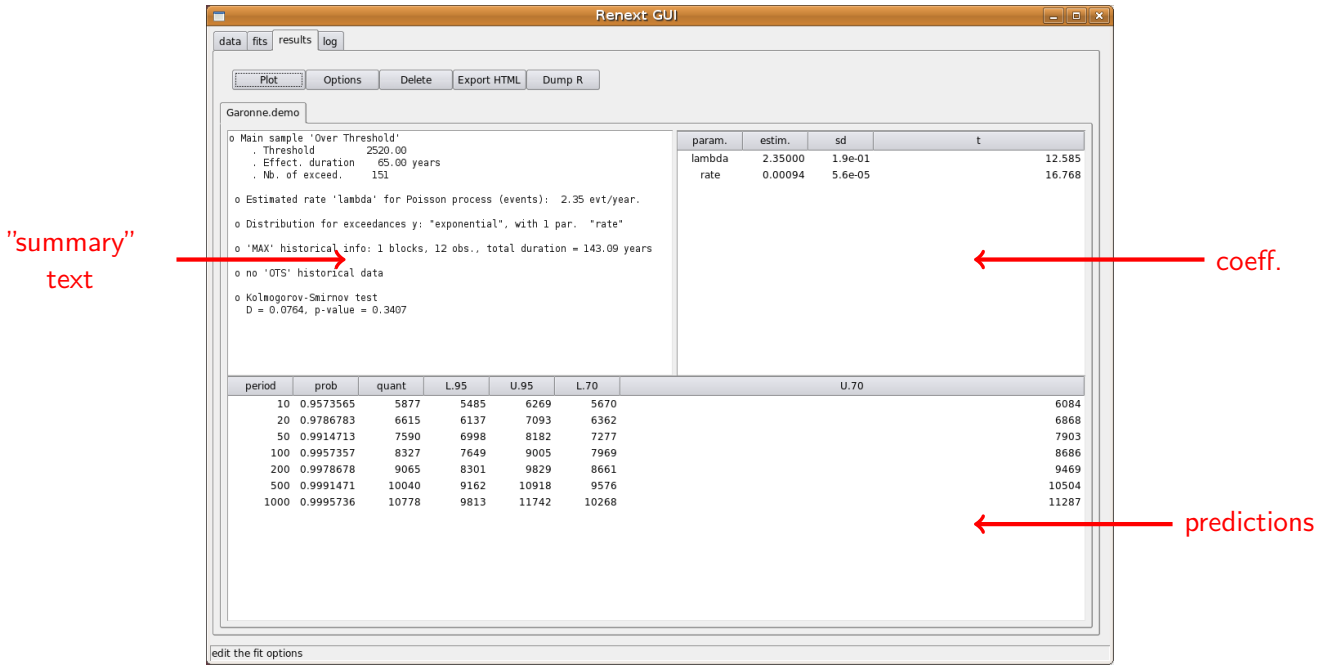


Figure 2.5: The results notebook tab embeds another notebook with one tab by fit. Here only one tab exists corresponding to the fit named `GaronneFit`.

because they describe the source of the data or other information. Their number can be given by the user.

An optional yet very useful information is a *date column*. This contain t_i in correspondence with the observations X_i. Although the preferred format is the POSIX `"%YYYY-%mm-%dd"`, some other date formats are possible like `"%dd/%mm/%YYYY"`. Using the given column delimiter and decimal separator, the program tries to guess which column can be used as date t_i and which can be used as sample X_i. See table 2.1 for examples of usable files.

If an acceptable choice of column(s) and format is provided or found, the OK button becomes active and can be used to validate the file reading options. Once read, the data can be displayed by using the Plot button. If a date column is used, the plot is a time plot $[t_i, X_i]$ as shown at the left of figure 2.2; else the plot uses the points $[i, X_i]$.

Example

As an example, we will use a csv file with first lines shown as column **file 1** in table 2.1. The variable name `surge` is found in the header (first line) and a date column is found. An example file can be created by simply copying the first lines into a new text file.

A new project is created by the following steps illustrated on figures 2.6 and 2.7.

