| rrisk **User Guide** |

Probabilistic scenario modelling using R
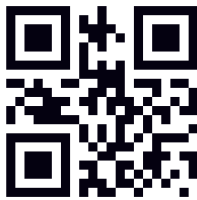
BfR

Bundesinstitut für Risikobewertung

rrisk User Guide. Probabilistic scenario modelling using R.

Prof. Dr. Matthias Greiner (MSc, DipECVPH)
Federal Institute for Risk Assessment (BfR)
Max-Dohrn-Strasse 8–10, 10589 Berlin, Germany

matthias.greiner@bfr.bund.de

www.bfr.bund.de

October 19, 2012

Check http://v28k.qr.ai for updates.

rrisk@BfR

Contents

Preface

`rrisk` means risk modelling using R.

$$\text{R} + \text{risk} = \texttt{rrisk}$$

Why R? R is an open source software for statistical computing and visualisation, powerful and free of charge. Not surprisingly, it enjoys a wide distribution. We have reviewed available R packages for functionalities that can support quantitative risk modelling. In contrast to existing packages in R, we were aiming at a tool that can be used without knowledge of the R command language. In this endeavour, we have extracted, partly re-implemented and extended the work from other authors and built a kind of user-interface. It is our particular pleasure to acknowledge the excellent work of other colleagues from which we were able to benefit by sourcing functionalities directly from their packages:

fitdistrplus M. L. Delignette-Muller, R. Pouillot, J.-B. Denis and C. Dutang (2010), fitdistrplus: help to fit of a parametric distribution to non-censored or censored data

gdata Gregory R. Warnes, with contributions from Ben Bolker, Gregor Gorjanc, Gabor Grothendieck, Ales Korosec, Thomas Lumley, Don MacQueen, Arni Magnusson, Jim Rogers and others (2012). gdata: Various R programming tools for data manipulation. R package version 2.11.0. http://CRAN.R-project.org/package=gdata

grDevices R Core Team (2012). R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. ISBN 3-900051-07-0, URL http://www.R-project.org/.

methods R Core Team (2012). R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. ISBN 3-900051-07-0, URL http://www.R-project.org/.

msm Christopher H. Jackson (2011). Multi-State Models for Panel Data: The msm Package for R. Journal of Statistical Software, 38(8), 1-29. URL http://www.jstatsoft.org/v38/i08/.

network Carter T. Butts, Mark S. Handcock, and David R. Hunter. (March 1, 2012). network: Classes for Relational Data. R package version 1.7-1. Irvine, CA. http://statnet.org/

OpenBUGS Lunn, D., Spiegelhalter, D., Thomas, A., Best, N. (2009). The BUGS project: Evolution, critique, and future directions, *Statistics in Medicine*, **28**, 3049-3067. http://www.openbugs.info

png Simon Urbanek (2011). png: Read and write PNG images. R package version 0.1-4. http://CRAN.R-project.org/package=png

rgdal Timothy H. Keitt, Roger Bivand, Edzer Pebesma and Barry Rowlingson (2012). rgdal: Bindings for the Geospatial Data Abstraction Library. R package version 0.7-12. http://CRAN.R-project.org/package=rgdal

sna Carter T. Butts (2010). sna: Tools for Social Network Analysis. R package version 2.2-0. http://CRAN.R-project.org/package=sna

tcltk R Core Team (2012). R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. ISBN 3-900051-07-0, URL http://www.R-project.org/.

tkrplot Luke Tierney (2011). tkrplot: TK Rplot. R package version 0.0-23. http://CRAN.R-project.org/package=tkrplot

tools R Core Team (2012). R: A language and environment for statistical computing.

R Foundation for Statistical Computing, Vienna, Austria. ISBN 3-900051-07-0, URL http://www.R-project.org/.

tree Brian Ripley. (2012). tree: Classification and regression trees. R package version 1.0-30. http://CRAN.R-project.org/package=tree

xtable David B. Dahl (2012). xtable: Export tables to LaTeX or HTML. R package version 1.7-0. http://CRAN.R-project.org/package=xtable

In addition to the rrisk package, we have released two other R packages in the context of the project. Although these packages can be used standalone, they were specifically designed to integrate with the rrisk:

rriskBayes Natalia Belgorodski, Matthias Greiner and Alexander Engelhardt (2012). rriskBayes: Predefined Bayes models fitted with Markov chain Monte Carlo (MCMC) (related to the 'rrisk' project). R package version 2.3. http://CRAN.R-project.org/package=rriskBayes

rriskDistributions Natalia Belgorodski, Matthias Greiner, Kristin Tolksdorf and Katharina Schueller (2012). rriskDistributions: Collection of functions for fitting distributions (related to the 'rrisk' project). R package version 1.8. http://CRAN.R-project.org/package=rriskDistributions

We hope that our tool will contribute to a richer range of options from which risk modellers can choose.

Matthias Greiner, Berlin, 1 October 2012

Introduction

Quantitative risk assessment (QRA) models are required in many risk analysis frameworks such as for example food safety and international trade with animals, animal products and food. QRA models aim at an evaluation of the risk of an adverse consequence (endpoint) associated with a given hazard and risk scenario. The use of QRA is governed by international guidelines, which encourage the use of stochastic (probabilistic) methods if suitable data are available for model parameterisation. The transparency of the risk model is an important pre-requirement for scientific review and mutual acceptability of QRA results by trading partners and competent authorities. Both the conduct and the review of QRA models require an interdisciplinary approach. A well-structured and comprehensive model documentation is often difficult to achieve in practice and a lack of clarity in the reporting of the evidence, the modelling concepts, model assumptions and results are not uncommon. The limitations of narrative model descriptions along with mathematical formulas are often due to the most serious restriction: expert time.

Against this background, `rrisk` has been developed in R [R Development Core Team, 2009] (www.r-project.org) to (i) guide the modeller through comprehensive and consistent model documentation, (ii) ensure the identity between definition, implementation and documentation of the model and (iii) to automatically generate a technical model documentation. The open source licence under which the tool is distributed guarantees transparency to the level of source code. Furthermore, it provides a basis for sharing the

It is recommended to study `Demomodel1_report.pdf` as an example of a simple model generated and reported with `rrisk`. This guide describes how to get started with using `rrisk` (Section 1) and explains how to define a new model (Section 2). The concept of scoring systems for assessing uncertainties is introduced (Section 3) and then the essential steps for setting up, configuring, running and reporting a model are illustrated (Section 4). Finally, the functionalities under evaluation and a few features for further development (Section 5) and a glossary are provided (Section 6).

The Annex contains the full technical specifications and list of features of `rrisk` (Annex A), a technical guide for defining model items (Annex B) a summary of `rrisk`-syntax rules (Annex D), a brief guide to LaTeX-syntax (Annex E) and some frequently asked questions (Annex F).

Please visit the `rrisk` homepage at www.bfr.bund.de/cd/52162 for further information.

1. Getting started with `rrisk`

1.1. Software installation (Windows)

Step 1: The R software is "a language and environment for statistical computing" and the software on which `rrisk` was built upon. Download and install the R software (www.r-project.org). Choose "CRAN" from the left hand menu, and select the mirror closest to you. Select "Windows" on the next page, and then "base" for the base installation of R. The next page holds a link to download the most recent version of R as an .exe file.
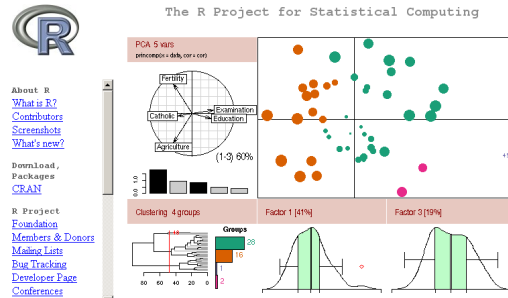
> **Note:** `rrisk` requires version R 2.10.0 or higher. If you already have R installed, make sure it is up to date.

Step 2: The report files (in pdf format) will be generated with the LaTeX environment, which yields beautiful results and typesetting. MiKTeX is the most well known LaTeX–distribution for Windows. Download and install the latest MiKTeX edition (http://miktex.org/). Execute the MiKTeX Updating program.
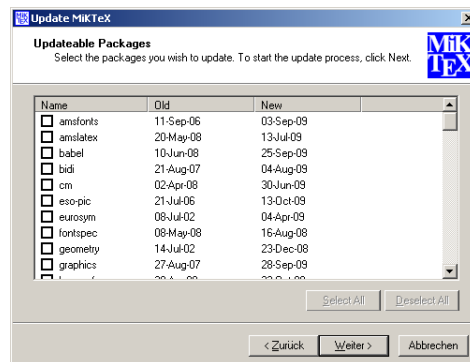
> **Note:** This update is usually done twice, first for updating the main program and then for packages. It should be repeated regularly, say annually.

Execute the MiKTeX Settings program and click on Refresh FNDB and Updates Formats. The option "Yes" must be chosen to install packages on-the-fly.
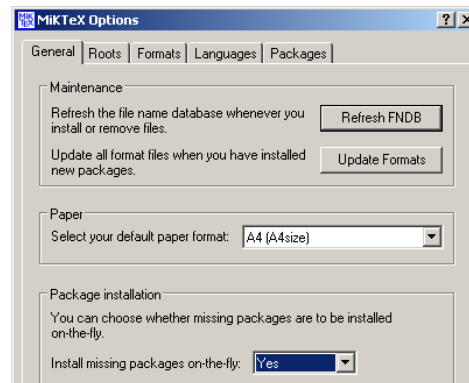
Step 3 (optional): For creating and maintaining a literature data base, you might want to use JabRef. Download and install the latest JabRef edition (http://jabref.sourceforge.net/). Generate a reference data base using JabRef by importing references from other applications or via web search. Store the data base (bib-file) in the working directory).
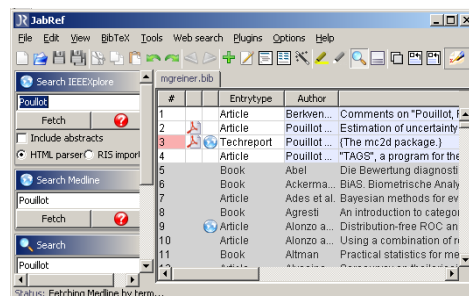
Each data base entry has a unique BibTeX-key (see bottom field). This key is used when citing any reference (see Annex E). Using Options, Preferences, BibTeX key generator, a standard key format can be set.
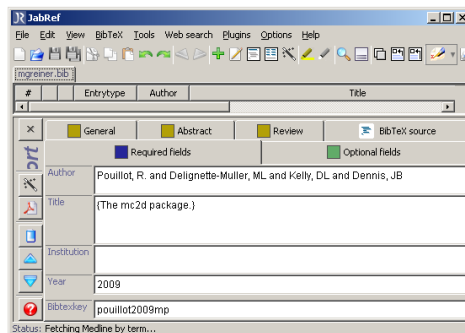
1.2. Software installation (Mac OS X)

Follow the Windows instructions to download and install R, but choose MacOS X as your operating system. Download the latest R version by clicking the link to the .pkg file. Install it by double–clicking that icon in the finder.

You will also need to install the tcltk-package from the following link: `http://cran.r-project.org/bin/macosx/tools/` Click on the link to the tcltk .dmg file to download it, and install the package to your computer.

A comprehensive LaTeX distribution for Mac OS is MacTeX. You can find it at: `http://www.tug.org/mactex/`

Download and install the MacTeX package.

> **Note:** The MacTeX distribution is 1.6GB large. Expect this download to take a while. !

For a comfortable BibTeX editor, you can use JabRef. This is a program that runs on the Java VM, and thus can be executed in Windows, MacOS or Linux. Please refer to the Windows installation.

1.3. Software installation (Linux)

Depending on your distribution, you most certainly have one of two package managers installed: The Debian–based `apt` system, or the Redhat–based `rpm` manager. You can either use the command line with your respective package manager, or a graphical tool to install the following packages:

- `r-base`
- `texlive-latex-base`
- `texinfo`

Your package manager might install more than those packages because of package dependencies. An example shell command for an Ubuntu distribution is:

```
sudo apt-get install r-base texlive-latex-base texinfo
```

If you want to use JabRef (see section 1.1), install `JabRef` as well.

1.4. Installing the `rrisk` package

> **Note:** Connection to the internet is required when starting your first `rrisk`-session as several R-packages need to be installed. !

The `rrisk` package depends on the following packages, which will be installed automatically during the installation of `rrisk`: eha, fitdistrplus, gdata, grDevices, mc2d, methods, mgcv, msm, network, png, rgdal, rriskBayes, rriskDistributions, sna, tcltk, tkrplot, tools, tree, xtable. Moreover, the `OpenBUGS` software should be installed directly from it's homepage (http://www.openbugs.info/w/Downloads) bevor installing `rrisk`. Open the R software and choose the option to install a new package from a CRAN repository. If the `rrisk` package is available, you are able to install `rrisk` directly by selecting it from the list. Alternatively, you may enter into the R console the command:

```
> install.packages{"rrisk",dependencies=TRUE)
```

> **Note:** The availability of `rrisk` in the CRAN repositories is beyond control of the maintainers. You may always obtain the latest versions of `rrisk` from www.bfr.bund.de/cd/52162 and proceed as explained below.

Installation of the `rrisk` package from the BfR download site only requires that the `rrisk`-folder (not the zip archive) is copied into the library directory of your R installation. Thereafter it is possible to load the package via the standard R menu.

1.5. Loading the `rrisk` package and opening the user guide

After this, the `rrisk` package can be loaded by

```
> library(rrisk)
```

The `rrisk` package contains the pdf-document "**rrisk User Guide**", which is opened using

```
> openUserGuide()
```

to open the user guide.

1.6. Starting `rrisk`

You can now start an `rrisk` session by typing

```
> rrisk.start()
```

`rrisk` loads a few demo models by default (see section 1.8). You can specify which demo models shall be loaded with the function parameter `useDemoModels`. For example, to start `rrisk` without loading any demo models, type

```
> rrisk.start(useDemoModels="no")
```

Instead of `"no"`, you can type `"all"` to load all of the three demo models (this is the default), or `"Demo model 1"`, `"Demo model 2"` and `"Demo model 3"`, respectively, to only load one of the three models.

1.7. The `rrisk` main menu

When first starting `rrisk`, you will be shown the main menu:

rrisk main menu

```
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
1    Session infos
2    Available models
3    Scoring systems
4    Create new model
5    Save model
6    Remove model
7    Load demo models
8    View/edit/run model
9    Set working directory
10   Show working directory
11   About rrisk
12   Exit rrisk
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
```

To choose a menu point, enter the corresponding number (e.g. 3 for `Scoring systems`) and press the return key.

1. **Session infos:** Choose this option to display a short summary of the available models and scoring systems.

2. **Available models:** Only the available models are displayed, but with more detailed information than in the menu item above. You see a summary of the model name, its authors and its parts and items.

3. **Scoring systems:** Enter here to view, create, edit or save scoring systems. The sub–menu is explained in detail in section 3.

4. **Create new model:** This option will display a sub–menu which allows you to create a new model, or import or duplicate another model. See section 2 for details.

5. **Save model:** Go here to export your model as an R object. A sub–menu will ask you for the model you wish to save. The following pop–up window allows you to save that object (an .Rdata file) to your hard disk. This file will contain all the info about the model, including possible data sets it contains, concept and networks graphs, etc. If you want to exchange your model with a colleague, it will suffice to transfer only this one file.

6. **Remove model:** Enter here if you want to remove a model from your workspace. You will not be asked for confirmation. It is recommended to first save the model to your hard disk before you remove it from `rrisk`. Later on, demo models can be loaded again (see the next item).

7. **Load demo models:** If you have started `rrisk` without loading the included demo models, you can choose this option to subsequently load those models.

8. **View/edit/run model:** This option opens a dialog where you are asked which model you want to work on. After choosing a model, you will be shown a sub–menu with numerous options to view, edit, save or run your model. All sub-menu items can be used for user defined models. However, some of them are not allowed for rrisk demo models, which is intended not to be changed mainly. Hence, if you wish to use a modified rrisk demo model, duplicate this model (see section 2) and vary this user defined copy ad libitum. Refer to section 4.4 for details.

9. **Set working directory:** After entering here you are shown a new window to select a folder for your new working directory. Afterwards, click on `OK`. Setting a new working directory might be adjuvant if you wish to save all of your created data files, as graphics and reports, in a certain folder.

10. **Show working directory:** This option lets `rrisk` show your current working direc-

Concentration c_0 of E. coli 0157:H7 in raw beef before processing	*Occurrence*
Concentration of E. coli 0157:H7 at the end of the processing time t with exponential growth rate k and growth function $c_t = c_0\, e^{kt}$	*Processing*
Fraction i of E. coli inactivated during cooking/heating `dishsize` of one serving ground beef for the risk group Serving size s fitted to data `dishsize` Number n of viable E. coli O157:H7 per serving ground beef	*Exposure*
Dose response: theoretical probability r of illness caused per single E. coli cell (single hit) Probability P of illness associated with consumption of one serving of ground beef	*Consequences*

Figure 1: *DemoModel* example model.

tory, where all of your desired models have to be saved. If you want to change your working directory, you can go to `Set working directory`.

11. **About rrisk:** Enter here to view a pop–up window with information about `rrisk` and its license.

12. **Exit rrisk:** The last option in any dialog exits this dialog and brings you one level back. This option in the main menu exits `rrisk`.

> **Note:** Make sure to save your models before you quit the program!   !

1.8. Example models

`rrisk` comes pre–shipped with example models, or *demo models*. Those models are of the class "demo model", as opposed to "user defined model", which are basically all the models you will create.

The example model "DemoModel1" has been described by Pouillot et al. [2009] for demonstration of their risk modelling package in R. This model – although too simplistic to answer a real microbiological risk question – is well suited to demonstrate the functionality of `rrisk` for documenting limitations, assumptions and uncertainties. For the purpose of demonstration, the model has been structured in four parts, in which ten items are defined (Fig. 1). The last item P reflects the outcome of interest.

The model is implemented in `rrisk` and can be viewed and run. You can also generate a report from it.

> **Note:** Models of the class "demo model" are protected against any changes. However, you can still edit or extent a demo model either by duplicating or saving and re-importing of the demo model.   !

2. Developing new models in `rrisk`

2.1. First steps in model development

A whole range of questions needs to be addressed when a new model is developed: *What is the risk question? What kind of answers are required? What is known about the problem? How should the model reflect what is known and what is not known? What are the principles, components and elements of the model?*

The first steps in development of a new model aim at the agreement between subject matter experts and risk modellers on a conceptual framework for addressing the given risk question. During this phase, scientific information is gathered, structured and visualised. Pathways of propagation of risks are drafted. Parameters and required data to inform the parameters are identified. `rrisk` does currently not provide specific tools to support this model development phase. Knowledge modelling tools such as Cmap (http://cmap.ihmc.us/) can be recommended as a platform to develop, visualise and share conceptual models. In the following it is assumed that a scientifically plausible model concept exists.

> **Recommendation:** The model concept should first be sketched and discussed with subject experts without using any modelling tools. The description and definition of a model should then be elaborated in `rrisk`. One or more graphics (jpg or png) should be prepared to explain the model.

∗

2.2. Setting up a new model with `rrisk`

To create a new model, proceed as follows:

- From the `rrisk` main menu, select `Create new model` by entering the number 4. The following prompt gives you four possibilities of creating a new model:

    - **Empty model:** Create a new, empty model from scratch.

    - **Load model definition template (.Rmdt file):** You can import a model from a model definition template (MDT). (in development)

    - **Load rrisk model object (.Rdata file):** If you have saved your model as an R object, this option can load it back into R.

    - **Duplicate an available model:** This option duplicates an existing model, for example to try new options, or to duplicate a demo model to be able to edit its options.

- Enter a model name in the popup prompt and click "OK" to confirm. Note that you must choose a unique name. If the name you enter is already taken by another model, you will be prompted to enter another name.

To set and edit the properties of your newly created model, please refer to section 4.4.

3. Scoring systems

A systematic assessment of uncertainties is mandatory in `rrisk`. The structure and definitions for assessing uncertainties are referred to as "scoring system". Uncertainties are documented at a general level (e.g. scenario and modelling approach) and at the level of each model item (individual parameters, data, functions). The uncertainty related to each aspect (e.g. the scenario uncertainty related to the hazardous agent or the daily amount of intake for a particular food item) can be assessed using a set of criteria (e.g. describing the level of uncertainty, the degree of agreement among experts). The elements of the scoring system are shown in Fig. 2. The scoring system is flexible and can be adapted to the desired detail.

The default criteria for assessing uncertainties (and the knowledge base) are described in the following section. The scoring values only serve the purpose of visual summaries and have no impact on any weighting or calculation. The user can modify or replace the default scoring system to adapt to any user-defined set of criteria and score values.

Generally, a scoring system can be either of type *rrisk scoring system* or of type *user defined*. The *rrisk scoring systems* are included by default and can not be edited by the user. If you wish to edit them, you should duplicate the scoring system to a *user defined* scoring system. Scoring systems that are duplicated or otherwise created by the user are automatically of type *user defined*.
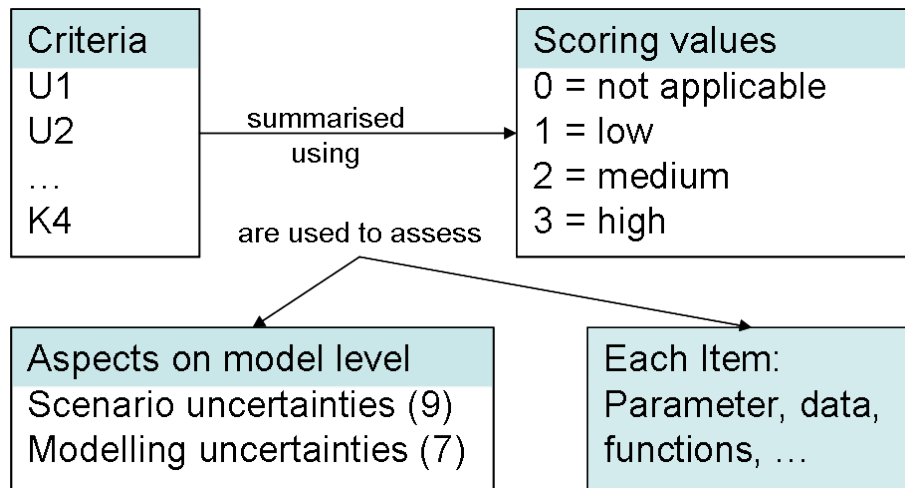
Figure 2: The components of a scoring system in `rrisk`, consisting of criteria and scoring values, for assessing general model uncertainties and each model item. The lightblue colour indicates a mandatory element of a scoring system. The default criteria, score values and aspects are shown on white background. They can be defined by the user for a given model.

To work with scoring systems, choose `Scoring systems` from the `rrisk` main menu.

3.1. Default scoring system

Table 1 shows the criteria and scores of the default scoring system implemented in `rrisk`. This scoring system is also used in the demo models, please refer to those models for an example. By default, this scoring system is applied to 16 general aspects of scenario and modelling uncertainties and to each model item.

3.2. View existing scoring systems

To view a list of all available scoring systems, choose `View scoring systems (summed up)` for a quick summary, or `View scoring systems (full info)` to select one system and view its full definition.

3.3. Importing or setting up a new scoring system with `rrisk`

To create a new scoring system, proceed as follows:

- From the `rrisk` main menu, select `Scoring systems` by entering the number 3, then `Load/new scoring system`. The following prompt gives you four possibilities of creating a new scoring system:

  - **Empty scoring system:** Create a new, empty scoring system from scratch. You will be prompted to enter the new system's name. To further edit the new system, please refer to section 3.4.

  - **Load scoring system definition template (.Rsdt file):** You can import a scoring system from a scoring system definition template (SDT).

  - **Load rrisk scoring system object (.Rdata file):** If you have saved your scoring system as an R object, this option can load it back into R. (in development)

  - **Duplicate an available scoring system:** This option duplicates an existing scoring system, for example to try new options, or to duplicate the default scoring system to be able to edit its options.

- Enter a scoring system name in the popup prompt and click "OK" to confirm. As with the model name mentioned earlier, you must choose a unique name for the scoring system.

3.4. Editing a scoring system

To edit your scoring system and its individual scores, choose `Edit scoring system`. Select your desired scoring system in the following dialog. The emerging window lets you specify the scoring system's name and its descriptions, as well as the individual score symbols and their names and explanations. You can switch between the individual scores by using the "previous" and "next" buttons. Note that the upper five items are the scoring system's description are the same in each window. Only the individual scores, i.e. the lower three items in the list change when you skip through the items. To add a new score, click "next" until you are at the end of the list and click "add new". You can also delete single scores with the "delete" button. Here is an explanation of the items the pop–up window displays:

Fields for the scoring system as a whole:

- **Name of the scoring system:** The name of the scoring system.

- **Scoring values:** These are the numeric values a single score can take on.

- **Scoring colors:** The respective colors for the values in the model uncertainty graph.

- **Scoring meanings:** This should be space–separated words, one for each scoring value

- **Explanation:** Briefly explain the scoring system's use here

- **Table header:** The scoring system will be printed as a table in the final model report. This entry will be printed as that table's caption.

Fields for individual scores

- **Score symbol:** Provide a short symbol for identifying a single score here.

- **Score name:** This is a short name for this score.

- **Score explanation:** This field should hold a longer explanation describing the purpose of the score.

  Note: Edit the default scoring system using a copy of a demo model to review the entries above. !

3.5. Save/export an existing scoring system

To save a scoring system to your hard disk, select `Save/Export scoring system` from the scoring system menu. The following prompt lets you choose the scoring system you want to save. You can save the system as a text file by choosing `Export scoring system definition template (.Rsdt file)` or—the recommended way—as a .RData file by choosing `Export rrisk scoring system object (.Rdata file)`. A window will pop up and ask you for the local folder in which you want to save your scoring system data.

3.6. Remove a scoring system

To remove a scoring system from the `rrisk` session, choose `Remove scoring system`. Select the scoring system you wish to remove, and confirm your choice in the emerging window.

Table 1: *Default criteria and qualitative scores for uncertainty (U) and knowledge (K) base (modified after Van der Sluijs et al. [2005]).*

Criterion	Scores and definitions (interpretation guide)[a]
(U1) Plausibility	1=high (highly plausible despite absence of factual evidence), 2=medium (plausible despite absence of factual evidence), 3=low (plausibility questionable or not assessed; fictive or speculative assertion)
(U2) Inter-subjectivity	1=high (expert opinions are consistent among peers), 2=medium (expert opinions vary among peers to some extent; deviating minority opinions exist), 3=low (expert opinions vary considerably among peers; no clear majority opinion exists)
(U3) Choice space	1=limited (hardly any choice from alternative assumptions), 2=moderately limited (limited choice from alternative assumptions), 3=wide (ample choice from alternative assumptions)
(U4) Sensitivity to limitations	1=hardly sensitive (more resources such as time, money, etc. would not markedly reduce the uncertainty), 2=moderately sensitive (more resources would markedly reduce the uncertainty), 3=very sensitive (more resources could ressolve this aspect of uncertainty)
(U5) Sensitivity to interests	1=hardly sensitive (conflict of interest (COI) can be excluded), 2=moderately sensitive (no declared interest or COI unlikely to occur), 3=sensitive (declared interest or COI likely to occur)
(U6) Influence	1=low (local effect in the model, few pathways affected), 2=medium (several pathways affected), 3=high (many pathways affected)
(K1) Proxy	1=high (exact measure or good fit for purpose), 2=medium (well correlated), 3=low (weak correlation or not clearly correlated)
(K2) Strength of evidence	1=high (strong evidence, systematic review and meta-analysis of suitable studies, official statistics, census data, controlled randomised trial, observational study with appropriate study design and adjustment for confounding), 2=medium (some evidence, like level 1 but constraints of internal or external validity apply), 3=low (no evidence, expert opinion, assumption)
(K3) Internal validity	1=high (internal validity assessed with positive outcome, measurement methods validated and considered best-suited for the purpose), 2=medium (internal validity assessed with acceptable outcome, measurement methods validated and considered acceptable for the purpose), 3=low (internal validity not assessed or assessed with questionable outcome)
(K4) External validity	1=high (external validity for the purpose of this assessment can be presupposed), 2=medium (external validity for the purpose of this assessment can be presupposed with limitations), 3=low (external validity for the purpose of this assessment may not be given)

[a]For all criteria the level "0=not applicable" applies.

4. Viewing, configuring and saving the model

This section explains the main menu point `View/edit/run model` in detail. After entering this menu point and choosing which model you wish to work on, you will be presented the following menu structure:

```
Choose action to be executed
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
1     Summing up
2     View model
3     Show model graphs
4     Edit model topics
5     Get topics from other models
6     Model items
7     Load reference list
8     Load concept graph(s)
9     Export model objects and definition
10    Scoring system
11    Change sequences in model topics
12    Run model
13    Create model report
14    Save model
15    Exit dialogue
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
```

4.1. Summing up

This option shows you a summary of your chosen model, consisting of the model name, its version as well as its authors, and information about its parts and items.

4.2. View model

This option opens a sub–menu which gives you access to the full description of the model's properties. Enter the according number at the prompt, and `rrisk` will output detailed information about that property.

1. **Model complete description:** This prints out an extensive list of all the model's properties. See below how to view single parts of this description.

2. **Model version:** This outputs the model's status (draft or release), its version, subtitle and the last author to edit it.

3. **Model settings:** Here you can see the basic settings of the model. For a detailed explanation please refer to section 4.4.3

4. **Model name and authors:** This option makes `rrisk` output the name of the model as well as its authors and their institutions. If available, their e–mail addresses are also printed.

5. **Model name, version, authors and items:** This option prints a summary of the model and its parts and items.

6. **Model basic description:** This prints the model's description, which consists of the optional name of a document to which the model relates to (mainDoc), the model's background, its objectives and its scope and concept.

7. **Scoring system for uncertainty and knowledge base:** This option prints the name and description of the scoring system used in this model. For a detailed explanation on scoring systems, see section 3.

8. **Assessment of general uncertainties:** Here you will see a list of the model's uncertainties. `rrisk` will print their categories, explanations, and a list of scores according to the chosen scoring system.

9. **Model parts:** Choose this option to see a list of the model's parts. You will also see a list of items associated to each part. Note that an item can only belong to one part. The parts provide a structure for the report. They do not interfere with the sequence of the execution of the model.

10. **Model items:** This prints a list of each of the model's associated items. You will see a detailed table of many properties, among them an item's name, definition, its scores according to the scoring system, and remarks.

11. **Model related data:** If the model contains items of type `data`, this option will print the item's associates variables and the first few values in the data set.

12. **Model validation and verification status:** This option prints the text associated with the model's validation and verification status.

13. **Comments:** Comments can be reviewed by using this command.

14. **Conclusions:** Here you can view the conclusions gained after the model was run.

15. **References:** This prints the reference list.

16. **Glossary and abbreviation:** The glossary and abbreviation lists are printed in combination by this command.

17. **Figures provided by user:** This will open a graphics window for every image the user provided (see section 4.8 on how to include graphics in your model). `rrisk` will also print the explanations for the graphs in the console window.

4.3. Show model graphs

Displays a sub–menu where you can choose from displaying different graphs: The title of each displayed graph shows the graphic's name, and in the next line how you can reference this image. For example, if the graph's title says `\ref{fig:Demomodel1_conceptGraph1}`, you can refer to that image in your report by typing this line. An example sentence would be:

```
Refer to figure \ref{fig:Demomodel1_conceptGraph1} for details.
```

1. **Concept graph(s):** Shows the concept graph or graphs. These are the graphics provided by the user.

    Note: Don't use this command when the R–window is expanded to full size   !

2. **Network graph:** Shows the Network graph `rrisk` generated from the information about the items.

3. **Model uncertainties:** Shows an image of the uncertainties of the model, color–coded for easier interpretation.

4. **Close all graphic devices:** If you opened multiple windows with graphics, choose this option to close all open graphics.

4.4. Edit model topics (only for user defined models)

To edit a model, select (from the main menu) `View/edit/run model`, choose your model and select `Edit model topics` in the following dialogue. You can edit the following settings:

```
Choose model topic for editing
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
1        Edit model name and version
2        Edit model authors
3        Edit model settings
4        Edit basic descriptions
5        Edit references
6        Edit uncertainties
7        Edit model parts
8        Edit glossary
9        Edit abbreviations
10       Edit validations
11       Edit comments
12       Edit conclusions
13       Exit dialogue
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
```
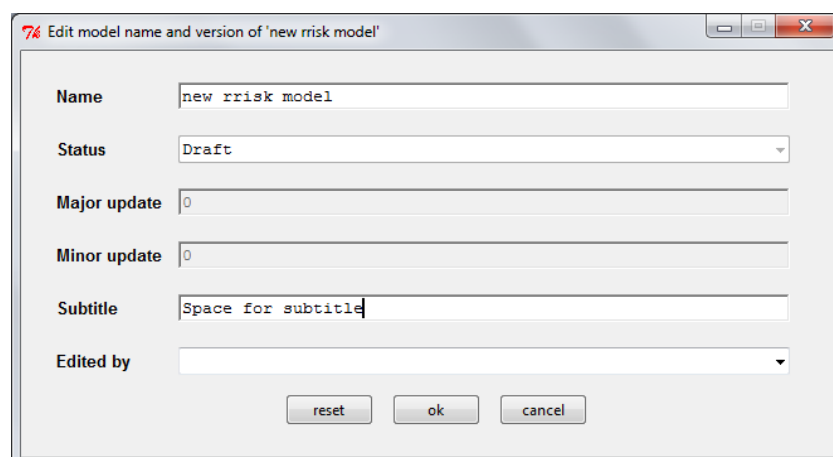
4.4.1. Edit model name and version

The basic information of the model including model name, status, minor update, major update, subtitle, cover header and the last editor can be edited via the number 1.

> **Note:** You cannot edit the model's major and minor version directly. Those numbers are automatically updated once you generate a report file. As well, the status is not editable. **!**

The "Name" field must contain a unique name. If you enter a name that already exists for another model, you will be prompted to enter another name. If the status is set to "Draft" and a report is generated, a water mark will be printed across all its pages. The status and the model's major and minor update will be modified during the report's compilation. You can add and edit the subtitle on the model report's title page in the "Subtitle" field. Before entering something there, no subtitle exists. next to the term "Edited by", you have a choice of names. Choose the one person who edited the model at last. The interface is as follows:
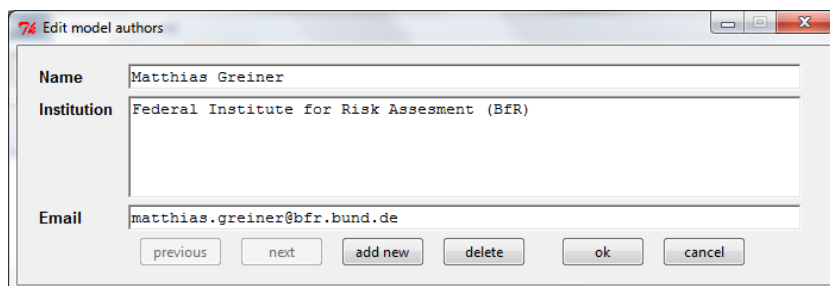
4.4.2. Edit model authors

The basic information of the model authors including their names, institutions and e-mails can be edited by entering the number 2. For a new registration the name and the institution is required, while providing an e-mail is optional.
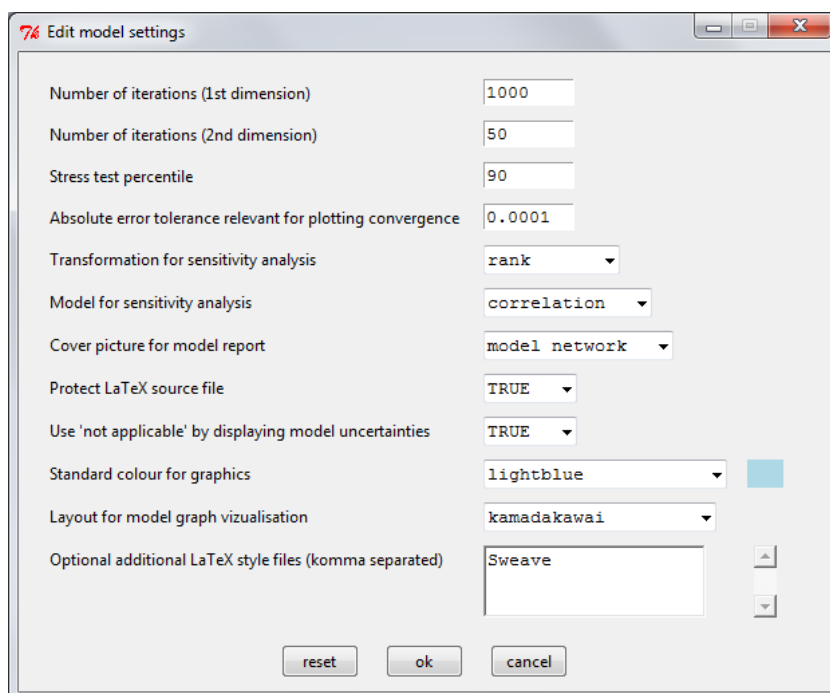
> **Note:** To add a new author, you must first click the "next" button until you reach the end of the list. This applies to every dialogue from now on. After, click on "add new" to get a window with empty fields.

You must provide an institution for every author. The e–mail field is optional. The interface is as follows:

4.4.3. Edit model settings

Enter the number 3 at the R prompt. The following screenshot shows the default settings.

Number of iterations (1st dimension). Possible entries: numerical value N for the number of iterations for standard (1st dimension) Monte-Carlo simulation.

Number of iterations (2nd dimension). Possible entries: numerical value $N2d$ for the number of iterations for second order (2nd dimension) Monte-Carlo simulation.

Stress test percentile. (in development)

Absolute error tolerance relevant for plotting convergence.

Transformation for sensitivity analysis. Possible entries: `rank`, `z-score` (number of standard deviations from the mean) or `identity` (original data) to define the transformation of variables.

Model for sensitivity analysis. Possible entries: `correlation` (Pearson's correlation coefficient) and `regression` (multi-variable linear regression). The latter option uses a multiple regression model of the outcome function and the `mcrv`-items as independent variables while the first option does not presuppose any linear association between outcome and `mcrv`-items.

Cover picture for model report. Possible entries: Model network (default); concept graph; none.
If you choose "concept graph" and have added multiple concept graphs to the model, the first concept graph in your list will be used. To change the order of your graphs and make another graph the first one in your list, refer to section 4.11.

Protect LaTeX source file. Possible entries: TRUE and FALSE. If TRUE is entered, the source file (`Report.tex`) for the report will not be deleted after the .pdf file is generated, and thus will be available for editing.

Use 'not applicable' by displaying model uncertainties. If this option is set to `FALSE`, the uncertainties marked as "not applicable" will not be displayed in the uncertainty graphics. If it is set to `TRUE`, the uncertainties marked as "not applicable" will be visualized as blue points.

Standard colour for graphics. Choose one of the listed colours as default color for the generated graphics. For example, the columns of the histogram of the outcome function and of the tornado charts as well as the lines of the convergence plots and of the uncertainty intervals for the cdf quantiles and, finally, the nonparametric regression plots will take this colour.

Layout for model graph visualisation. Possible entries: `kamadakawai`, `fruchtermanreingold`, `circle` will affect the placing of symbols for items on the model network graph. Note that the fist two options result in random orientation of the coordinates.
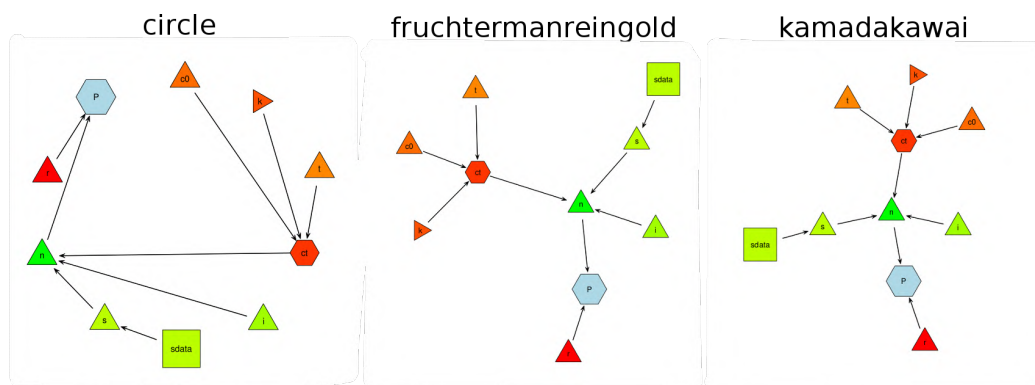
Figure 3: The three possible layouts, shown on the "Demo model 1"

Optional additional LaTeX style files (comma separated). Possible entries: any additional LaTeX style file (comma separated) if needed. Pre-installed files are `hyperref`, `multirow`, `amsmath`, `amssymb`, `gensymb`, `natbib`, `fancyhdr`, `longtable`, `pgf`, `graphicx`, `[latin1]inputenc`, `color`, `lscape`, `treeparttable`, `setspace`, `textcomp.`, `type1cm`, `eso-pic`, `here`,

`csquotes`, `caption`, `listing`.