1. Activate the data tab in main window *Renext GUI*, as in figure 2.6 read 1.

2. Click on the button labelled New, which opens a dialog window with name *csv File Import Wizard* read 2a.

	file 1	file 2	file 3

first lines	`date;surge`
	`1956-11-27;67.862`
	`1956-12-03;30.859`
	`1957-01-12;51.839`
	more lines
		`this is a comment`
		`27/11/1956;67,862`
		`03/12/1956;30,859`
		`12/01/1957;51,839`
		more lines
			`67.862`
			`30.859`
			`51.839`
			`30.832`
			more lines

parameters	`sep = ";"`
	`dec = "."`
	`header = TRUE`
	`skip = 0`
		`sep = ";"`
		`dec = ","`
		`header = FALSE`
		`skip = 1`
			`dec = "."`
			`header = FALSE`
			`skip = 0`

Table 2.1: First four lines for three possible csv files. For example files 1 and 2, a date is available in the respective formats `%YYYY-%mm-%dd` and `%dd-%mm-%YYYY`. The parameters are formal arguments of the `read.table` function which is used by the GUI.

3. In the *csv File Import Wizard* window, click on the browse button (figure 2.6 read 2a), and then select the wanted file to read from. Once the file is selected, some fields are filled as in figure 2.7 read 2b.

   The first lines read are shown. If a date column is found, it is selected for use, and its format is displayed. The default variable name and default project name appear in the corresponding fields.

4. Change the *project* and *variable* names, if wanted, by using the text edit widgets (figure 2.7 read 2c).

5. Click on OK (figure 2.7 read 2c) thus closing the window.

When the user changes controls such as the column delimiter (field sep), the decimal separator (dec sep), ... the file is read and the possible choices for the date and the variable columns are refreshed. This must be done if the default choices are not suitable for the chosen file.

Once the data have been read, it is possible to edit the newly created project from the data tab of the main window by the effective duration, adding some historical data, and so on. As exposed before in **Choose/edit an existing project**, section 2.5, this is done by selecting the project within the combobox, and then by using the text edit fields and validating choices by clicking the OK button. The project data can be plotted using the Plot button.

The newly created project is available for selection in the fits tab as exposed above in **Fitting a model** of the section 2.5.

2.7 Customizing graphics

As mentioned before, the Plot button of the results tab draws a return level plot with defaults settings: title, axis limits and labels. The Options button allows the user to change the graphics parameters interactively. It opens the dialog window shown in the left panel of figure 2.8.

Note that the chosen parameters have effect only when the return level plot is redrawn using the Plot button. When the fit is modified FIT button in the fits tab, all graphical parameters are re-initialized to their default values.