4.4.4. Edit basic descriptions

The edit interface for basic descriptions of the model is activated by entering the number 4. Click the `previous` and the `next` button to edit the information including main document (mainDoc), background, objectives, scope and model concept description. It is not permitted to add a new item of descriptions or to delete an existed one. By clicking the `help button` the instructions for how to complete the fields will be opened. The interface is as follows:
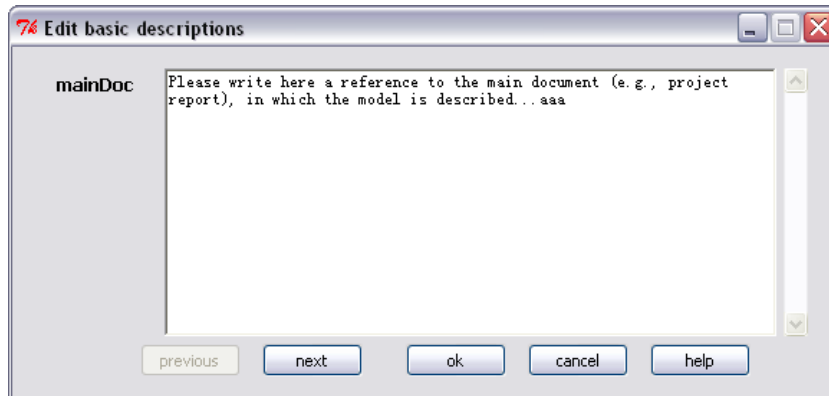
4.4.5. Edit references

Entering the number 5 opens the list of references used in the model, in BibTeX format.

> **Note:** If you use a BibTeX system, be careful not to mess up the syntax in this window, e.g. by deleting or adding a bracket. Editing of references should be done in other pregrammes such as JabRef.

The interface is as follows:
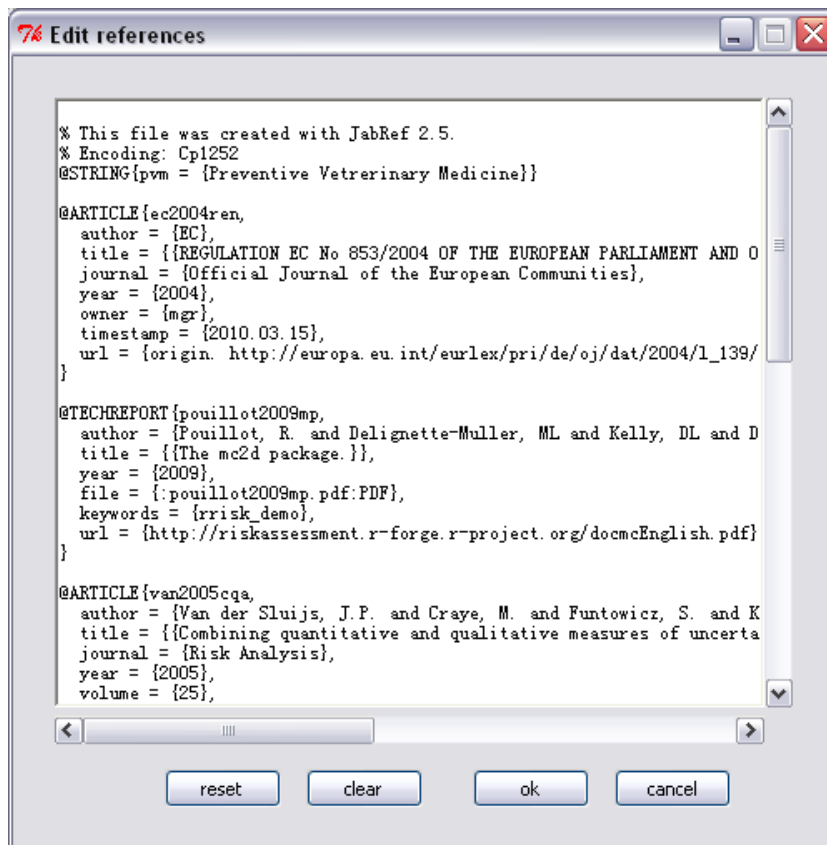
4.4.6. Edit uncertainties

The model uncertainties can be edited by entering the number **6**. The entry in the note field is the same over all uncertainties. The scoring system can not be edited and serves as a reference for the score values to be entered.

Use the "previous" and "next" buttons to switch between the different uncertainties. Main and sub category identify each uncertainty, and the explanation field should hold a short description of the specific uncertainty.

The scores must be entered as a list of numbers, separated by a single space character. The "Scoring system" field can be used as a reference for the meaning of each number and its values. The number of scoring values should match with the number of criteria.

Please refer to section 3 and 4.10 for details about the scoring system for uncertainty and knowledge base.

The interface is as follows:

4.4.7. Edit model parts

All of the model parts can be edited by choosing 7. The interface is as follows:

Name: Edit the required name field. Note, that every model part must have a name.

Explantation: Edit the explantation field to describe the part. Likewise, every model part must be explained.

Items: You cannot type anything in the "Items" field, but you can select and de-select items into a part by clicking on these items. Note that one item can only belong to one part. If you choose an item that already belongs to another part, then this item will be removed from its former part.

> **Note:** If you delete a part, the items associated with it will also be deleted! If you do not intend to do that, you must first delete the items from the "Items" field. If you

4.4.8. Edit glossary

The glossary of the model can be edited by entering 8. The interface is as follows:
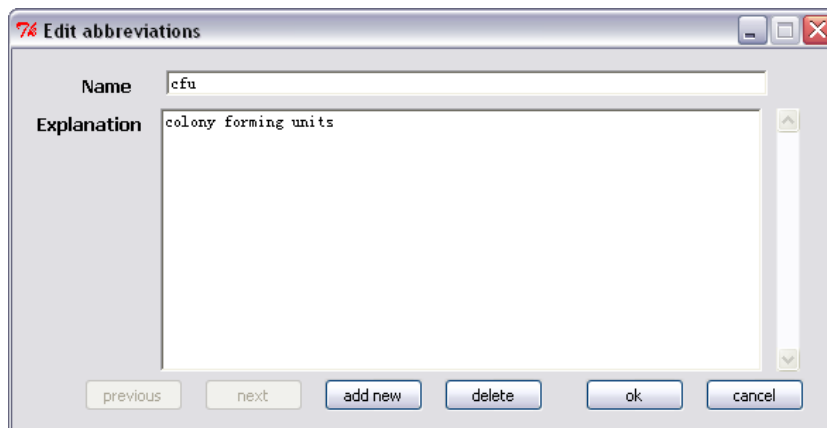
You can add new entries, or delete and edit existing entries.

4.4.9. Edit abbreviations

The abbreviations used in the model can be edited by entering 9. The interface is as follows:

You can add new entries, or delete and edit existing entries.

4.4.10. Edit validations

The validation status of the model can be edited with the number 10. The interface is as follows:

Both the name and the explanation are required for editing an existing element or adding a new element. The name will be used as heading for the respective subsection of the report.

4.4.11. Edit comments

Comments can be added or edited by choosing 11. The interface is as follows:

4.4.12. Edit conclusions

The model's conclusions can be edited by choosing 12. The interface is as follows:
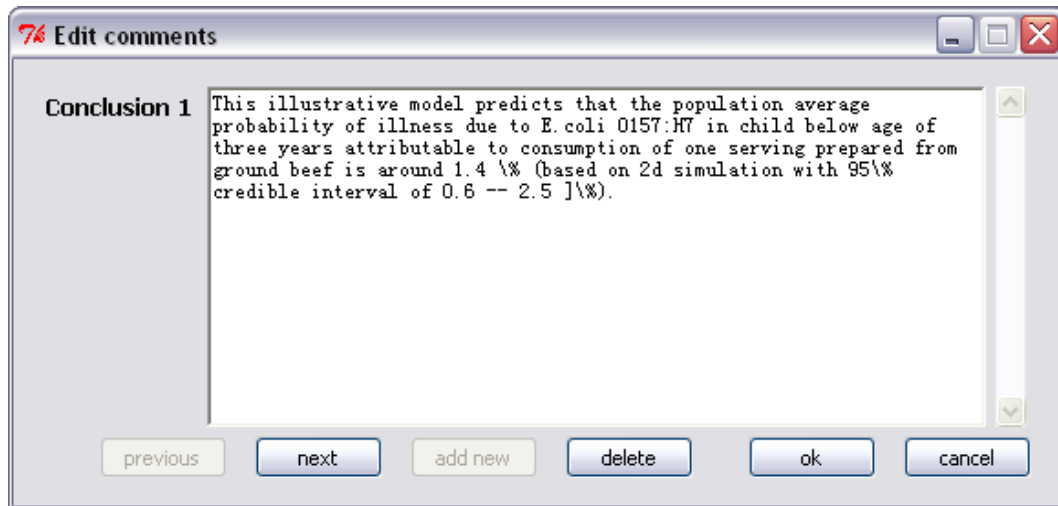
4.5. Get topics from other models

With this menu point you can import properties from other models. After selecting this menu point, you are prompted to choose from which model you want to import a topic. You can then choose what to import:

```
Choose the topic that you wish to import into the current model
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
1    Get basic descriptions
2    Get general model uncertainties
3    Get validations
4    Get comments
5    Get conclusions
6    Get glossary
7    Get abbreviations
8    Get concept graphs
9    Exit dialogue
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
```

Selecting one of those menu items will replace all current entries for the selected topic.

> **Note:** Care should be exercised when importing model topics as these changes cannot be made undone. **!**

Get basic descriptions: Choosing this option, the reference to the main document (**mainDoc**) as well as the explanations about the background of the model (**Background**), about the purpose and the objectives of the model (**Objectives**), about the scope and limitations of the scope (**Scope**) and about the model itself (**Description**) will be imported from the chosen model.

Get general model uncertainties: Hereby, the note and the description of the scoring system as well as all main categories, subdivided in their subcategories, including their explanations and scores will be taken from your chosen model.

Get validations: All names and explanations referring to validations will be imported from your chosen model.

Get comments: All comment sections will be imported from the chosen model.

Get conclusions: All conclusion sections will be imported from the chosen model.

Get glossary: All elements (names) of current glossary including their explanations will be replaced by the ones of your chosen model.

Get abbreviations: All abbrevations (like cfu for colony forming units) will be imported from the chosen model.

Get concept graphs: All concept graphs of your chosen model will be taken into the model which is to modify.

You can view all modifications of the topics in 4.2.

4.6. Model items

4.6.1. Edit available items

To edit a model's items, choose `View/edit/execute model` from the main menu and select your model in the following dialogue. Select `Model items` in the next dialogue. You can edit an available item by selecting `Edit available items`. This will bring up a list of current items. Choosing one will open a popup window where you can view and edit the properties of the corresponding item.

Here is a list of the item's properties:

- **Name:** The short, mathematical name of the item. Note that this name must be unique among all the items of this model. Only names of new added items can be edited - not already existing ones.

- **Title:** This is the item's detailed name.

25

- **Role:** This dropdown list lets you select if the item explains uncertainty, variability, both, if it is an outcome function or if it is not defined (nd).

- **Type:** The item's type. Refer to annex B for an explanation of each item type.

- **Stratum:** This optional field is only required for stratified `numv`, `mcrv` and `fnrv` items (see B). A stratified item can be generated by entering its (existing) stratum identifications item.

- **Report part:** The part which the item belongs to. An item must belong to one part, and cannot belong to more than one part. It is possible to select no part at all if the item should be non-assigned.

- **Explanation:** This is an elaborate explanation of what the item describes.

- **Unit:** The unit of the item, e.g. h if the item represents a time in hours. Moreover, the unit of observation (e.g., herd, animal, product) can be defined in this field.

- **Plausible min and max:** These two fields let you specify a plausible range in which the item's value can possibly lie. This field can only be filled in if the item has `mcrv`- or `mxrv`-type.

- **Dependent items:** This is a comma separated list of the items which depend on the current item. For example, if the item `ct` (concentration after processing) depends (among others) on the processing time `t`, the item `t` has `ct` in this line. This field is read–only. If you wish to change the dependency structure, go to `Edit item`.

- **Scores:** A space–separated list of score values (ranging according to a scoring system (for example from 0 to 3), one for each assessed criterion of uncertainty.

- **Assumptions:** This field contains the assumptions underlying the item.

- **Remark:** This field contains noteworthy comments on the item.

- **Reference:** Provide one or more bibliographic references. BibTeX–style is supported.

You can add a new item and specify its properties by choosing `Add new item` and remove items with `Remove item`.

To copy and import items from other existing models (which first have to be imported into the `rrisk` session), choose `Get items from other models`. You will be shown a list of available models — choose the one from which you want to copy an item.

Note that if you try to copy an item whose name is already taken by another item in your model, you will be prompted to confirm your import. If you do so, the new item will be appended to the end of your current item list.

- **Get a choice of items:** This will print a list of the model's items, of which you can choose the items you wish to import.

- **Get a choice of parts inclusive items:** This will print the model's parts and lets you import a whole part with all corresponding items.

- **Get all items from part (without part definition):** This prints out the model's parts, as above. However, it will only import the items, without the part and its defintion.

- **Get all items:** Choose this option to import every item stored in the model.

4.6.2. Get items from other models

Here you can import items from other models saved in your current `rrisk` session. Selecting this menu point will ask you to choose a model from which you want to import items, and

then present you a sub-menu:

Get a choice of items: This point will show you a list of the chosen model's items. Entering the according number will import that item. If you are done, choose `Exit dialogue`.

Get a choice of parts inclusive items: This point will show you a list of the chosen model's parts. Selecting one part will import that part and all according items into your current model.

Get all items from part (without part definition): This point will show you a list of the chosen model's parts. Selecting one part will import and all according items–but not the part–into your current model.

Get all items: This option will import all items of the specified model into your current model.

4.6.3. Add new item

Here you can specify a completely new item. You will be able to specify the same properties as explained in section 4.6.1. You can find more explicit explanations in B.

4.6.4. Remove item(s)

If you wish to delete specific items from your model, enter here and choose the according item in the following dialogue by typing its corresponding number. You can also delete several items simultaneously by entering their corresponding numbers,

- divided by a space (e.g. `1 3 4`)

- written as an R-vector (e.g. `c(1,3,4)`)

- using the colon if you wish to choose an ascending or descending series of numbers (e.g. `1:5` instead of 1 2 3 4 5 and 5:1 instead of `5 4 3 2 1`) or

- typed as a composition of the above possibilities (e.g. `9 7 8 c(1:5,10)`).

Make sure that, after deleting some items, you check the definition, full and relax commands of the remaining items!

4.6.5. View items

Choose this option to get a list of all items and select the item you wish to view.

4.7. Load reference list

This option lets you import a reference list (which should be a BibTeX–file in .bib format) from your hard disk.

> **Note:** After you imported the reference list, it is saved within the R model. You will not need the external .bib file anymore.

!

4.8. Load/edit concept graph(s)

This option lets you import a .png image file of a concept graph (see section 2.1). This graph will not be generated by `rrisk` and has to be provided by the user. It should show a rough sketch of the model concept.

`rrisk` will pop up a window where you can see the definitions for the current graphs. You can skim through them with the *previous* and *next* buttons. To add a new graph, skip to the end with the *next* button, and click *add new* afterwards. You should provide a legend in the text field, and use *load picture* to import a .png file. Press *OK* to confirm your changes.

4.9. Export model objects and definition

This option lets you export various items of the model:

- **Export model definition template (.Rmdt file):** This option saves your model and its properties in a .Rmdt file. Note that this option does not export model related data sets, BibTeX reference lists and graphics. You will have to use this menu's following options to export those properties of the model.

- **Export scoring system definition template (.Rsdt file):** This option lets R save you current scoring system as a .Rsdt file, a sort of text files. Having decided for this menu item, choose a folder where you wish to save your file.

- **Export scoring system as R object (.Rdata file):** Hereby, a .Rdata file containing the same information of the scoring system as the .Rsdt file will be saved in a folder you must choose.

- **Export model related data sets (.txt file):** If some of your model's items are data sets, this option lets you export this data as a .txt file.

- **Export reference list:** If your model has a reference list, you can export this list to a .bib–file to your hard drive.

- **Export model graphs:** With this option you can export all concept graphs the user provided for the model as .png files, as well as the network graph and graphs of the model's uncertainties as .pdf files.

  Note: If you want to save the whole model instead of single parts, please use "Save model" (see section 4.12.3). $!$

4.10. Scoring system

Choose this menu point to set or edit the model's scoring system. You can set the current scoring system to empty, or import a scoring system from the list of available systems.

Edit scoring system:

After setting at least one empty scoring system (if no scoring system already exists), you are able to choose that option. In this case, you are getting the following window:

Depending on your desires, fill in the fields according to the terms on the left.

4.11. Change order of model topics

Having chosen this option, you get to the following submenu:

```
Choose the model topic where you wish to edit sequencies
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
1    Order of authors
2    Order of parts
3    Order of items
4    Order of uncertainties
5    Order of validations
6    Order of comments
7    Order of conclusions
8    Order of glossary
9    Order of abbreviations
10   Order of concept graphs
11   Exit dialogue
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
```

Depending on which sequence you want to re-sort, press the corresponding number. Afterwards, you are presented that sequence in a certain numbered sort of order. There are three possibilities to sort:

- Type the single numbers, divided by a space! For example: 1 8 0 1 9 2

- Type a vector c(. . .) containing the single numbers, divided by a comma! For example: c(1,8,0,1,9,2)

- If you wish to take a part of the given sequence, you can use the colon as an abbrevation. For example: Type 1:3, if you wish to adopt the first three items in the same order. If you prefer the first three items in the reverse order, type 3:1.

Then a positive reply appears. You are allowed to mix the entry types, too: For example, 2 3 1 6:4 c(8,9,7) ist possible.
However, the order of items cannot be edited, because it might be important due to the possible dependence of the items on each other.

4.12. Running the model and accessing its results

To access the functions to run your model and access its result, choose `View/edit/run model` from the main menu. The relevant menu points for running the model and viewing its results are as follows:

4.12.1. Run model

Choose this option to execute the following menu items:

> **Note:** Running the model involves Monte-Carlo simulation and generation of graphical and numerical results. Various settings that affect these procedures can be changed as described in section 4.4.3.

> **Note:** Running the model requires that at least one item is defined as a model outcome function (OF). If more than one item is declared as OF, one of them must be chosen as the main OF.

After running the model, the tabulated results of the quantiles of the OFs are printed to the R console. Comments (see section 4.4.11) and conclusions (see section 4.4.12) should then be formulated or edited.

Run model (1d) This option launches a standard simulation (one dimension) for combined assessment of variability and uncertainty. The default total number of iterations is `N` = 1000. The OF is established `N` times by drawing `N` random numbers from the specified `mcrv`-items.

Run model (1d+2d) This option launches a two-dimensional simulation for separate assessment of variability and uncertainty. In 2d simulation, `N` random numbers are drawn from `mcrv`-items representing variability, while only one single random number is drawn from `mcrv`-items representing uncertainty or variability and uncertainty combined. This procedure is repeated by default `N2d` times for the 2nd dimension of the 2d simulation. Thus, the 2d simulation yields a total number of `N`×`N2d` results of the OF. The `N2d` repeats for the 2nd order simulation are used to establish uncertainty intervals of the empirical cumulative density function (ECDF) of the OF.

Show outcome function(s) Histograms, convergence graphs and cdf graphs of the outcome functions will be presented.

- The histograms show the distribution of each outcome function using `N` iterations (variability and uncertainty combined).

- A graph is generated showing the median value of the outcome function (OF) established using cumulative iteration numbers. Comparing this trace wit the tolerance limits should be used to assess the convergence of the model. Convergence is reached if the median OF remains within the tolerance limits.

- The ECDF (empirical cumulative distribution function) graph shows the ECDF of the main OF as a line. The 95% uncertainty envelope obtained by 2d simulation is overlaid.

Run sensitivity analysis In this option you can choose one out of four types of sensitivity analyses, each resulting in graphical output.

- The tornado chart of the effect of stochastic (`mcrv`) items on the OF shows by default the correlation coefficients using rank transformed input and output (OF) data. The left figure refers to the full risk model as it was used to estimate the outcome function whereas the right figure refers to a relaxed model, where all distributional assumptions were replaced with uniform distributions spanning the absolute plausible ranges of each input parameter. Items representing uncertainty are shared in lightblue.

- Nonparametric regression: This plot illustrates the relationship between the selected "uncertain" `mcrv`-item and the OF using a smoothed regression line. A shaded confidence band of the regression line is overlaid. One plot is done using `N` data points from the standard simulation and another plot is based on `N×N2d` data points from the 2d order simulation. The regression line of the latter my appear waved as a result of the randomly chosen support points for 2nd order simulation. Enter `?npreg` in the R console to obtain more information.

- Traffic light matrix of mcrv-items: This summarises the numerical uncertainty scores for all model items. For further explanation, see the Demo model report.