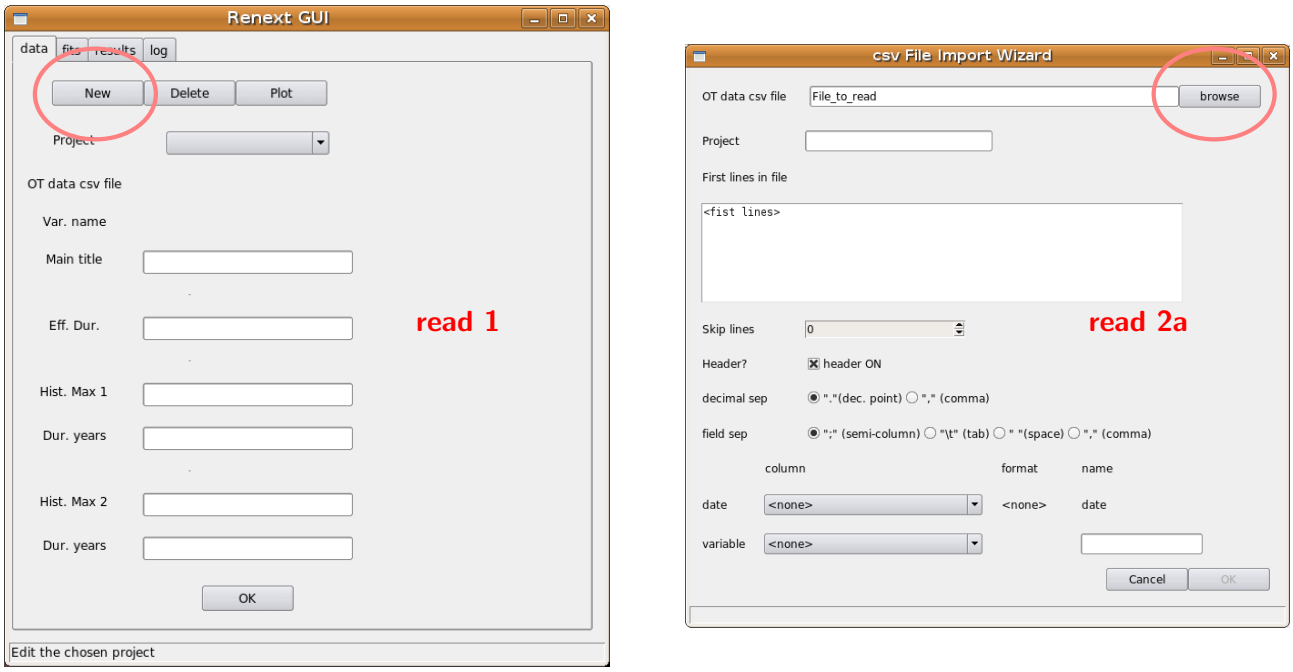


Figure 2.6: Defining a new project with data read from a csv file. When the the tab **data** is activated in the main window, a click on the New button (left panel) opens the window *csv File Import Wizard* (right panel).

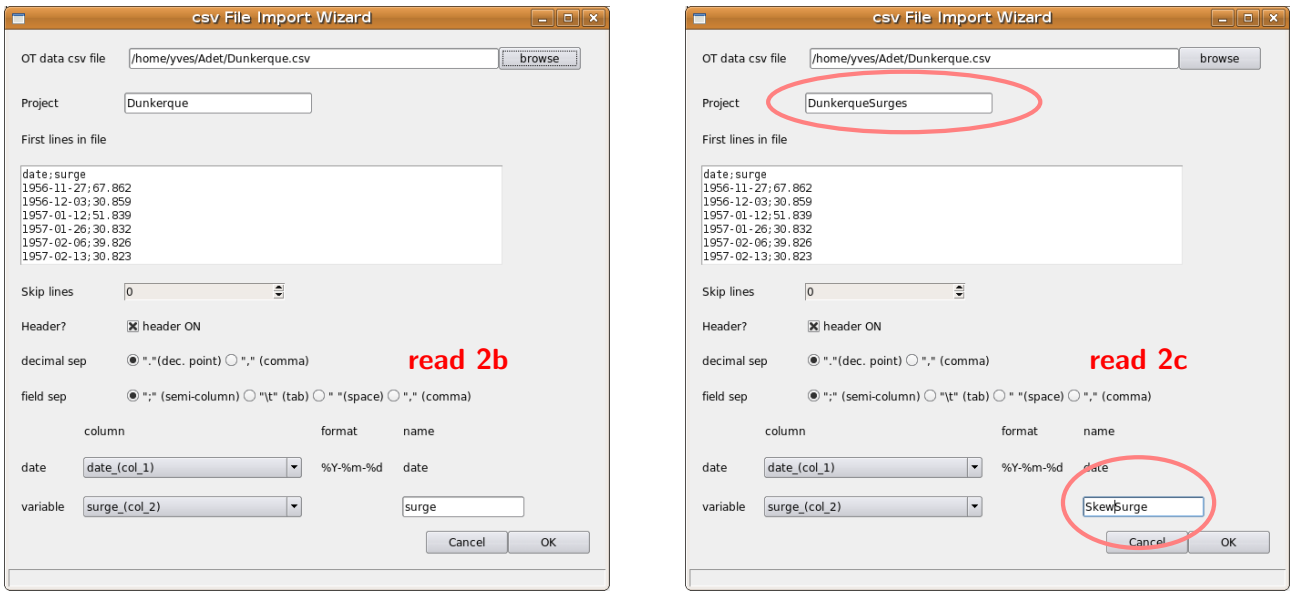


Figure 2.7: In the *csv File Import Wizard* window, once a suitable file has been selected, the first lines read are shown and guessed choices are proposed (left panel). The project and variable names can be changed (right panel).