- Regression tree: This plot is useful to analyse the effect of model input `mcrv`-items on the OF item by choosing sequentially that input variable and respective cutpoint on its scale, that results in the best separation of the OF item into two distinct groups (e.g., low and high risk). This process is repeated, yielding a sequence of input variables sorted by their effect on explaining the remaining variance after the previous split. Enter `?tree` in the R console to obtain more information.

Reset simulation results All simulation results will be deleted. This may be useful before saving and exporting the model to save space.

Close all graphic devices All opened graphics windows will be closed.

4.12.2. Create model report

After having run the model, you can create and save the report. Choose the folder in which you want to save the report. Furthermore, enter whether you are updating or whether the current model report is for release:

```
 Please choose version update
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
1       No update
2       Minor update
3       Major update
4       Release model
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
```

Your choice will result in the following effects:

No update: On the cover of the model report, nothing will be changed.

Minor update: The minor draft version number of the model will be increased by one.

Major update: The major draft version number of the model will be increased by one.

Release model: The water-mark will disappear and the report's state changes from draft to release.

If LaTeX runs into any errors while compiling the document, you will get an error message here. In that case, please re-check your model definitions and syntax for any errors and retry afterwards.

> **Note:** The most likely reason for failure in compiling the report pdf is the incorrect use of the (optional) LaTeX language in the free text fields. !

4.12.3. Save model

This option will save your model to the hard disk. It will be saved as an R object, in the `.Rdata` format. This file will contain all the info about the model, including possible data sets it contains, concept and networks graphs, etc.

If you want to exchange your model with a colleague, it will suffice to transfer only this one file.

5. Known limitations and further extensions

`rrisk` is a project under development. A few functionalities are under evaluation and therefore disabled. Moreover, we have noticed some areas for improvements.

- It should be possible to define a new `mcrv`-item using a correlation coefficient defining the correlation between the new item and an existing `mcrv`-item.

- Existing code for generating copula models within `rrisk` should be integrated into the documentation scheme and made available.

- The empirical cumulative density function (ECDF) of the outcome function (OF) item in two-dimensional simulation should be established using the complete set of results (1st and 2nd order simulation).

- The correlation coefficients of `mcrv`-items and the OF item should be visualised in the matrix plots for uncertainties in stochastic model parameters.

- The two-dimensional simulation takes considerable time for large models. The sampling algorithm should be improved or outsourced.

The following areas were identified as priorities for further development.

- Development of a library for user-defined functions.

- Further development of a library for Bayesian models.

- Further development and implementation of sensitivity analysis methods.

- The outcome function should be evaluated for high percentiles of plausible ranges of uncertain `mcrv`-items items as a "stress test".

- Integration of mapping and geographical information system functionalities.

6. Glossary and abbreviations

Glossary

Items Data set or element or the `rrisk` model, which contains numerical input or evaluated functions.

Parts A set of model items that describe one given subsystem in the real world risk scenario. Parts can also be used to structure the assessment according to its components.

Variability (Syn. aleatory uncertainty, irreducible uncertainty, inherent uncertainty, objective uncertainty, stochastic uncertainty) Concept to describe the true variability of real world features, phenomena or processes.

Uncertainty (Syn. epistemic uncertainty, reducible uncertainty, subjective uncertainty) Concept to describe our ignorance of real world features, phenomena or processes.

Model uncertainty (MU) Ignorance about the correct mathematical representation of a real world feature, phenomena or process. In the context of `rrisk` analysis, MU is handled by stratified evaluation of the outcome function.

Parameter uncertainty (PU) Ignorance about the correct mathematical representation of a population parameter. In the context of `rrisk` analysis, PU is represented using Monte Carlo random variates.

Outcome function (OF) One or more evaluated expressions that qualify the relevant endpoint(s) of the risk assessment.

Two-dimensional simulation Monte Carlo method for differentiating the effect of variability and uncertainty in qualitative risk assessment. Also known as second-order modeling.

Iteration One realisation of random numbers drawn from each of the specified sampling distributions for probabilistic (stochastic) model input quantities and used to establish one value of the outcome function.

Full model Terminology in `rrisk` for the probabilistic (stochastic) model, which is the basis for the risk estimation.

Relaxed model Terminology in `rrisk` for the probabilistic (stochastic) model, which is identical with the full model except that all distributions that reflect uncertainty are replaced with uniform distributions spanning the absolute plausible boundaries for the given quantity provided by the user.

Abbreviations

N1d	Number of iterations for Monte Carlo simulation (1st dimension)
N2d	Number of iterations for Monte Carlo simulation (2nd dimension)
data	item type: data
stid	item type: stratum identification
stdi	item type: stratum distribution
strv	item type: stratum random variate for assigning each iteration to one of the defined strata
numv	item type: numerical value(s)
mcrv	item type: Monte Carlo random variate
fnrv	item type: Function of mcrv and other item(s)
corv	item type: correlated random variate
mxrv	item type: mixture distribution variate
rsrv	item type: bootstrap (distribution of mean and standard deviation) random variate
GAM	generalized additive model
v	role of a model item to represent variability
u	role of a model item to represent uncertainty
uv	role of a model item to represent uncertainty and variability
OF	role of a model item to represent an outcome function
pdf	probability density function
cdf	cumulative probability density function
2d	two-dimensional

A. Technical specifications of `rrisk`

The technical specifcations are given in Tab. 2 (p. 36). `rrisk` is a project under development and it is likely that that the range of functionality will grow. Please check the website for updates.

Table 2: Technical specifications of `rrisk`.

Availability	`rrisk` is distributed in form of an R-package with the name *rrisk* and can be obtained from the CRAN repository (once available). The package is also available from the BfR homepage (short: http://v28k.qr.ai)
Building blocks	model documentation, uncertainty documentation, model structure consisting of parts and items, simulation tools, tabulated outcomes, graphical outcomes (see details below)
Model documentation	*On the level of the model* auhors, background, objectives, scope, description, concept graph, dummy answers, glossary, list of abbreviations, list of references *On the level of the items* type, name, part, definition, role, plausible minimum, plausible maximum, unit, uncertainty scores, explanation, assumptions, remark, reference
Uncertainty documentation	*Model uncertainty categories* maximum 20 main or sub-categories of general model uncertainties; default system adapted from WHO [2008, 2009] *Model uncertainty assessment system* maximum 12 criteria; default system adapted from Van der Sluijs et al. [2005] *Uncertainty assessment of model items* data, parameters, numerical values, functions, etc. are assessed for uncertainty and knowledge base; default system as above
Model structure	unlimited number of parts and items; supported types of items are data, indexing items, numerical value(s), Monte-Carlo random variates, functions, Bayes domain joint posterior distribution, correlated random variate (check update), mixture distribution item, resampling item
Simulation tools	*Monte-Carlo simulation and distribution fitting to data or percentiles (*)* Bernoulli, binomial*, discrete, geometric, hypergeometric, multinomial, negative binomial*, Poisson*, beta*, beta Pert, Cauchy, chi-square*, chi-square, non-central, exponential*, F*, gamma*, logistic*, lognormal*, Normal*, truncated Normal, t*, triangular, uniform, Weibull* *Two-dimensional simulation* Algorithm: 1. Draw n_1 realisations from `mcrv`-distributions for input variables reflecting variability (Role `v`) and draw only one single realisation from `mcrv`-distributions for input variables reflecting uncertainty (Role `u` or `uv`). These steps refer to the first dimension of the simulation. 2. Evaluate the main outcome function OF using the data in step 1, resulting in n_1 estimates in the first dimension. 3. Repeat n_2 times the steps 1 and 2 and collect the OF results in the two-dimensional $(n_1 \times n_2)$ matrix. *Bayes domain joint posterior distribution* Pre-defined functions are made available that allow the estimation of inter-related parameters using Markov chain Monte-Carlo technique.

Continuing on the next page

36

Quality assurance	*Entry validation* Context-sensitive plausible ranges and number of values for numerical input (score values, model parameters, numerical model inputs), checks for illegal name duplicates
	Import data validation Summary statistics given for each variable, check and active confirmation by user is required
	Version control Draft and release mode, each with major and minor update numbers
	Change log Any change of the model in major updates of models is logged (version, file name, operator, data field, old and new entry)
	User guidance context sensitive menus; all entries checked for plausibility
	Review and revision Comprehensive model report including logged changes of released models, indication whether source file was protected against editing
Graphics	*General model characterisation* Network graph of model items, traffic light matrix for general model uncertainties and knowledge base
	Distribution fitting histogram, density plots, qq-plots, pp-plots, cdf-plots (fitting to percentiles)
	Outcome function (OF) histograms of all OF, empirical cumulative probability function with 95% uncertainy envelope based on 2d simulation
	Sensitivity analysis Tornado chart, traffic light matrix for uncertainty, knowledge base and sensitivity effect on OF, Generalized Additive Models (GAM) plots for nonparametric regression, Binary recursive partitioning (classification and regression tree, CART)
Packages	`rriskBayes` Collection of functions for fitting Bayesian models. The functions can be used as stand-alone application or launched during an `rrisk` session; available on CRAN
	`rriskDistributions` Collection of functions for fitting distributions to given data or by known quantiles. Two main functions `fit.perc()` and `fit.cont()` provide users a GUI that allows to choose a most appropriate distribution without any knowledge of the R syntax. Note, this package is a part of the `rrisk` project; available on CRAN

End of table

37

B. Item types

`rrisk` differentiates various item types, which represent the building blocks for a risk model. This section provides detailed technical guidance on the use of the item types. First, we want to use `Demo Model 1` in the examples of the following item descriptions. Therefore, as already explained in 2, I duplicated the default `rrisk` Demo Model 1 and called its copy "`demo model 1 new`".

Choose `View/edit/run model` in the main menu, then choose any model containing items, go to `Model items` and, at last, `Add new item`. What you will get, is explained in 4.6.1.

The following item types are offered:

B.1. Data item (`data`)

A data item contains a data set (e.g. written as a table) and its description according to the `rrisk` documentation style.

If you want to upload your data file as a data item, you have to decide for example (depending on which data sets are available at first)

```
Choose the data set you wish add to the current model item
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
1   NorthSouth (rrisk data set)
2   sdata (rrisk data set)
3   zipdata (rrisk data set)
4   load user-defined data set
5   Exit dialog
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
```

A data item should contain a data set. The data set may have any number of columns (variables) and rows (observations) and should contain unique variable names as headers.

The data item may contain no data although this is not recommended. A reason may be that you want to document the data set using the data item without actually uploading the data into the model. In this case you cannot build any new items using mathematical functions that require data access. If this is what you want, choose `Exit dialog`.

Choose the first or the second option to import specific data which are provided by the `rrisk` package. You may do this for testing purposes.

Choose the third option to select a new data set, for example some `.txt` or `.csv` data file. In the next two examples, the item's name and type are entered:

Type its name in the name field and choose `Data item (data)` in the type field. Fill out all relevant information including the uncertainty scores.

Data item based on rrisk data set: If you wish your data item to contain a `rrisk` data set, choose one of the first three options.

> **Example:** Let's call our new item `dataitem1` and let's take "`1  NorthSouth (rrisk data set)`". Then your loaded data set will be displayed in a pop-up window

> and in the console; in addition, you find the first rows of the data set and the summary statistics in the console:

```
          N  n k
North 1500 12 5
South  850 12 0
...

Summary statistics:
      N             n            k
 Min.   : 850   Min.   :12   Min.   :0.00
```
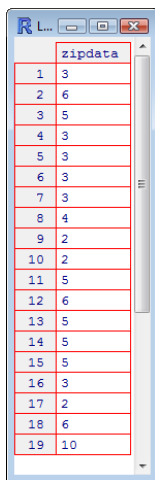
```
1st Qu.:1012    1st Qu.:12    1st Qu.:1.25
Median :1175    Median :12    Median :2.50
Mean   :1175    Mean   :12    Mean   :2.50
3rd Qu.:1338    3rd Qu.:12    3rd Qu.:3.75
Max.   :1500    Max.   :12    Max.   :5.00

Data set has been successfully added to the current model item.
```

Data item based on user-defined data set: In this case, an extra window will be opened for you to select a data set, for example a `.txt` data file.

Example: Let's create a new data item. Call it `dataitem2`:
Suppose you want this new data item to have the `rrisk` data set `zipdata.txt`, which is saved on your computer. So choose "4   load user-defined data set" and find your data set. Then your loaded data set will be displayed in a pop-up window

and in the console; in addition, you find the first rows of the data set and the summary statistics in the console:

```
   zipdata
1        3
2        6
3        5
4        3
5        3
6        3
...

Summary statistics:
     zipdata
 Min.   : 1.0
 1st Qu.: 3.0
 Median : 4.0
 Mean   : 4.2
 3rd Qu.: 5.0
 Max.   :10.0
```

At the end of both examples, the following menu appears:

```
Do you want to save this new item?
```

```
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
1   yes
2   no
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
```

If you want to save this new item, choose 1, otherwise 2.

Note that your data set must be existent in the form `read.table(file, header=TRUE)` in order to be able to be read in. For example, if you only have "`zipdata.rda`", enter `data(zipdata)` outside your **rrisk**-session (before starting **rrisk** or after closing **rrisk**) and use

```
> write.table(zipdata,file="zipdata.txt",header=TRUE,header=TRUE,row.names=FALSE)
```

to get "`zipdata.txt`".

B.2. Documentation of user–defined function (`fdoc`)

Functions written in R-syntax can be declared as part of the model definition and uploaded to the model documentation. Describe your function term an extra window that pops up. The generic format for writing functions in R is

```
myfun <- function(...) {
    code
    return(.)
    }
```

where `myfun` is the name of the function and "`...`" are function arguments. It is recommended to use the item name as name of the function. The last line of `code` will typically evaluate an expression, the results of which are returned by the function. Uploading the precise code to the documentation will ensure that the report covers the actual methods used. The code should be annotated following good programming practice. Moreover, the item explanation should contain a verbal description of what the function is supposed to do. For complicated functions, the explanation may also contain a description using pseudo code. Finally, close the extra window e.g. by saving the function.

> **Recommendation:** An `fdoc`-item is not required for functions provided by any existing R-package, because those are documented anyway. Outside your **rrisk**-session, you can check by `exists("myfdoc")` whether an item name is already used. $*$

Example: Let's create a new `fdoc`-item with the item name `getcirc`. After entering the item name, choosing `fdoc` as an item type, providing all other relevant information, clicking on `OK`, you can define a function.

Suppose you desire this function item to output the circumference, depending on the circle radius. In this case, the corresponding R-code is

```
function(radius){
circ <- 2*pi*radius
return(circ)
}
```

Note: The command `return(.)` is required for generating the output of the function.

A pop-up window appears and should be checked with 1 for a "yes".

```
Do you want to save this new item?
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
1    yes
2    no
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
```

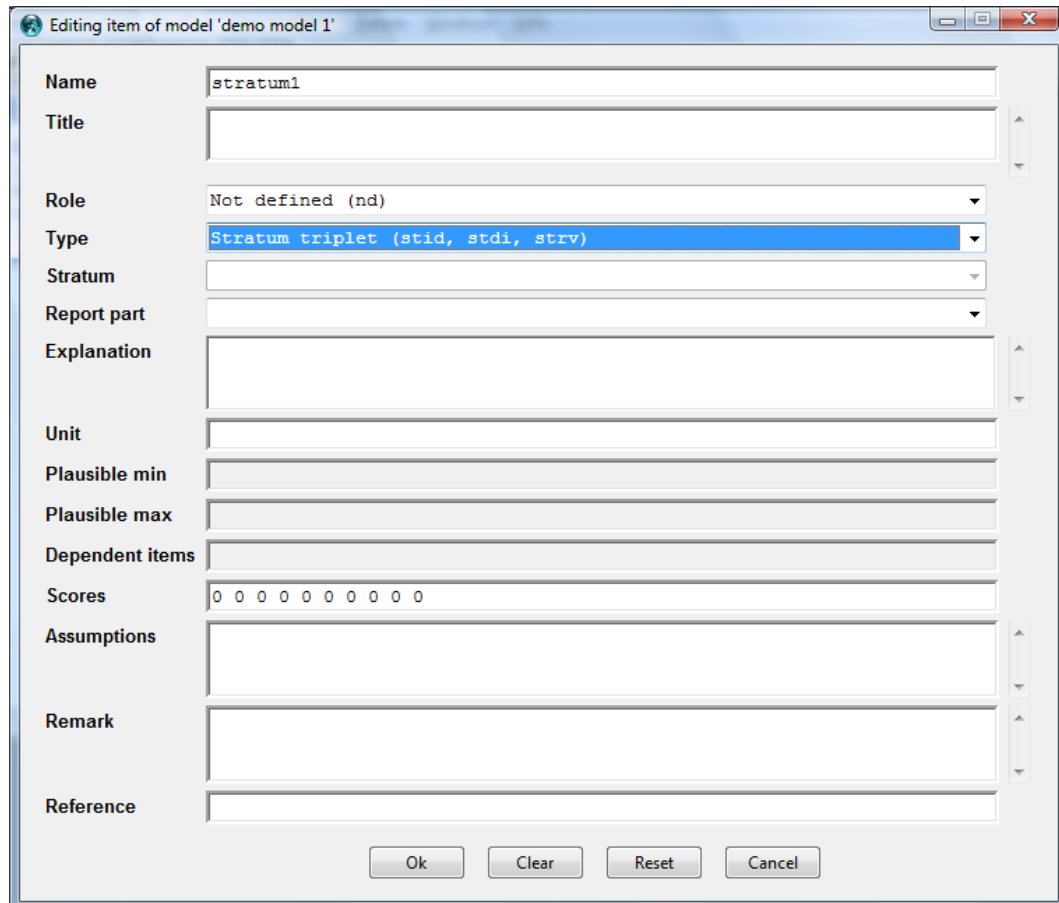If you want to save this new item, choose 1, otherwise 2.

Now, you can use the item to evaluate a new `fnrv`-item. In the example above, you may generate the new item `circumference` using the existing `mcrv`-item `radius` simply by

```
getcirc(radius)
```

B.3. Stratum identification (`stid`), stratum distribution (`stdi`) and stratum identification random variate item (`strv`)

A stratum triplet is generated to introduce stratification into the model. `stid`, `stdi` and `strv` contain the definition of strata, its relative weights (distribution) and a random variate with allocations into the defined strata, respectively.

Firstly, give your new stratum index a name and choose `stratum triplet (stid, stdi, strv)`:

If you want to save your created stratum index, you have to decide

```
Please choose the way to define stratum item
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
1   Weights derived from existing items
2   Weights entered manually
3   Exit dialog
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
```

Two examples are presented below, showing the first two options from above:

Example 1: Create a stratum item `stratum1` by deriving the weights from `dataitem1`. Then, choose "1 `Weights derived from existing items`" and put the row names of the data item into your stratum triplet as an identification (as an example of any expression delivering a character, single or as a vector):

```
Please enter names of stratum levels (should evaluate to required number
of character values)
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
rownames(dataitem1)
```

Then, take the first column of the table (to get the stratum distribution) by:

```
Enter weights of stratum levels (should evaluate to required number of
numeric values)
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
dataitem1$N
```
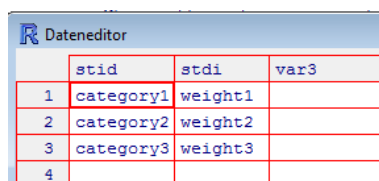
After this, the console will display the absolute numbers of the stratum distribution and the summary statistics of the whole stratum item:

```
Stratum weights:
[1] 1500  850

Stratum data frame has been evaluated to
   stid stdi
1 North 1500
2 South  850
```

Example 2: Create a stratum item `stratum2` by entering the weights manually. After, choose 2   `Weights entered manually`. You will get an extra window, containing a table:

and "`Please enter in the label names and the relative weights of the`
`strata in the first and second column, respectively`" appears in the console.
In the first column all stratum identifications should be written, in the second one all of their weights, which do not have to result in 1 after summing them up (R fits its sum to 1 and creates its distribution according to this automatically). So, type 1, 2 and 3 as weights in each box of the second column from top to bottom. Close this window after you have finished your entries, so the console will display:

```
Stratum data frame has been evaluated to
       stid stdi
1 category1    1
2 category2    2
3 category3    3
```

In contrast to the previous option, here you cannot import any imformation from other items.

At the end of boh examples, the following menu appears:

```
Do you want to save this new item?
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
1   yes
2   no
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
```

If you want to save this new item, choose 1, otherwise 2.
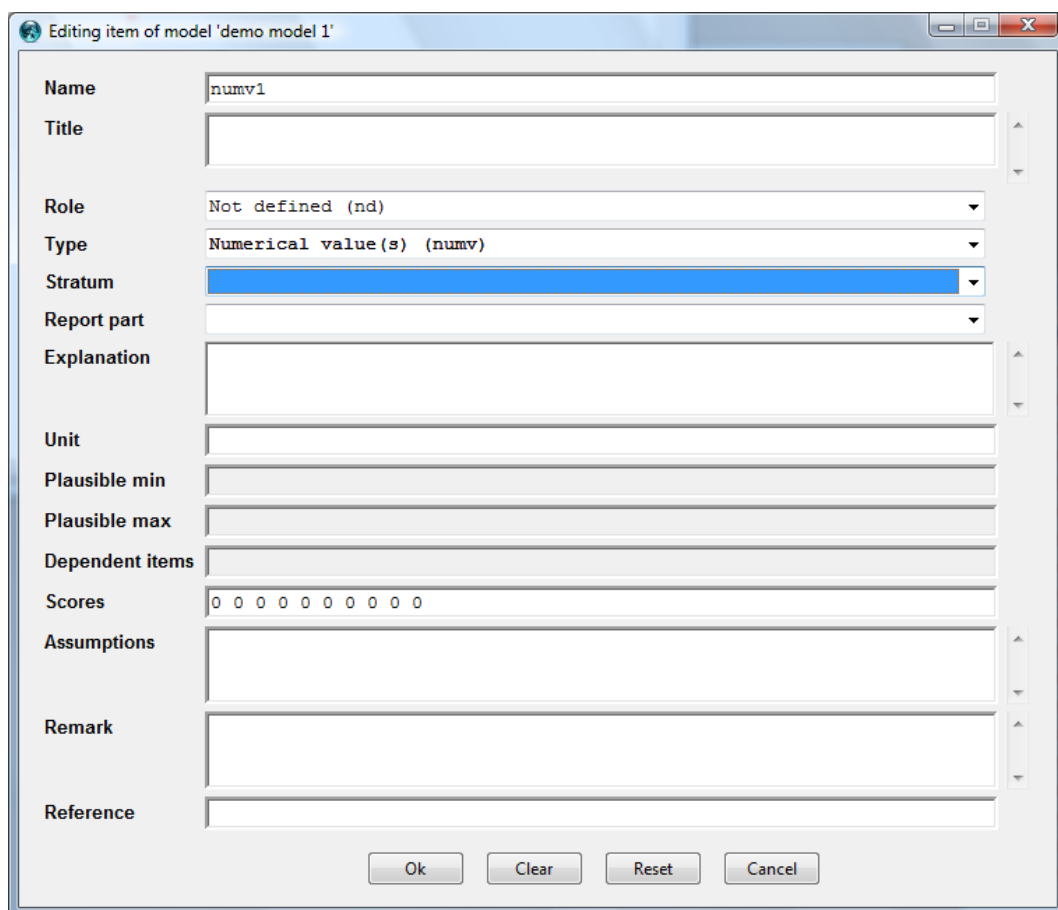
B.4. Numerical value item (`numv`)

A numerical value item is used to define a single value (in the general case) or a sequence of values (if it is stratified). Both numerical entries and entries of expressions (in R-syntax) are allowed. For example the value 2.0 (vector length = 1) could be entered directly as "2" (without quotes) or as "4/2" or as "mean(1:3)".
After choosing `numv`, you are asked:

```
Enter expression that could be evaluated to 1 numerical values(s)
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
```

For this, there are six possible kinds of entries:

Creating non-stratified items: In this case, the `numv`-item will have only one numerical value. As every item, each `numv`-item must be named and defined as an `numv`-item:

There are many possibilities for enter expressions:

Entering a value directly: Let's call this item `numv1`. Type any number, e.g. `2`. Then the console will display, that you entered `2` and that `rrisk` was checking this entry:

```
Entry evaluates to: 2
```

```
**************************************
Full command for numv1
**************************************


Dimension of test evaluation:


item@fullc= 2


Generated item value: 2


**************************************
Symbolical definition for numv1
**************************************


item@definition= 2
```

The result of the Relaxed command can be ignored for `numv`-items.

Using R-syntax in combination with a data set: Let's call this item `numv2`. Suppose you want to use `sdata`, because you are interested in the mean of the column which is named `dishsize`. So enter `mean(sdata$dishsize)`. Then you can read in the console:

```
Entry evaluates to: 48.84


**************************************
Full command for numv2
**************************************


Dimension of test evaluation:


item@fullc= mean(sdata$dishsize)


Generated item value: 48.84225


**************************************
Symbolical definition for numv2
**************************************


item@definition= mean(sdata$dishsize)
```

Here you can find both your entered expression and its evaluation.

Using R-syntax in combination with a random variate: Let's call this item `numv3`. Suppose you want to use the standard deviation of the `mcrv`-item `c0`, so type `sd(c0)`. (For example, `c0[1]` the first entry of c0, is a possible entry, too.) According to `numv2`, you will get a description of the evaluation, in which your entering expression and ist value is written:

```
Entry evaluates to: 1.98


**************************************
Full command for numv3
**************************************


Dimension of test evaluation:


item@fullc= sd(c0)
```

```
Generated item value: 1.983043


****************************************
Symbolical definition for numv3
****************************************


item@definition= sd(c0)
```

Creating stratified items: In contrast to the previous items, all of the following ones will consist of two values, because the stratum triplet `stratum1` from B.3 is used.

> **Recommendation:** If your `numv`-item is stratified, write the stratum's name after the
> actual `numv`-name, divided by a dot! For example: Call your item `mynumv.stratum1`
> instead of only `mynumv`, so that the kind of stratification can be seen immediately.

 *

The entry possibilities for stratified items are the same as for non-stratified items, in principle:

Entering the values directly: Let's call this item `numv4.stratum1`. In the pop-up window, choose `stratum1` in the stratum field and click on `OK`. `stratum1` causes that the created item contains two numerical values. Type single numbers written as an R-vector (e.g. `c(1,4)`), type an ascending or descending sequence using the colon (e.g.`1:2` or `2:1`) or use any combination of them. In this case, type `c(1,4)`. This expression and its evaluation will be visible in the description on the console:

```
Entry evaluates to:
[1] 1 4
```

```
...

***************************************
Full command for numv4
***************************************


Dimension of test evaluation:

item@fullc= c(1,4)

Generated item values (full command):
North South
     1     4


***************************************
Symbolical definition for numv4
***************************************


item@definition= c(1,4)


The structure of this description is the same as the ones of the previous examples.

\item [Using a data set and a random variate:]
Let's call this item \texttt{numv5.stratum1}. In the pop-up window, choose \texttt{stratum
\begin{verbatim}
Entry evaluates to:
[1] 48.842246  1.962387
...

***************************************
Full command for numv5
***************************************


Dimension of test evaluation:

item@fullc= c(mean(sdata$dishsize), sd(c0))

Generated item values (full command):
    North      South
48.842246   1.962387


***************************************
Symbolical definition for numv5
***************************************


item@definition= c(mean(sdata$dishsize), sd(c0))
```

If you are not sure what this description means, see the explanations of the previous examples.

Importing a vector from an existing item: Let's call this item `numv6.stratum1`. In the pop-up window, choose `stratum1` in the stratum field and click on `OK`. `stratum1` causes that the created item contains two numerical values. Suppose you want the stratified

item to contain the first column of the data set `dataitem1`. So, enter `dataitem1$n`. As in the other examples before, the console will display a description about you new item:

```
Entry evaluates to:
[1] 12 12
...


***************************************
Full command for numv6
***************************************

Dimension of test evaluation:

item@fullc= dataitem1$n

Generated item values (full command):
North South
   12    12



***************************************
Symbolical definition for numv6
***************************************

item@definition= dataitem1$n
```

At the end, the following menu appears:

```
Do you want to save this new item?
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
1   yes
2   no
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
```

If you want to save this new item, choose 1, otherwise 2.

> **Note:** Expressions in `rrisk` containing one or more `numv`-items of length > 1 (i.e. vectors) are evaluated element-wise. This means, the expression is calculating for each element of the vector(s). This is standard R-behaviour.

B.5. Monte–Carlo random variate item (`mcrv`)

A Monte-Carlo random variate item introduces stochastic variability into a model to reflect variability or uncertainty. Several discrete (Tab. 6) and continuous probability density functions (Tab. 7) are available for modelling in `rrisk`. The user is prompted to select the scale type (discrete or continuous)

```
Choose type of probability density
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
1      discrete
2      continuous
3      Exit dialog
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
```

and then the discrete/continuous distribution family or, alternatively, the option for fitting to data or percentiles (see below).

The following strategies (the first two ones are not possible for stratified items) may be considered for defining an `mcrv`-item.

1. Fitting to empirical data (only for continuous distributions): Select a variable of an uploaded data set and follow the instructions. Graphical (histogram and cumulative distribution plot, qq-plot, pp-plot) and numerical (logL, AIC, BIC, chi-square, AD, KS) support is given to assess the goodness of fit for a range of candidate distribution families. The distribution fitting uses the functions implemented in the `rrisk Distributions` package (see Belgorodski et al. [2012b] and Engelhardt et al. [2012]) and the parameters for the selected distribution were estimated by standard methods.

2. Definition of parameters by fitting to percentiles (only for continuous distributions): Follow the instructions and enter percentages and percentiles. Graphical (cumulative distribution plot) and numerical (tabulated percentiles) support is given to select a suitable distribution family. The parameters for the selected distribution are found by functions provided by the `rrisk Distributions` package (see Belgorodski et al. [2012b] and Engelhardt et al. [2012]).

3. Definition of parameters as direct numerical entries: Whole values or real values – depending on the nature of the model parameter – can be entered directly. The values should be documented and justified as part of documentation of the `mcrv` item. This method is not very flexible because the entries cannot be updated as a function of other items. On the other hand, the direct entry is recommended, if the parameter values are not used elsewhere in the model.

4. Definition of parameters as `numv` item: A `numv`-item or a function of existing items can be entered. This method is flexible because the entry is linked to another item or a function thereof.

5. Definition of parameters as `data` items: For example, the minimum, most likely, maximum and shape value of a beta Pert distribution could be generated as a `data`-item (using the NoHeader option; see Appendix B.1) saved as `PertPara`. The `mcrv`-item could then be generated using the function `rpert(PertPara[1],PertPara[2],PertPara[3],PertPara[4])` although this is not very efficient.

6. Definition of parameters as an `mcrv` item: Another `mcrv`-item can be entered. This option can be used to model empirical overdispersion, whereby the distribution parameter can be thought of a random variate with its own pdf (e.g., beta-binomial model). This option is also suitable if the pdf of the distribution parameter reflects parameter uncertainty and the outcome random variate reflects variability.

   Note: `rrisk` uses the Mersenne-Twister as random number generator, which is the default setting of R.

   Note: Existing simulation results are overwritten each time when the model is run.

   Note: See Section 4.12.1 for more details on simulation techniques.

Examples for non-stratified items: At the beginning of every example, a new `mcrv`-item has to be named and to be given an item type as well as - in contrast of most of the other items - a minimum and maximum value (e.g. 0 and 9):

In this paragraph, three possibilities for creating a non-stratified item are explained:

Fitting to data: Let's call this item `mcrv1` and enter 0 and 9 for minimum and maximum value. After clicking on `OK`, choose the continuous scale type of probability density. Then take "`Fit continuous distribution to data`":

```
Choose family of continuous probability density function
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
    1        Fit continuous distribution to data
    2        Fit continuous distribution to given percentiles
    3        beta (beta)
    4        beta Pert (pert)
    5        Cauchy (cauchy)
    6        chi-square (chisq)
    7        chi-square, non-central (chisqnc)
    8        exponential (exp)
    9        F (f)
   10        gamma (gamma)
   11        logistic (logis)
   12        lognormal (lnorm)
   13        Normal (norm)
   14        Truncated Normal (tnorm)
   15        t (t)
   16        triangular (triang)
   17        uniform (unif)
   18        Weibull (weibull)
   19        Gompertz (gompertz)
```

```
20      Exit dialog
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
1
```

Then you have a choice of data set like:

```
Please choose data set for fitting
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
1       sdata$dishsize
2       dataitem1$N
3       dataitem1$n
4       dataitem1$k
5       Exit dialogue
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
4
```

Take `sdata$dishsize` and you will get a table which contains the elements of your selected data set and you will be displayed a pop-window which shows graphical and numerical support, as described as the first strategy above:
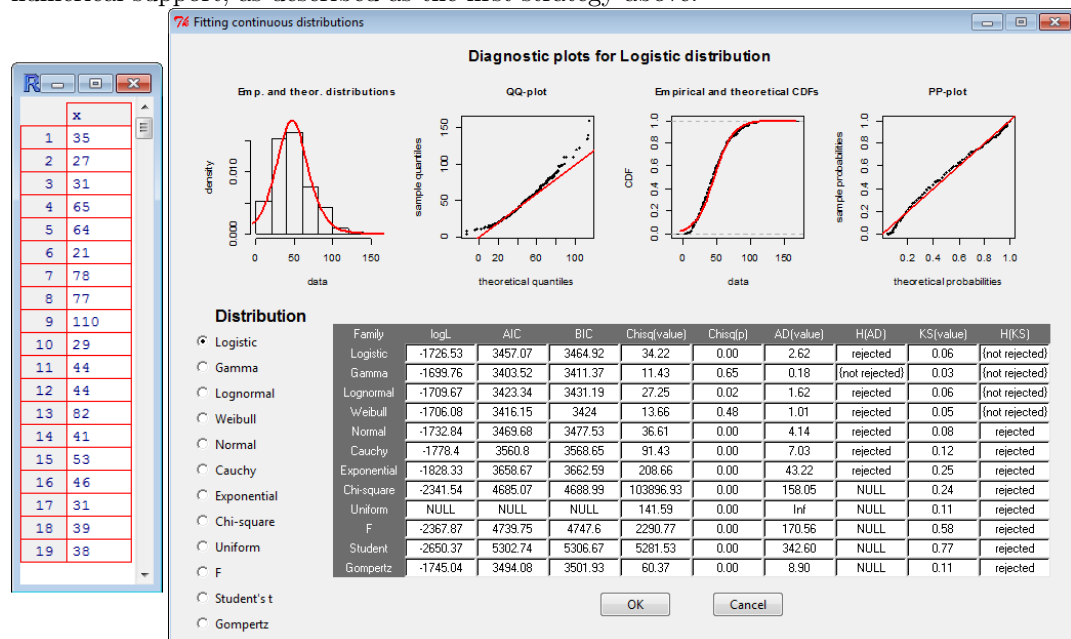
The four plots depend on which distribution has been chosen. These distributions are the possible ones for `mcrv1`. In our case, select the Normal distribution and click on `OK`. That also means, that the new item will be Normal distributed. Then the first six rows of your new random variate and its summary statistics - in relation to full command (independent from your minimum and maximum value) and relaxed command (i.e. fit to your minimum and maximum value) - will be presented on the console:

```
Chosen continuous distribution is:  Normal (norm)
Fitted parameters are:
    mean        sd
48.84225 24.88636
----------------------------------------------------------------------


**************************************
Full command for  mcrv1
**************************************


Dimension of test evaluation:  100 1
```

```
item@fullc= matrix(ncol=1,data=rnorm(n=1*rriskModel@settings@N,mean=48.842,
sd=24.886),byrow=TRUE)
```

Generated item values (full command):
```
      mcrv1
1 77.936785
2 21.793187
3  3.398098
4 73.610583
5 48.546808
6 33.919648
...
```

Summary statistics of item values (full command):
```
     mcrv1
 Min.   :-11.49
 1st Qu.: 33.11
 Median : 51.81
 Mean   : 49.51
 3rd Qu.: 66.93
 Max.   :124.89
```

```
****************************************
Symbolical definition for mcrv1
****************************************
```

```
item@definition= norm(mean=48.842,sd=24.886); fitted to sdata$dishsize
```

```
****************************************
Relax command for mcrv1
****************************************
```

Dimension of test evaluation: 100 1

```
item@relaxc= matrix(ncol=1,data=runif(rriskModel@settings@N,min=0,max=9),
byrow=TRUE)
```

Generated item values (relax command):
```
      mcrv1
1 2.6945202
2 2.4926766
3 0.5544848
4 3.1978921
5 5.1933398
6 4.8152840
...
```

Summary statistics of item values (relax command):
```
     mcrv1
 Min.   :0.07333
 1st Qu.:2.22984
 Median :4.78384
 Mean   :4.57903
 3rd Qu.:6.88015
 Max.   :8.95753
```

Fitting to percentiles: Let's call this item `mcrv2`. After clicking on OK, choose the continuous scale type of probability density. Then take "`Fit continuous distribution to given percentiles`":

```
Choose family of continuous probability density function
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
1       Fit continuous distribution to data
2       Fit continuous distribution to given percentiles
3       beta (beta)
4       beta Pert (pert)
5       Cauchy (cauchy)
6       chi-square (chisq)
7       chi-square, non-central (chisqnc)
8       exponential (exp)
9       F (f)
10      gamma (gamma)
11      logistic (logis)
12      lognormal (lnorm)
13      Normal (norm)
14      Truncated Normal (tnorm)
15      t (t)
16      triangular (triang)
17      uniform (unif)
18      Weibull (weibull)
19      Gompertz (gompertz)
20      Exit dialog
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
2
```

You will get this pop-up window:
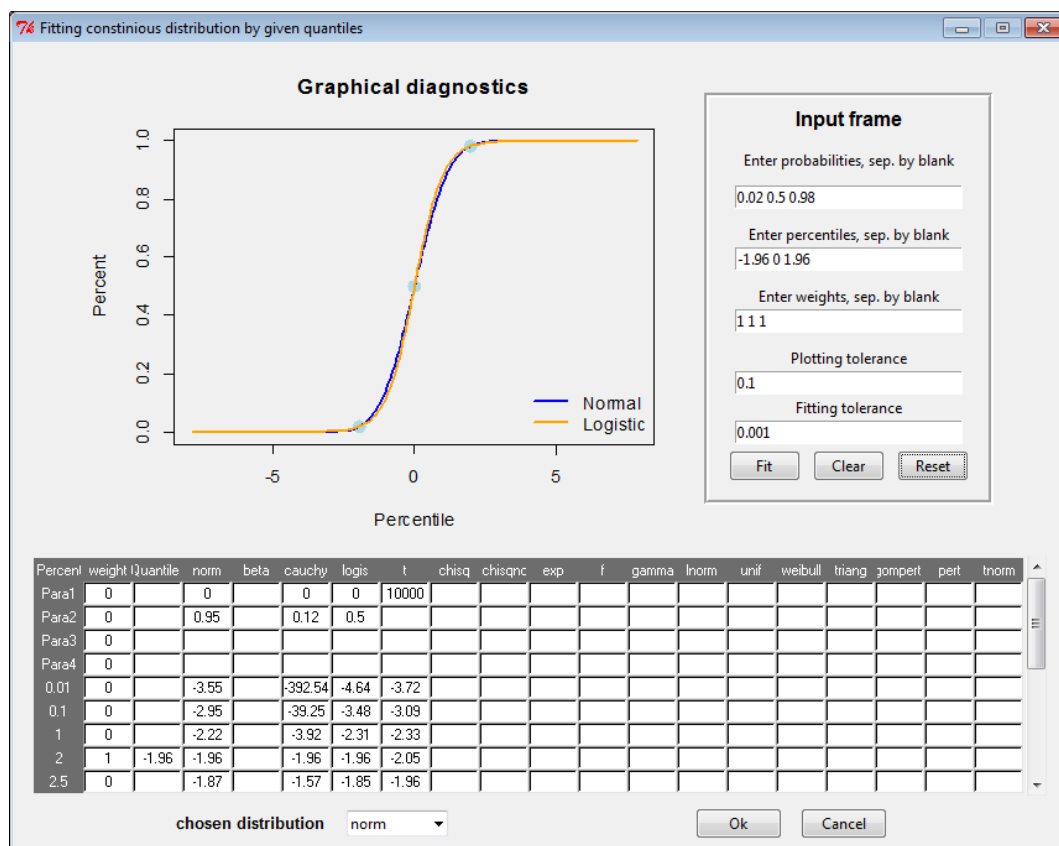
It indicates three parts:

1. The table contains all parameters and all percentiles of those distributions which can be used to the given percentiles and probabilities. The new created random variate can have one out of only these distributions.

2. The input frame is the only area to edit. You can enter your delivered probabilities and its according percentiles. If there is no cdf which maps the entered percentiles to the entered probabilities, you will get an error message. The percentiles should be weigthed reasonably, because the elements of the `mcrv`-item will get those values according to these weights. You can determine the decimal place of the dots in the cdf plots (see below) and of the cdf values corresponding to the entered probabilities and percentiles. Click on `Clear`, if you wish to delete all entries, and click on `Reset`, if you wish to get the entries from the beginning. However, if you want the plot and the table to show those entries, you have to click on `Fit`, again.

3. The graphical diagnostics illustrate the cdf-plots of all distributions which fulfill the conditions from the input frame. So at first, only the cdfs of the Normal and the logistic distribution map (`-1.96 0 1.96`) to (`0.02 0.5 0.98`).