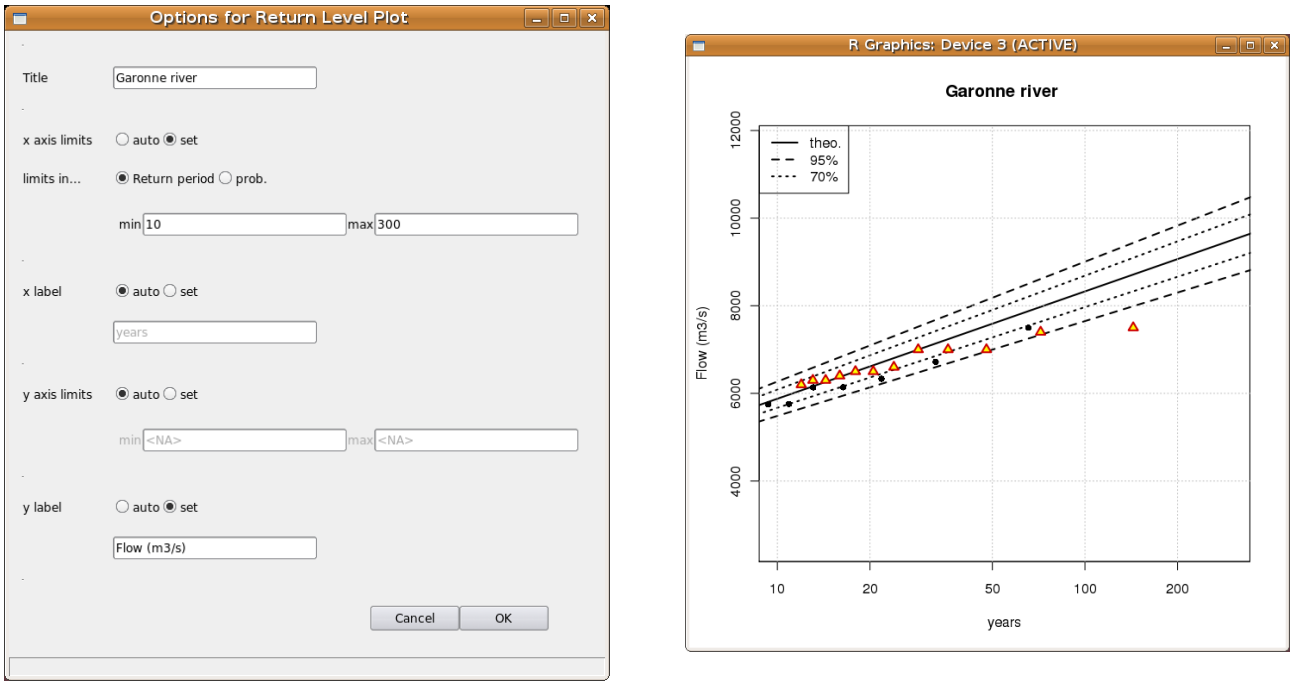


Figure 2.8: Changing graphical parameters interactively. The main title and the y-axis label were changed, and the displayed range for return periods was set to the (10, 300) years interval.

2.8 Exporting results

Several possibilities

Results can be exported using plain *copy and paste*. On Windows systems, the tables (coefficients and predictions) are copied as such to the clipboard using the third mouse button. They should then be properly pasted in a spreadsheet or converted into a table within a word processor document.

Beside this "poor man export", two export possibilities are provided, corresponding to buttons Export HTML and Dump R of the results tab (see figure 2.5). They are described now.

HTML

To export the results in HTML format, the user is prompted to choose a *directory* where the results will be stored as files. In this directory, a HTML file named `index.html` will be created, as well as the return level plot saved in bitmap format as a `RLplot.png` file. Two css[1] stylesheets are also written in this directory for use with screen and printer media.

When the directory is created or filled, **RenextGUI** tries to browse its content with the default web browser used by R. This can be queried/changed using the function `getOption` and `options`

```
R> getOption("browser")
[1] "/usr/bin/firefox"
```