On the bottom of this pop-up window, you can choose which distribution the new random variate should get. In this example, choose `t` and do not change the input frame entries. If you have finished entering/choosing all the settings as you wish, click on `OK` and the new random variate will be generated. Its first six rows and its summary statistics will be displayed on the console:

```
Chosen continuous distribution is:  Student's t (t)
Fitted parameters are:
   df
10000
----------------------------------------------------------------------

**************************************
Full command for mcrv2
**************************************

Dimension of test evaluation:  100 1

item@fullc= matrix(ncol=1,data=rt(n=1*rriskModel@settings@N,df=10000),byrow=TRUE)

Generated item values (full command):
        mcrv2
1 -0.5600379
2 -1.0845123
3  0.5248667
4 -0.0313868
5  1.5620064
6 -0.1923430
...

Summary statistics of item values (full command):
     mcrv2
 Min.   :-2.17353
 1st Qu.:-0.55389
 Median : 0.07665
 Mean   : 0.10263
 3rd Qu.: 0.81148
 Max.   : 2.39218
```

```
****************************************
Symbolical definition for mcrv2
****************************************


item@definition= t(df = 10000) fitted to quantiles q=(-1.96,0,1.96)
using p=(0.025,0.5,0.975)


****************************************
Relax command for mcrv2
****************************************


Dimension of test evaluation:  100 1

item@relaxc= matrix(ncol=1,data=runif(rriskModel@settings@N,min=0,max=9),byrow=TRUE)


Generated item values (relax command):
      mcrv2
1 5.8550798
2 2.1152467
3 8.7886917
4 3.6261012
5 2.0667414
6 0.9467193
...


Summary statistics of item values (relax command):
     mcrv2
 Min.   :0.03652
 1st Qu.:2.20803
 Median :4.74443
 Mean   :4.56055
 3rd Qu.:6.81854
 Max.   :8.99143
```

Discrete distribution: Let's call this item `mcrv3` and enter 0 and 9 for minimum and maximum value. After clicking on `OK`, choose the discrete scale type of probability density.

```
Choose family of discrete probability density function
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
1       Fit discrete distribution to data
2       Fit discrete distribution to given percentiles
3       Bernoulli (bern)
4       binomial (binom)
5       discrete (discrete)
6       geometric (geom)
7       hypergeometric (hyper)
8       multinomial (multinom)
9       negative binomial (nbinom)
10      Poisson (pois)
11      uniform discrete (udiscrete)
12      Exit dialog
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
5
```

> **Note:** The first two options of this submenu are still under evaluation, so you cannot
> work with them!   ⚠

A table in an extra window will appear:

Elements in the first column will be the (possible) values for the random variate and elements in the second column will indicate the probability of each value in the same row, i.e. the appoximate percentages of the values appearing in the new item. However, in both columns you can enter any numbers as you want, because R will fit the values according to your minimum and maximum value and the probabilities according to percentages automatically. After you have finished typing the numbers, close the table and you will get two descriptions of your new random variate:

```
Evaluating item values with predefined distribution parameters...
-----------------------------------------------------------------------


*****************************************
Full command for mcrv3
*****************************************


Dimension of test evaluation: 100 1

item@fullc= matrix(ncol=1,data=rdiscrete(n=1*rriskModel@settings@N,
values=c(2,4,6),probs=c(0.4,0.4,0.2)),byrow=TRUE)


Generated item values (full command):
  mcrv3
1      2
2      4
3      6
4      6
5      4
6      4
...


Summary statistics of item values (full command):
     mcrv3
 Min.   :2.00
 1st Qu.:2.00
 Median :3.00
 Mean   :3.48
 3rd Qu.:4.00
 Max.   :6.00



*****************************************
Symbolical definition for mcrv3
*****************************************


item@definition= discrete(values=c(2,4,6),probs=c(0.4,0.4,0.2))


*****************************************
Relax command for mcrv3
*****************************************
```

```
Dimension of test evaluation: 100 1

item@relaxc= matrix(ncol=1,data=rudiscrete(n=1*rriskModel@settings@N,
min=0,max=9),byrow=TRUE)

Generated item values (relax command):
  mcrv3
1     8
2     8
3     5
4     2
5     9
6     9
...

Summary statistics of item values (relax command):
     mcrv3
 Min.   :0.00
 1st Qu.:2.75
 Median :5.00
 Mean   :4.56
 3rd Qu.:7.00
 Max.   :9.00
```

The first one shows the absolute values you have entered in the table and the second one the modified ones according to your minimum and maximum value. Both present the first six rows and the summary statistics of the new item.

Example for a stratified item: Let's call this new item `mcrv4.stratum1`. As the previous three items before, it must have a minimum and a maximum value, e.g. 0 and 9, again. Furthermore, we want it to be stratified by stratum1:

Editing item of model 'demo model 1'

Name	mcrv4.stratum1
Title	
Role	Not defined (nd)
Type	Monte-Carlo random variate (mcrv)
Stratum	stratum1
Report part	
Explanation	
Unit	
Plausible min	0
Plausible max	9
Dependent items	
Scores	0 0 0 0 0 0 0 0 0 0
Assumptions	
Remark	
Reference	

Ok   Clear   Reset   Cancel

Having clicked on OK, choose the continuous scale type of probability density. Then take the t distribution:

```
Choose family of continuous probability density function
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
1       Fit continuous distribution to data
2       Fit continuous distribution to given percentiles
3       beta (beta)
4       beta Pert (pert)
5       Cauchy (cauchy)
6       chi-square (chisq)
7       chi-square, non-central (chisqnc)
8       exponential (exp)
9       F (f)
10      gamma (gamma)
11      logistic (logis)
12      lognormal (lnorm)
13      Normal (norm)
14      Truncated Normal (tnorm)
15      t (t)
16      triangular (triang)
17      uniform (unif)
18      Weibull (weibull)
19      Gompertz (gompertz)
20      Exit dialog
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
15
```

Now you have to enter two positive values; the t distribution needs one parameter, but
`stratum1` causes two vectors in one element. Enter a vector in R-syntax like `c(6,7)`. After
this, you are shown the first six rows and the summary statistics of your new random variate
- both twice: firsty without considering your minimum and maximum value and secondly
by considering your minimum and maximum value:

```
Entry evaluates to: 6, 7


------------------------------------------------------------------------
Evaluating item values with predefined distribution parameters...
------------------------------------------------------------------------


***************************************
Full command for mcrv4.stratum1
***************************************

Dimension of test evaluation: 100 2

item@fullc= matrix(ncol=2,data=rt(n=2*rriskModel@settings@N,df=c(6,7)),byrow=TRUE)

Generated item values (full command):
        North       South
1   1.66288826 -1.5315886
2   1.75535569  1.2803722
3  -1.86456918  2.2338640
4   0.49297826  0.2633083
5  -2.43681088  1.3064434
6   0.05057805 -2.2479533
...

Summary statistics of item values (full command):
      North              South
 Min.    :-2.43681   Min.    :-4.69674
 1st Qu.:-0.70280    1st Qu.:-0.63250
 Median : 0.00553    Median :-0.01299
 Mean    : 0.03031   Mean    : 0.03316
 3rd Qu.: 0.81508    3rd Qu.: 0.96157
 Max.    : 2.37704   Max.    : 3.76991



***************************************
Symbolical definition for mcrv4.stratum1
***************************************


item@definition= t(df=c(6,7))

***************************************
Relax command for mcrv4.stratum1
***************************************

Dimension of test evaluation: 100 2

item@relaxc= matrix(ncol=2,data=runif(2*rriskModel@settings@N,min=0,max=9),byrow=TRUE)

Generated item values (relax command):
     North     South
1 7.101761 8.9240271
```

```
2 7.990134 2.3129902
3 7.435449 3.7431572
4 4.367848 4.0615327
5 4.473519 7.2218838
6 3.501170 0.3881304
...

Summary statistics of item values (relax command):
     North            South
 Min.    :0.007479   Min.    :0.03789
 1st Qu.:2.360045    1st Qu.:2.39510
 Median :4.276462    Median :4.57362
 Mean    :4.475627   Mean    :4.59666
 3rd Qu.:6.610420    3rd Qu.:6.56362
 Max.    :8.856418   Max.    :8.92403
```

At the end of every example, the following menu appears:

```
Do you want to save this new item?
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
1   yes
2   no
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
```

If you want to save this new item, choose 1, otherwise 2.

B.6. Function item using Monte–Carlo random variates (`fnrv`)

A new item can be generated as a function of one or more `mcrv`-items. An `fnrv`-item always evaluates to a vector with length equal to the number of iterations (in `demo model 1 1000`) requested. As well as a `numv`- and an `mcrv`-item, a `fnrv`-item can be stratified, too.

Example 1: Let's create a non-stratified `fnrv`-item. Call it `fnrv1`:

After clicking on OK, you are asked:

```
Enter expression that could be evaluated to data frame consisting of 1 columns
and 1000 rows
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
```

As c0 is an mcrv-item consisting of one column and 1000 rows, it is an admissible input argument for fnrv1. So enter exp(c0); that means, the exponential function is applied to each component of the vector c0. Among others, the first rows of your new item and the summary statistics will be shown on the console:

```
**************************************
Full command for fnrv1
**************************************

Dimension of test evaluation:  1000 1

item@fullc= exp(c0)

Generated item values (full command):

         fnrv1
[1,]    5233.337
[2,] 161514.146
[3,]    1984.981
[4,]   10746.671
```

```
[5,] 129458.224
[6,] 166131.090
...

Summary statistics of item values (full command):
     fnrv1
 Min.   :        59
 1st Qu.:      6301
 Median :     24553
 Mean   :    135691
 3rd Qu.:     84109
 Max.   :  11710154


**************************************
Symbolical definition for fnrv1
**************************************

item@definition= exp(c0)

**************************************
Relax command for fnrv1
**************************************

item@relaxc= exp(c0)
```

Example 2: In the following, a stratified `fnrv`-item will be created. Name it `fnrv2.stratum1`:

After clicking on `OK`, you are asked:

```
Enter expression that could be evaluated to data frame consisting of 2 columns
and 1000 rows
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
```

In contrast to the previous, not a 1000×1 input vector, but a 1000×2 input matrix is expected. So use the 1000×1 `mcrv`-item `c0` again and type `cbind(exp(c0),log(c0))`: You will find `exp(c0)` in the first and `log(c0)` in the second column. Afterwards, the console will display the first six rows of your `fnrv`-item and its summary statistics, above:

```
Entry evaluates to:
          [,1]      [,2]
[1,]   4516.315 2.130070
[2,] 50614.526 2.382504
[3,] 87134.633 2.431436
[4,] 13935.013 2.255720
[5,] 12972.577 2.248192
[6,] 66376.906 2.407225
...


****************************************
Full command for fnrv2.stratum1
****************************************

Dimension of test evaluation:  1000 2
```

```
item@fullc= cbind(exp(c0),log(c0))
```

Generated item values (full command):

```
          North    South
[1,]   4516.315 2.130070
[2,] 50614.526 2.382504
[3,] 87134.633 2.431436
[4,] 13935.013 2.255720
[5,] 12972.577 2.248192
[6,] 66376.906 2.407225
...
```

Summary statistics of item values (full command):

```
     North             South
 Min.   :        21  Min.   :1.117
 1st Qu.:      6979  1st Qu.:2.180
 Median :     21742  Median :2.301
 Mean   :    150928  Mean   :2.286
 3rd Qu.:     91650  3rd Qu.:2.436
 Max.   :15114246   Max.   :2.805
```

```
***************************************
Symbolical definition for fnrv2.stratum1
***************************************
```

```
item@definition= cbind(exp(c0),log(c0))
```

```
***************************************
Relax command for fnrv2.stratum1
***************************************
```

```
item@relaxc= cbind(exp(c0),log(c0))
```

At the end of both examples, the following menu appears:

```
Do you want to save this new item?
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
1   yes
2   no
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
```

If you want to save this new item, choose 1, otherwise 2.

B.7. Mixed distribution item using Monte–Carlo random variates (`mxrv`)

Mixed distributions may be appropriate to model heterogeneous data. Special examples include the zero-inflation models, which can be used to model left-censored measurement data. The weights of the distribution components are obtained from an existing stratification item triplet, whereby `stid` and `stdi` contain definitions of the strata and their relative weights, respectively, and `strv` contains random numbers (length equal to the number of iterations) for allocation into strata. Two typical cases occur in practice.

Case 1: Mixture components are from the same distribution family. A stratified `mcrv`-item is already specified and has a number of columns equal to the number of distribution components.

Example: Let's create a new `mxrv`-item. Call it `mxrv1.stratum1` and choose the correct item type. It has to be stratified, so choose **stratum1**. Furthermore, enter 0 and 9 for minimum and maximum value.

> **Note:** As well as an `mcrv`-item, an `mxrv`-item needs a minimum and maximum value. !

Then you can decide if you want to create the new random by sampling from one stratified mcrv items or from two non-stratified items with the same distribution:

```
Please choose the way how you wish to define the mxrv item
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
1       by sampling from stratified mcrv item
2       by sampling from several non-stratified items
3       Exit dialog
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
1
```

In this case, we prefer the first option. After you have taken it, a short description of your chosen stratification (**stratum1** in this case) will be displayed on the console:

```
The corresponding strata are:
   stid stdi
```

```
1 North 1500
2 South  850
```

Then you have a choice of all of your stratified `mcrv`-items, for example:

```
Please choose stratified mcrv item to define the mxrv item
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
1   mcrv4.stratum1
3   Exit dialog
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
1
```

Take `mcrv4.stratum1` from the `mcrv`-subsection, which will be offered, and the first six rows as well as the summary statistics of your new item, without and then with regard to your minimum and maximum value, will be displayed on the console:

```
---------------------------------------------------------------------
Evaluating item values with predefined distribution parameters...
---------------------------------------------------------------------


***************************************
Full command for mxrv1.stratum1
***************************************


Dimension of test evaluation: 1000 1

item@fullc= matrix(ncol=1,data=rrisksample(data=mcrv4.stratum1,
weights=stratum1$strv),byrow=TRUE)

Generated item values (full command):
  mxrv1.stratum1
1      2.66077953
2      1.04914345
3      0.38247521
4     -0.95090443
5      0.02702451
6     -1.28596113
...

Summary statistics of item values (full command):
 mxrv1.stratum1
 Min.    :-4.13034
 1st Qu.:-0.77247
 Median :-0.01598
 Mean    :-0.01285
 3rd Qu.: 0.73929
 Max.    : 4.82880


***************************************
Symbolical definition for mxrv1.stratum1
***************************************


item@definition= Mixed distribution based on: mcrv4.stratum1


***************************************
```

```
Relax command for mxrv1.stratum1
**************************************

Dimension of test evaluation: 100 1

item@relaxc= matrix(ncol=1,data=rudiscrete(n=rriskModel@settings@N,min=0,max=9),
byrow=TRUE)

Generated item values (relax command):
  mxrv1.stratum1
1               9
2               6
3               7
4               1
5               6
6               7
...

Summary statistics of item values (relax command):
 mxrv1.stratum1
 Min.   :0.00
 1st Qu.:1.00
 Median :4.50
 Mean   :4.22
 3rd Qu.:7.00
 Max.   :9.00
```

Case 2: Mixture components are from different distribution families. A number of
different (non-stratified) `mcrv`-items is already specified, one item for each distribution component.

Example: Let's create a new `mxrv`-item. Call it `mxrv2.stratum1` and choose the correct
item type. It has to be stratified, so choose `stratum1`. Furthermore, enter 0 and 9 for
minimum and maximum value (see the example in the first case). As you want to use
random variates from different distribution families, you have to take the second option in
the next menu:

```
Please choose the way how you wish to define the mxrv item
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
1       by sampling from stratified mcrv item
2       by sampling from several non-stratified items
3       Exit dialog
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
2
```

As in the first example, a short summary of your stratification `stratum1` will appear on the
console:

```
The corresponding strata are:
   stid stdi
1 North 1500
2 South  850
```

Then all non-stratified mcrv items of your model are listed. In this case, take `mcrv1` (Normal
distributed) and `mcrv3` (discrete distributed) (two items because of the features of `stratum1`)

by entering their corresponding numbers in R-syntax `c( , )`. After this, the first six rows as well as the summary statistics of your new item, without and then with regard to your minimum and maximum value, will be displayed on the console:

```
----------------------------------------------------------------------
Evaluating item values with predefined distribution parameters...
----------------------------------------------------------------------


***************************************
Full command for mxrv2.stratum1
***************************************


Dimension of test evaluation: 1000 1

item@fullc= matrix(ncol=1,data=rrisksample(data=data.frame(North=mcrv1,
South=mcrv3),weights=stratum1$strv),byrow=TRUE)


Generated item values (full command):
  mxrv2.stratum1
1       50.56422
2       88.47828
3        2.00000
4        4.00000
5       57.71497
6       80.37012
...


Summary statistics of item values (full command):
 mxrv2.stratum1
 Min.   :-42.56
 1st Qu.:  4.00
 Median : 31.64
 Mean   : 33.35
 3rd Qu.: 57.03
 Max.   :127.94



***************************************
Symbolical definition for mxrv2.stratum1
***************************************


item@definition= Mixed distribution based on: North=mcrv1,South=mcrv3


***************************************
Relax command for mxrv2.stratum1
***************************************


Dimension of test evaluation: 100 1

item@relaxc= matrix(ncol=1,data=rudiscrete(n=rriskModel@settings@N,min=0,max=9),
byrow=TRUE)


Generated item values (relax command):
  mxrv2.stratum1
1               7
2               3
3               5
```

```
4            5
5            9
6            4
...

Summary statistics of item values (relax command):
 mxrv2.stratum1
 Min.   :0.00
 1st Qu.:2.00
 Median :4.50
 Mean   :4.52
 3rd Qu.:7.00
 Max.   :9.00
```

At the end of both examples, the following menu appears:

```
Do you want to save this new item?
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
1   yes
2   no
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
```

If you want to save this new item, choose 1, otherwise 2.

B.8. Resampling item (`rsrv`)

A re-sampling item is generated by sampling with replacement from a `data`-item or an `mcrv`-item. A resampling item is a vector of length equal to the number of iterations. Each value in the specified source data has the same sampling weight.

Example: Let's create a new resamplng item. It is to be called `myrsrv`:

Click on OK, after. Now you have to select a data set, from which you sample some of its components with replacement casually:

```
Please choose data set for resampling
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
1    sdata$dishsize
2    zipdata$zipdata
3    zipdata$zipdata
4    c0
5    t
6    ct
7    i
8    s
9    n
10    r
11    P
12    otherOF
13    Exit dialog
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
```

In this example, choose the first option. Then the first six random values and the summary statistics of the new item are displayed in the console:

```
sdata$dishsize :   35 27 31 65 64 21 ...


**************************************
```

```
Full command for myrsrv
**************************************

Dimension of test evaluation:  100 1

item@fullc= sample(x=sdata$dishsize,size=rriskModel@settings@N,replace=TRUE)

Generated item values (full command):
  myrsrv
1     64
2     21
3     35
4    110
5     32
6     29
...

Summary statistics of item values (full command):
     myrsrv
 Min.   : 12.00
 1st Qu.: 33.50
 Median : 48.00
 Mean   : 51.96
 3rd Qu.: 63.00
 Max.   :130.00


**************************************
Symbolical definition for myrsrv
**************************************

item@definition= resample(sdata$dishsize)

**************************************
Relax command for myrsrv
**************************************

item@relaxc= sample(x=sdata$dishsize,size=rriskModel@settings@N,replace=TRUE)
```

At the end, the following menu appears:

```
Do you want to save this new item?
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
1   yes
2   no
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
```

If you want to save this new item, choose 1, otherwise 2.

B.9. Bootstrap random variate item (`bsrv`)

Bootstrap methods can be applied to estimate the uncertainty distribution for the mean and standard deviation of some data set. These estimates can later be used to specify the distribution parameters (mean and standard deviation) of an `mcrv`-item, allowing separation of uncertainty and variability. `rrisk` uses parametric and nonparametric bootstrap methods

implemented using R's `fitdistrplus` package. The parametric option realises bootstrap sampling from distributions fitted to the bootstrap samples, whereas the nonparametric option uses bootstrap sampling directly from the specified data. Type `?bootdist` for more technical details. The role of `bsrv`-items is automatically assigned to "u" for uncertainty. The `bsrv`-item has always a dimension of `.N` rows and two columns, one for the mean (`mean`) value and one for the standard deviation (`sd`), respectively.

In the following example, the names and the types of the new items are defined by this pop-up window:

Example: Assume we wish to generate the `bsrv`-item named `mybsrv` as nonparametric bootstrap distribution for the parameter estimates (mean and standard deviation) of the variable `dishsize` in the data set `sdata`. (Of course, the parametric bootstrap method is a possible option, too.)

```
Choose bootstrap method
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
1   Parametric (sampling from fitted Normal distribution)
2   Nonparametric (sampling from data)
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -

2


Please choose data for bootstraping
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
1       sdata$dishsize
```

```
2       dataitem1$N
3       dataitem1$n
4       dataitem1$k
5       Exit dialog
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -

1
```

This item can later be used to parametrise a Normal distribution mcrv-item. Automatically, the columns of the new bdjp-item will be named mean and sd. On the console, the first six rows and the summary statistics of this item will be written:

```
****************************************
Full command for mybsrv
****************************************

Dimension of test evaluation:  100 2

item@fullc= bootdist(fitdist(sdata$dishsize,'norm',method='mle'),
bootmethod='nonparam',niter=rriskModel@settings@N)$estim

Generated item values (full command):
      mean         sd
1 51.27818 26.06529
2 46.96503 23.77754
3 47.65833 24.95369
4 49.53057 25.65056
5 49.27199 24.47758
6 49.83026 25.35918
...

Summary statistics of item values (full command):
      mean              sd
 Min.   :46.29   Min.   :22.46
 1st Qu.:47.73   1st Qu.:23.86
 Median :48.72   Median :24.67
 Mean   :48.81   Mean   :24.74
 3rd Qu.:49.84   3rd Qu.:25.52
 Max.   :51.47   Max.   :28.04


****************************************
Symbolical definition for mybsrv
****************************************

item@definition= mean, sd (nonparam. bootstrap); fitted to sdata$dishsize

****************************************
Relax command for mybsrv
****************************************

item@relaxc= bootdist(fitdist(sdata$dishsize,'norm',method='mle'),
bootmethod='nonparam',niter=rriskModel@settings@N)$estim
```

> **Note:** When using the columns of a bsrv-item for definition of further items, the column names "mean" and "sd" can be used to refer to the bootstrap mean and standard deviation, respectively.

!

At the end, the following menu appears:

```
Do you want to save this new item?
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
1   yes
2   no
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
```

If you want to save this new item, choose 1, otherwise 2.

> **Note:** `bsrv`-items generated in the relaxed model are identical copies of their counterparts in the full model. ⚠

B.10. Bayes domain joint posterior distribution item (`bdjp`)

The `bdjp`-item allows to model input parameters which are interrelated via some under-lying sub-model. This approach is similar to Bayesian networks and overcomes the typical limitation of unidirectional (from-parameter-to-parameter) simulation in Monte-Carlo (MC) risk models [Smid et al., 2010]. The `bdjp`-item can only be used in connection with functions in rrisk, designed to establish a joint posterior distribution using Markov chain Monte-Carlo (MCMC) technique [Gilks et al., 1996]. rrisk uses the MCMC algorithm as implemented by the BRugs package [Thomas et al., 2006]. rrisk uses functions from the package "rrisk Bayes" (see Belgorodski et al. [2012a] and Greiner et al. [2012]) in order to estimate MCMC models. In this package the following two MCMC models are implemented:

- Bayesian prevalence estimation under misclassification
- Bayesian estimation of a zero inflated Poisson (ZIP) model

The examples below demonstrates how Bayesian domains may solve a problem of standard MC models.

If you wish to create a new `bdjp`-item, you must give it a name and an item type, as in all the other types. See the screenshot above how to enter. Then, click on `OK` and the following choice will be indicated on the console:

```
Please choose MCMC estimation method
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
1   Bayesian prevalence estimation under misclassification
2   Bayes estimation of a zero inflated Poisson (ZIP) model
3   Exit dialog
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
```

For the first two options, two examples are described:

Example 1: Let's take the first submenu item. A pop up window appears.
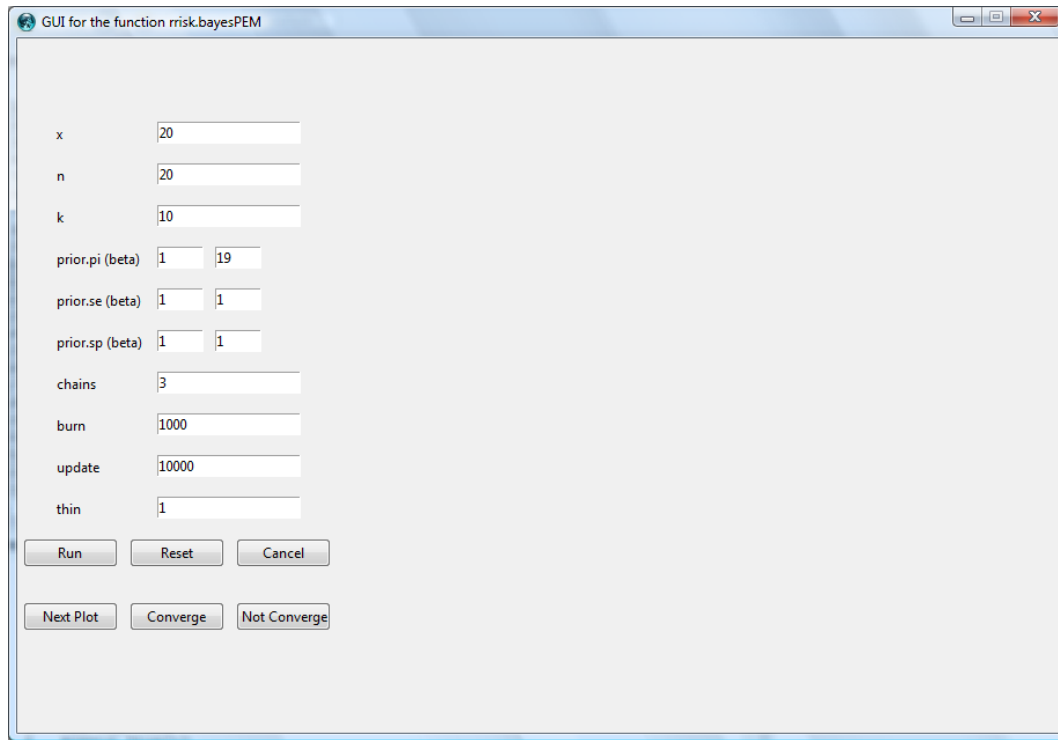
Here is a list of all fields within this window:

- **x**: scalar value for number of pools ($k > 1$) or individual outcomes ($k = 1$) with positive test result

- **n**: scalar value for number of pools tested ($k > 1$) or the sample size in application study ($k = 1$)

- **k**: scalar value for number of individual samples physically combined into one pool; set $k > 1$ for pooled sampling and $k = 1$ for individual sampling

- **prior.pi**: numeric vector containing parameters of a beta distribution as prior for prevalence pi, e.g. pi prior.pi(*,*)=beta(*,*)

- **prior.se**: numeric vector containing parameters of a beta distribution as prior for sensitivity se, e.g. se prior.se(*,*)=beta(*,*)

- **prior.sp**: numeric vector containing parameters of a beta distribution as prior for specificity sp, e.g. sp prior.sp(*,*)=beta(*,*)

- **chains**: positive single numeric value, number of independent MCMC chains (default value is 3)

- **burn**: positive single numeric value, length of the burn-in period (default value is 1000)

- **update**: positive single numeric value, length of update iterations for estimation (default value is 10000)

- **thin**: positive single numeric value (default value is 1). The samples from every kth iteration will be used for inference, where k is the value of thin. Setting `thin` > 1 can help to reduce the autocorrelation in the sample.

The button on the left bottom of this extra window have the following effects:

- **Run**: After changing the parameters, you can restart the MCMC estimation procedure by pressing this button. The diagnostic plots will appear in the right hand side of the GUI window:

77
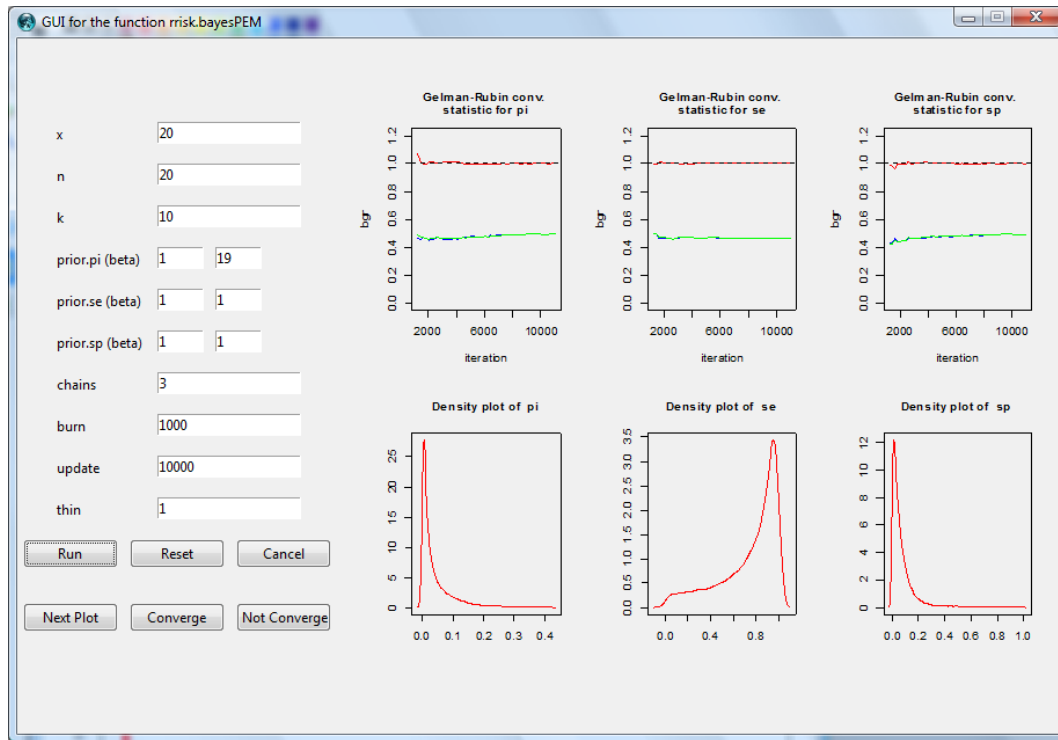
- **Reset**: Here you can replace all your entered numbers by the default ones. Click on **Run** to get default graphics.

- **Cancel**: This button lets you quit this pop-up window. Your last entries will be taken to **rrisk** for the creation of the new item.

- **Next Plot**: By clicking on this, you can switch between the available graphics.

- **Converge**: If the judgement of graphical diagnostics suggests that convergence has been reached, the user presses this button and the function returns an instance of the bayesmodelClass class containing following information:

  - **convergence**: logical, whether the model has converged (assessed by the user)

  - **results**: data frame containing statistics of the posterior distribution

  - **jointpost**: data frame giving the joint posterior probability distribution

  - **nodes**: names of the parameters jointly estimated by the Bayes model

  - **model**: model in BRugs/Winbugs syntax as a character string

  - **chains**: see above

  - **burn**: See above

  - **update**: See above

- **Not Converge**: In case of non-convergence the functions return as well an object of class **bayesmodelClass**, but the slot convergence takes a value **FALSE**.

rrisk comments on the console what it is doing after you have clicked on **Run**:

```
----------------------------------------------------------------
Begin model fitting...
model is syntactically correct
data loaded
```

```
model compiled
initial values generated, model initialized
1000 updates took 0 s
monitor set for variable 'pi'
monitor set for variable 'se'
monitor set for variable 'sp'
10000 updates took 1 s
End model fitting...
------------------------------------------------------------------
```

After you have pressed on `Converge`, the console will display the summary statistics of the new item. Furthermore, this `bdjp`-item outputs a `data`-item `bdjp1.data`, which consists of three columns, called `pi`, `se` and `sp`:

```
Simulation results has been successfully saved to the bdjp data item 'bdjp1.data'
$bdjpItem
Name:           bdjp1
Part:
Title:
Stratum:
Type:           Bayes domain joint posterior distribution item (bdjp)
Type code:      bdjp
Data:
      mean      sd  MC_error  val2.5pc  median val97.5pc start sample
pi 0.04026 0.05424 0.0008039 0.0004629 0.01791    0.2033  1001  30000
se 0.74550 0.25850 0.0029150 0.0901700 0.84630    0.9958  1001  30000
sp 0.07870 0.11260 0.0015780 0.0015370 0.04364    0.4048  1001  30000
Definition:     Bayesian prevalence estimation
Dependent items: bdjp1.data
Unit:
Role:           Not defined (nd)
Role code:      nd
Plausimin:
Plausimax:
Scores:         0 0 0 0 0 0 0 0 0 0
Explanation:
Assumptions:
Remark:
Reference:
Full command:
Relax command:


$bdjpDataitem
Name:           bdjp1.data
Part:
Title:          Posteriors of the bdjp-item 'bdjp1'
Stratum:
Type:           Data item (data)
Type code:      data
Data:
          pi         se         sp
1 0.015857792 0.7050791 0.15395674
2 0.018341560 0.8640602 0.07828263
3 0.009631266 0.7789524 0.07512484
4 0.053117897 0.9833072 0.09469400
5 0.035817262 0.9725133 0.08634455
6 0.009956134 0.9890266 0.10501574
```

```
...
        pi                  se                  sp
 Min.   :0.0000001   Min.   :0.0000589   Min.   :0.0000031
 1st Qu.:0.0058131   1st Qu.:0.6161278   1st Qu.:0.0178817
 Median :0.0179069   Median :0.8462561   Median :0.0436317
 Mean   :0.0402632   Mean   :0.7455305   Mean   :0.0786973
 3rd Qu.:0.0514172   3rd Qu.:0.9467235   3rd Qu.:0.0926195
 Max.   :0.4162369   Max.   :0.9999803   Max.   :0.9982929
Definition:
Dependent items:
Unit:
Role:           Not defined (nd)
Role code:      nd
Plausimin:
Plausimax:
Scores:         0 0 0 0 0 0 0 0 0 0
Explanation:
Assumptions:
Remark:
Reference:
Full command:
Relax command:
```

Example 2: Let's take the second submenu item. You are presented the columns of all `data`-items which are saved in your model and which contain positive integers, including zeros and of the minimal length 10; for example:

```
Please choose data set for bayesian estimation
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
1   sdata$dishsize
2   datatitem1$N
3   dataitem1$n
4   dataitem1$k
5   zipdata$zipdata
6   zipdata$zipdata
7   Exit dialog
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
```

Choose the fifth option. (The four ones above do not fit to this model, so in those cases error messages will appear.) Similar to the first example, a pop-up window will appear.
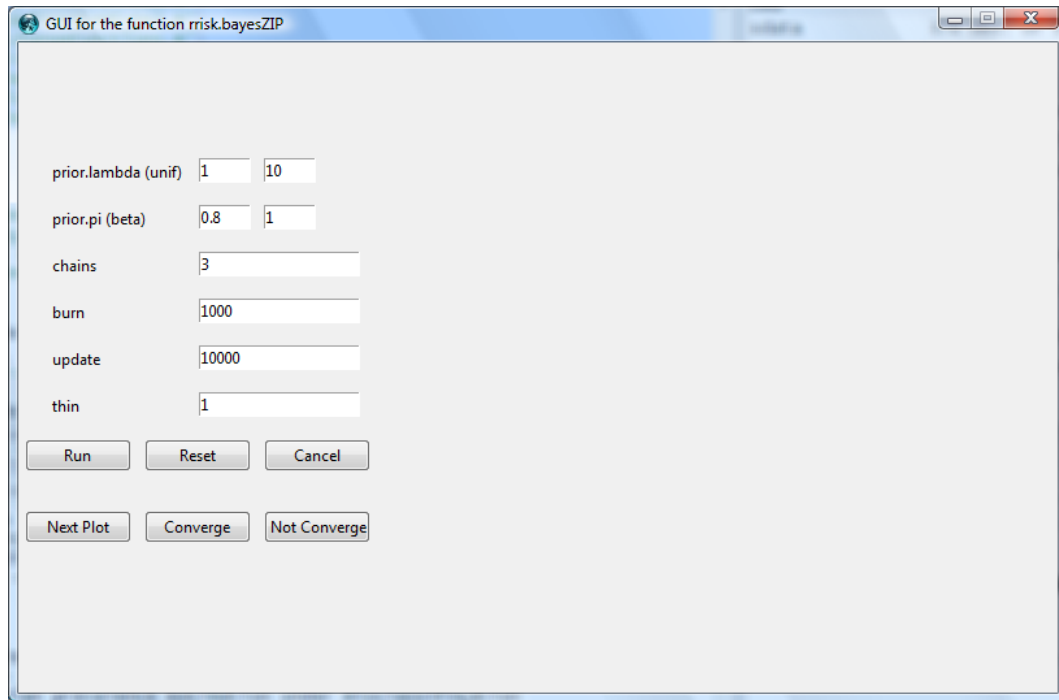
Here an accoring description of all fields is listed:

prior.lambda: numeric vector containing minimum and maximum of a uniform distribu-
tion used as prior for the Poisson parameter lambda, e.g. lambda prior.lambda(*,*)=unif(*,*)

prior.pi: numeric vector containing parameters of a beta distribution describing prior
knowledge about prevalence pi (proportion of contaminated samples), e.g. pi prior.pi(*,*)=beta(*,*)

chains: positive single numeric value, number of independent MCMC chains (default value
is 3)

burn: positive single numeric value, length of the burn-in period (default value is 1000)

update: positive single numeric value, length of update iterations for estimation (default
value is 10000)

thin: positive single numeric value (default value is 1). The samples from every kth
iteration will be used for inference, where k is the value of thin. Setting thin > 1 can
help to reduce the autocorrelation in the sample.

The buttons mean exactly the same as in the first example. For example, click on Run and
the plots of the same kind as in the first example will occur:
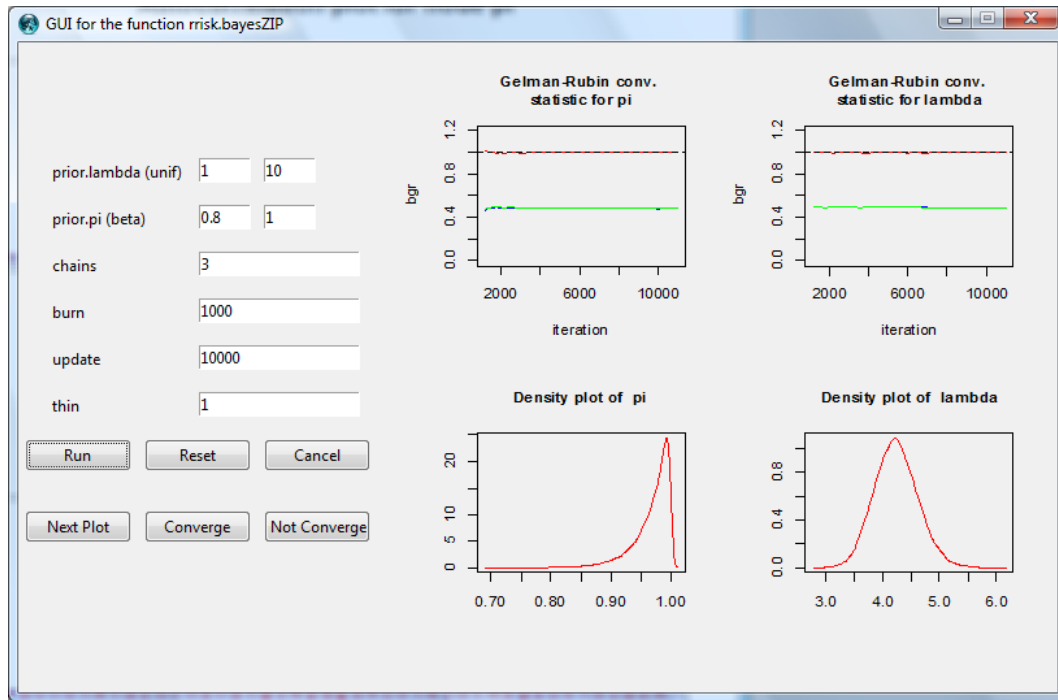
As well as in the first example, you can follow what `rrisk` is doing after you have clicked on `Run`:

```
------------------------------------------------------------------
Begin model fitting...
model is syntactically correct
data loaded
model compiled
initial values generated, model initialized
1000 updates took 0 s
monitor set for variable 'pi'
monitor set for variable 'lambda'
10000 updates took 6 s
End model fitting...
------------------------------------------------------------------
```

If you have pressed on `Converge`, you will be shown the summary statistics of the new item. According to the first example, this `bdjp`-item outputs a `data`-item `bdjp2.data`, but this has two columns, called `pi` and `lambda`. The first six rows and the summary statistics of this `data`-item will be indicated on the console, too:

```
------------------------------------------------------------------
Simulation results has been successfully saved to the bdjp data item 'bdjp2.data'
$bdjpItem
Name:            bdjp2
Part:
Title:
Stratum:
Type:            Bayes domain joint posterior distribution item (bdjp)
Type code:       bdjp
Data:
        mean      sd  MC_error val2.5pc median val97.5pc start sample
pi    0.9687 0.03041 0.0001815   0.8883 0.9779    0.9992  1001  30000
```

```
lambda 4.2370 0.37540 0.0021900   3.5400 4.2270    5.0080  1001  30000
Definition:       Bayesian estimation of ZIP model; fitted to zipdata$zipdata
Dependent items: bdjp2.data
Unit:
Role:             Not defined (nd)
Role code:        nd
Plausimin:
Plausimax:
Scores:           0 0 0 0 0 0 0 0 0 0
Explanation:
Assumptions:
Remark:
Reference:
Full command:
Relax command:


$bdjpDataitem
Name:             bdjp2.data
Part:
Title:            Posteriors of the bdjp-item 'bdjp2'
Stratum:
Type:             Data item (data)
Type code:        data
Data:
          pi    lambda
1 0.9853652 3.557990
2 0.9460345 3.567614
3 0.9894894 4.005904
4 0.9906653 4.072212
5 0.9270650 4.124577
6 0.9742287 4.408022
...
       pi                lambda
 Min.   :0.7009   Min.    :2.948
 1st Qu.:0.9563   1st Qu.:3.979
 Median :0.9779   Median :4.227
 Mean   :0.9687   Mean    :4.237
 3rd Qu.:0.9908   3rd Qu.:4.484
 Max.   :1.0000   Max.    :5.996
Definition:
Dependent items:
Unit:
Role:             Not defined (nd)
Role code:        nd
Plausimin:
Plausimax:
Scores:           0 0 0 0 0 0 0 0 0 0
Explanation:
Assumptions:
Remark:
Reference:
Full command:
Relax command:
```

At the end of both examples, the following menu appears:

```
Do you want to save this new item?
```

```
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
1   yes
2   no
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
```

If you want to save this new item, choose 1, otherwise 2.