[1] Cascaded Steel Sheet

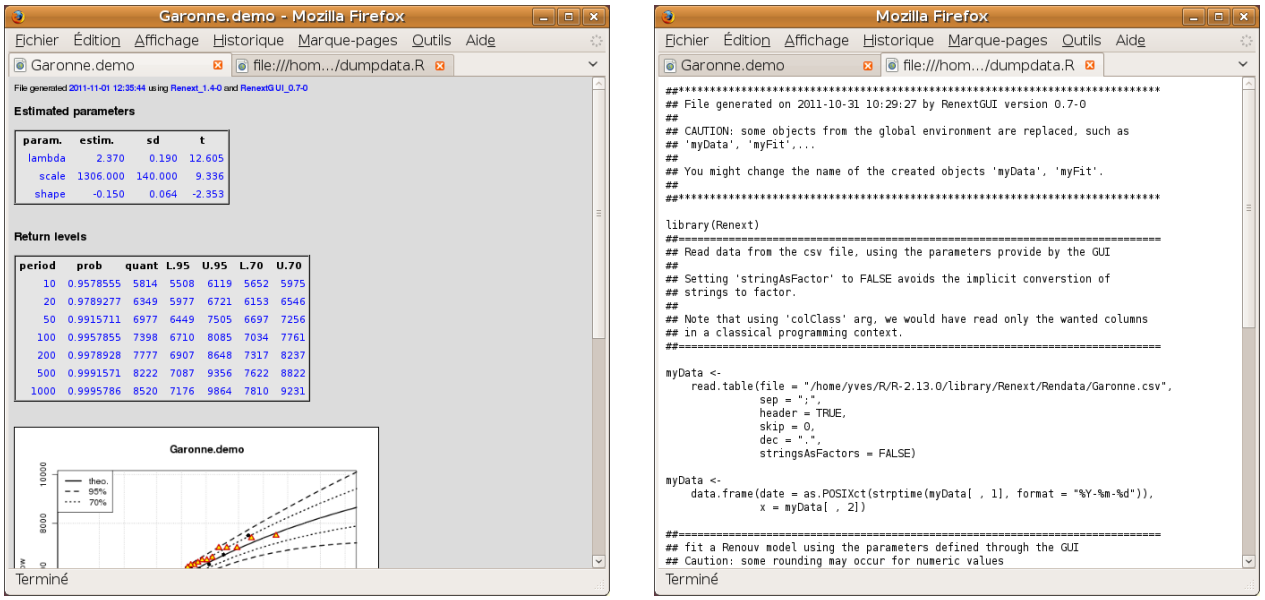


Figure 2.9: Exported HTML (left panel) and R dump (right panel) as displayed by Mozilla Firefox.

The results should appear as in figure 2.9, left panel.

Dump R

A "dump" R program can be created from the GUI. The result is a text file (ascii) with an extension usually chosen as `.R` or `.r`. No file extension is automatically provided or added. Windows users should thus remind to *set the file extension explicitly when creating a new dump file.*

Using R dumps requires some knowledge of the R system and of the R programming language.

The simple R program the created reads the data from the csv file using the parameters chosen in the GUI (column delimiter, . . .), and then proceeds to the POT fitting. This program is intended to be edited by the user who should for instance: rename the created R objects, change titles, ... It only gives hints about the use of the **Renext** package, and the interested user should refer to the manual or to the vignette of **Renext**.

Note that the program only retains the *filename* used as csv source and *not the file content.* Thus if the file has been edited, the results given by the program will differ from those shown in the GUI.

Appendix A

Known problems

A.1 Mixture of exponential distributions

Problem. When a mixture of exponential distributions is used in a POT fitting, the approximated Hessian matrix often degenerates. Then no usable standard deviation for the estimated parameter exist. The problem is due to the poor identifiability of this model (and also possibly to the parameterization used).

A.2 Validation

Problem. When using the spin button for the threshold selection (see left panel of figure 2.3), the threshold appearing on the screen may differ. This occurs when the mouse is kept on the defilement arrows (up/down) in order to achieve *several* steps.

Solution. Use the arrows only for one step or validate the field with carriage return.

Bibliography

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

[2] Yves Deville and IRSN. *Renext: Renewal method for extreme values extrapolation*, 2011. R package version 1.4-0.

[3] Original S functions by Stuart Coles, R port, and R documentation files by Alec Stephenson. *ismev: An Introduction to Statistical Modeling of Extreme Values*, 2006. R package version 1.32.

[4] Paul Gilbert. *numDeriv: Accurate Numerical Derivatives*, 2011. R package version 2010.11-1.

[5] Eric Gilleland, Rick Katz, and Greg Young. *extRemes: Extreme value toolkit.*, 2004. R package version 1.59.

[6] Michael Lawrence and Duncan Temple Lang. RGtk2: A graphical user interface toolkit for R. *Journal of Statistical Software*, 37(8):1–52, 2010.

[7] Michael Lawrence and John Verzani. *gWidgetsRGtk2: Toolkit implementation of gWidgets for RGtk2*, 2011. R package version 0.0-72.

[8] Eric Lecoutre. The R2HTML package. *R News*, 3(3):33–36, December 2003.

[9] John Verzani. Based on the iwidgets code of Simon Urbanek, suggestions by Simon Urbanek, Philippe Grosjean, and Michael Lawrence. *gWidgets: gWidgets API for building toolkit-independent, interactive GUIs*, 2011. R package version 0.0-44.

[10] R Development Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2010. ISBN 3-900051-07-0.

[11] Alec G. Stephenson. evd: Extreme value distributions. *R News*, 2(2):0, June 2002.

[12] John Verzani. *gWidgetstcltk: Toolkit implementation of gWidgets for tcltk package*, 2011. R package version 0.0-43.

Index