C. General information about using items

This section is about getting the elements of all items which consist of output numbers in order to calculate with them. Some example items of the previous chapter are listed here:

C.1. How to access the results of the items

The following expressions in the right column can be used to enter numerical values on the console. Using those expressions in functions is possible, e.g. `mean(bdjp2.data$lambda)`.

Table 3: *All possible entries for all item types having output values to leave their results*

Item type	stratification	item name	access to evaluted item results
data	non-stratified	dataitem1	dataitem1$N, dataitem1$k, dataitem1$n
numv	non-stratified	numv1	numv1
numv	stratum1	numv4.stratum1	numv4.stratum1$North, numv4.stratum1$South
mcrv	non-stratified	mcrv1	mcrv1
mcrv	stratum1	mcrv4.stratum1	mcrv4.stratum1$North, mcrv4.stratum1$South
mxrv	stratum1	mxrv1.stratum1	mxrv1.stratum1
rsrv	non-stratified	myrsrv	myrsrv
bsrv	non-stratified	mybsrv	mybsrv$mean, mybsrv$sd
bdjp	non-stratified	bdjp2	bdjp2.data$pi, bdjp2.data$lambda

D. `rrisk` **syntax and functions**

`rrisk` uses R-syntax, which applies therefore to all entries of expressions or references to data. Introductory text about R (e.g., http://cran.r-project.org/doc/manuals/R-intro.pdf). Table 4 summarises the basic rules for legal entries for model definition. The following functions can be explicitly called by the user to generate model items or any other purpose[1].

Table 4: *Examples for legal entries during the user-interactive model definition.*

Generic entries (always possible)	
?	to display context sensitive help (this option is currently under evaluation)
_d	to display data loaded inside the `rrisk`-model (this option is currently under evaluation)
_e	to display the explanations of the current model (this option is currently under evaluation)
_m	to display the current model definitions (this option is currently under evaluation)
Text string entries (e.g., used to define names of items)	
abc	to define the name `abc` for an item or to enter free text
abc_def	to define the name `abc` for an item associated with the index `def`
abc$def	to refer to data `def` in the data object `abc`[a]
Value entries (e.g., used to define parameter values)	
123	to enter the value 123
c(1:3)	to enter the three values 1,2,3[b]
abc	to refer to the values represented by the item `abc`

[a]The data `abc` are uploaded to the model definition for documentation purposes.
[b]Any expression in R syntax is accepted as long as the expression returns the required number of values.

---

[1]The code of all functions can be displayed by typing just the function name at the R prompt.

E. LᴬTᴇX syntax

Text entries in either the model definition template (MDT) or the model definition interfaces (MDI) can be made using ASCII symbols. However, entries are also fully compatible with the LᴬTᴇX markup language. See http://www.latex-project.org/. A review of the LᴬTᴇX-syntax is beyond the scope of this guide. A very useful source for TᴇXSyntax is http://de.wikipedia.org/wiki/Hilfe:TeX
A few useful commands are given below.

E.1. Text mode

LᴬTᴇX controls to a great extend the layout of text. For example, any blank space between words is adjusted to "optimal" typesetting width. Separation between paragraphs should be two lines. Line breaks can be inserted via "\\".

Commands used in text mode	Output result
\cite{pouillot2009mp}	Pouillot et al. [2009]
\citep{pouillot2009mp}	[Pouillot et al., 2009]
\trrisk\	rrisk
\enquote{Quotation marks}	"Quotation marks"
\%	%
\&	&
\$	$
\textbackslash	\
_	_
\#	#
\underline{underlined Text}	underlined Text
\emph{emphasized Text}	*emphasized Text*
\bf{bold Text}	**bold Text**
\it{italics}	*italics*
\tt{typewriter}	typewriter

```
\begin{itemize}
    \item item 1
    \item item 2
    \item item 3
\end{itemize}
```

- item 1
- item 2
- item 3

```
\begin{enumerate}
    \item item 1
    \item item 2
    \item item 3
\end{enumerate}
```

1. item 1
2. item 2
3. item 3

E.2. Math mode

Commands in the math mode must be enclosed in dollar signs ($).

Commands used in math mode	Output result
\sim	$\sim$
\{ \}	$\{\}$
x_i, x_{ij}	x_i, x_{ij}
y^a, y^{ab}	y^a, y^{ab}
\log c	$\log c$
a/b = \frac{a}{b}	$a/b = \frac{a}{b}$
\sum_{i=1}^{n}x_i	$\sum_{i=1}^{n} x_i$
\prod_{j=1}^{m}x_j	$\prod_{j=1}^{m} x_j$
\binom n k	$\binom{n}{k}$
\left[\frac{a}{b}\right]	$\left[\frac{a}{b}\right]$
\hat{x},\tilde{x}	$\hat{x}, \tilde{x}$
\widehat{xyz},\widetilde{xyz}	$\widehat{xyz}, \widetilde{xyz}$
\overline{x},\underline{x}	$\overline{x}, \underline{x}$
\sqrt{x-y^2}	$\sqrt{x - y^2}$
\alpha, \beta, \gamma, \dots, \omega	$\alpha, \beta, \gamma, \ldots, \omega$
\leq, \geq, \neq, \approx, \cong	$\leq, \geq, \neq, \approx, \cong$
\in, \notin, \cup, \cap	$\in, \notin, \cup, \cap$
\subset, \supset, \subseteq, \supseteq	$\subset, \supset, \subseteq, \supseteq$
\leftarrow, \rightarrow, \leftrightarrow	$\leftarrow, \rightarrow, \leftrightarrow$
\Leftarrow, \Rightarrow, \Leftrightarrow	$\Leftarrow, \Rightarrow, \Leftrightarrow$
\emptyset, \infty, \div, \cdot	$\emptyset, \infty, \div, \cdot$
\int_{-\infty}^{\infty}f(x)dx	$\int_{-\infty}^{\infty} f(x)dx$
\angle, \bot, \partial, \nabla	$\angle, \bot, \partial, \nabla$

Numbered equation systems are entered using the `eqnarray` environment.

```
Example (code)
\begin{eqnarray}
a &=& b + c \nonumber \\
2a &=& 2(b+c) \label{e.abc}
\end{eqnarray}
Eq.~\ref{e.abc} is just an
example.
```

Example (result)

$$
\begin{array}{rcl}
a & = & b + c \\
2a & = & 2(a + b)
\end{array} \qquad (1)
$$

Eq. 1 is just an example.

E.3. Tables and figures in `rrisk`

Only for advanced users! The `rrisk`-report contains standard tables and figures which are generated automatically. However, if you wish to add additional tables or figures to the report, this is easily done using `rrisk`'s simplified syntax. Tables and figures can be generated anywhere in the documentation text (description of background, parts, items or results). Both tables and figures are "floating" objects, i.e. their placement in the pdf file is done automatically by the LaTeX compiler. Both objects have also in common that LaTeX generates numbers which you can refer to using the `\ref{}` command. The `rrisk`-syntax for tables is

`\xtable{label}{width}{layout}{title}{table}{footnote}` where

label: is a short name for the table used in references

width: is the width and its unit of measurement

layout: is a sequence of the letters `l` (for left adjusted), `c` (for centered) or `r` (for right adjusted) columns. One letter for each column is required. Alternatively, for each column, a paragraph width can be defined using `p{width}`.

title: is the title of the table.

table: is the body of the table, entered row by row. For each new cell, the column separator (&) must be inserted. The end of row is marked with two backslashes (\\). Between the first row (header) and the other rows, a separation line can be inserted using `\midline`.

footnote: contains the text for a footnote. Use empty curled brackets `{}` if no footnote is needed.

```
Example (code)
\xtable{example}{4cm}
{My table$^a$.}{lcr}
{A &B &C \midline
left    &centered &right \\
1       &2        &3 }
{$^a$Just an example.}
Tab.~\ref{t.example} floats
to an optimal place in the
document.
```

Example (result)

Tab. 5 floats to an optimal place in the document.

The ~ character fixes the width of white space between the dot and the **\ref** command. A similar principle applies to the definition of figures. The **rrisk**-syntax differentiates whether you want to insert a jpg or a png file.

`\xpng{label}{width}{title}` or
`\xjpg{label}{width}{title}` where

label: is a short name for the table used in references. Note that the name of the file (without the extension) must be identical and that the file must be in the same working directory.

width: is the width and its unit of measurement.

title: is the title (legend) of the figure.

```
Example (code)
\xpng{car}{3cm}{My car.}
The image (Fig.~\ref{car})
is automatically placed
in the document.
```

Example (result)

The figure (Fig. 4) is automatically placed in the document.

Table 5: *My table*[a].

A	B	C
left	centered	right
1	2	3

[a]Just an example.

Figure 4: My car.

Note: The commands for tables and figures described in this section are simplified versions developed for **rrisk**. It is of course possible to use original LATEX-syntax for tables and figures.

F. FAQ – frequently asked questions

1. **Why does installation of `rrisk` automatically opens a browser window showing the documentation of the `tree` package?** This behaviour is defined in the `tree` package installation. The browser window can be closed. The installation process of `rrisk` should not be interrupted.

2. **What level of knowledge of the R software is required to run `rrisk`?** Very moderate. Knowledge of the R software is no critical requirement because all `rrisk` functions are linked to a menu structure which can be controlled by intuitive user-interaction. However, entries of mathematical expressions when defining new functions and references to data sets have to be made in R-syntax. Complex modelling tasks will also benefit from the rich functionality provided by R and its packages.

3. **Is it possible to use other literature reference systems instead of JabRef?** Yes. The only requirement is that the system can generate a list of references in pure BibTeX format.

4. **Is it possible to draw literature references from a database not located in the working directory?** Yes. The name of the BibTeX database should be replaced by the full path and the name of the data base without the bib-extension. The format should be as in this example: "`I:/public/JabRef/litbase`". Note the slash symbol to indicate directories. A local database in the working directory is advantageous if it is required to pack together all source files including the literature database for shipment to co-workers. On the other hand, the contents of local databases will likely be overlapping and it might be more convenient to maintain only one central literature database.

5. **Is it possible to move data or other objects between different models?** Yes. See import of model items.

6. **Can data items be linked dynamically to external data sources?** No. This option is not supported as this could compromise the integrity of the data-methods-results complex. Along with each version of results and report, a fixed snapshot of the underlying data is required.

7. **How many iterations are required?** That depends. If a number of distributions are used and some of them are very skewed, the upper (risky) percentiles of the outcome function cannot be estimated unless the number of iterations is large (say 10,000 or more). The convergence plot of the outcome function should be checked.

8. **Can I change the order of items in the report?** No. The items might be dependent from each other (for example, one parameter might influence the distribution of another one directly). Therefore, their order might play an important role.

9. **Error message when connecting to the server for download required packages for R or LaTeX?** Many reasons may apply. Either the Internet connection is lost or the connection setting in the Internet Explorer is blocking the server access. Protocols of the Internet Explorer are used as default for downloading packages even when other primary browsers are installed. Consult your local IT administrator to correct the connection setting if necessary.

10. **Why is the line representing the empirical cumulative density function (ECDF) sometimes outside the shaded area in 2d simulation?** It is no bug but a deficiency. The ECDF is established using the 1st order simulation even when a two-dimensional simulation has been done to produce the shaded uncertainty envelope. The 95% credible intervals of the quantiles are established using the

2d simulation results and they do – sometimes – not contain the entire ECDF plot. You should increase the number of 1st order iterations to avoid this problem. We shall soon fix this by establishing the ECDF from the entire 2d simulation data.

11. **Error message when compiling the report: I can't write on file 'Report.pdf'**
This error occurs when the file 'Report.pdf' is already open. The report file should be closed before a new version is generated.

12. **Error message when compiling the report: I found no database files—while reading file Report.aux**
This error occurs when a BibTeX sourcefile has been defined under References but no such file is found in the working directory. The BibTeX data file should be copied into the directory.

13. **Is it possible to generate `rrisk`-model reports without the use of LaTeX?**
Yes, with some limitations. The generation of a pdf file using LaTeX is the only way to obtain automatically a readable and formatted model report. However, the generated file `report.tex` may also be imported into other text processing software such as Microsoft-Word. An Internet search using the string "`latex2word`" should yield a variety of converters for this purpose. However, no guarantee can be given as to whether these systems yield satisfying results. The `.tex` file needs to be unprotected in the model settings as explained below.

14. **Is it possible to edit the model report?**
Yes, although this is not recommend. Go to `View/edit/run model`, choose the model you wish to get the report from, go to `Edit model topics`, then `Edit model settings`. A new window appears in which `Protect LaTeX source file` has to be set on `TRUE`. Having checked it, go back to the previous submenu and choose `Create model report`. Choose (or create) a folder to save all the data and choose whether and how you want to update. After receiving a positive reply you can find the editable `.tex`-document of the model report in that folder. The fact that the file was unprotected to allow editing is printed in the report. This statement must not be altered manually.

Table 6: *Discrete distribution functions available for Monte-Carlo random variate (`mcrv`) items.*

Distribution[a]	Notation in `rrisk` Parameter space Support	Density: $f(x)$
Bernoulli[a]	`bern(p=prob)` $0 \le p \le 1$ $x = 0, 1$	$p^x(1-p)^{1-x}$
Binomial	`binom(n=size,p=prob)` $0 \le p \le 1;\ n = 1, 2, \dots$ $x = 0, 1, \dots, n$	$\binom{n}{x}p^n(1-p)^{n-x}$

Discrete[a]	`discrete(X = random`
	`variable,P=pdf)`
	$0 \le P \le 1;\ \sum P = 1$
	$X \in \mathbb{R}$
		$P = p(x)$

Geometric	`geom(p=prob)` $0 \le p \le 1$ $x = 1, 2, \dots$	$p(1-p)^x$

Hypergeometric	`hyper(m = # success states`
	`in pop.,n=(pop. size) -`
	`m,k = # of draws)`
	$m = 1, ..;\ n = 1, ..;\ k = 0, 1, ..$
	$k \le m + n;\ x = 0, \dots, k$
		$\frac{\binom{m}{x}\binom{n}{k-x}}{\binom{m+n}{k}}$

Multinomial	`multinom(n=size,`$p_1,\dots,p_k$`=prob)` $x_j = 0, 1, \dots;\ j = 1, \dots, k$ $\sum_{j=1}^{k} x_j = n;\ \sum_{j=1}^{k} p_j = 1$	$n! \prod_{j=1}^{k} \frac{p_j^{x_j}}{x_j!}$
Negative Binomial	`nbinom(s=shape,p=prob)` $s = 1, 2, \dots;\ 0 < p \le 1$ $x = 0, 1, \dots$	$\frac{\Gamma(s+x)}{\Gamma(s)x!}p^s(1-p)^x = \binom{s+x-1}{x}p^s(1-p)^x$
Poisson	`pois(`λ`=rate)` $\lambda > 0$ $x = 0, 1, \dots$	$\frac{\lambda^x}{x!}\exp(-\lambda)$
Uniform Discrete[a]	`udiscrete(a=min,b=max)` $x = a, a+1, \dots, b$ $i, j \in \mathbb{Z}$	$\frac{1}{b-a+1}$

[a] Distribution provided by `rrisk`.

Table 7: *Continuous distribution functions available for Monte-Carlo random variate (`mcrv`) items.*

Distribution[a]	Notation in `rrisk` Parameter space Support	Density: $f(x)$
Beta	`beta(v=shape1,w=shape2)` $v > 0,\ w > 0$ $0 \leq x \leq 1$	$\frac{\Gamma(v+w)}{\Gamma(v)\Gamma(w)} x^{v-1}(1-x)^{w-1}$

Beta PERT[a]	`pert(a=min,b=mode,`
	`c=max,s=shape)`
	$a, c \in \mathbb{N};\ a < b < c$
	s usually equals 4
	$x = a, a+1, \ldots, b$
		$\mu = \frac{a+c+s \cdot b}{s+2}$
		$s_1 = \frac{(\mu-a)(2b-a-c)}{(b-\mu)(c-a)};\ s_2 = s_1 \cdot \frac{c-\mu}{\mu-a}$
		If $\mu = b \Rightarrow s_1 = 1 + \frac{s}{2};\ s_2 = s_1 \cdot \frac{c-\mu}{\mu-a}$
		$\frac{\Gamma(s_1+s_2)}{\Gamma(s_1)\Gamma(s_2)} x^{s_1-1}(1-x)^{s_2-1}$

Cauchy	`cauchy(s=scale,`
	`m=location)`
	$m \in \mathbb{R};\ s \in \mathbb{R}^+$
	$x \in \mathbb{R}$
		$\frac{1}{\pi s\left(1+\left(\frac{x-m}{s}\right)^2\right)}$

χ^2	`chisq(n=df)` $n \in \mathbb{N}$ $x \geq 0$	$\frac{x^{n/2-1}}{2^{n/2}\,\Gamma\left(\frac{n}{2}\right)}\ \exp\left(-\frac{x}{2}\right)$
χ^2, non-central	`chisqnc(n=df,`λ`=ncp)` $n \in \mathbb{N};\ \lambda \in \mathbb{R}^+$ $x \geq 0$	$\exp\left(-\frac{1}{2}(x+\lambda)\right) \sum_{j=0}^{\infty} \frac{x^{n/2+j-1}\,\lambda^j}{\Gamma\left(\frac{n}{2}+j\right)\,2^{jn}\,j!}$
Exponential	`exp(`λ`=rate)` $x \geq 0$	$\lambda\ \exp(-\lambda x)$
F	`f(`m`=df1,`n`=df2)` $n \in \mathbb{N};\ m \in \mathbb{N}$ $x \geq 0$	$\frac{\Gamma\left(\frac{m+n}{2}\right)}{\Gamma\left(\frac{m}{2}\right)\Gamma\left(\frac{n}{2}\right)}\ \frac{m^{m/2}\,n^{n/2}\,x^{m/2-1}}{(mx+n)^{(m+n)/2}}$
Gamma	`gamma(a=shape,s=rate)` $a > 0;\ s > 0$ $x \geq 0$	$\frac{x^{a-1}}{s^a\,\Gamma(a)}\ \exp\left(-\frac{x}{s}\right)$

[a]Distribution provided by `rrisk`.

Tab. 7 continued.

Distribution[a]	Notation in `rrisk` Parameter space Support	Density: $f(x)$
Gompertz	`gompertz(`μ`=shape,b=scale)` $\mu, b > 0$ $x \geq 0$	$b\mu e^{bx} e^{\mu} \exp\left(-\mu e^{bx}\right)$
Logistic	`logis(m=location,s=scale)` $m \in \mathbb{R};\ s \in \mathbb{R}^+$ $x \in \mathbb{R}$	$\dfrac{\exp\left(\frac{x-m}{s}\right)}{s\left(1-\exp\left(\frac{x-m}{2}\right)\right)^2}$
Lognormal	`lnorm(`μ`=meanlog,`σ`=sdlog)` $\mu > 0;\ \sigma > 0$ $x \geq 0$	$\dfrac{1}{x\sigma\sqrt{2\pi}}\ \exp\left(-\dfrac{(\log x - \mu)^2}{2\sigma^2}\right)$
Normal	`norm(`μ`=mean,`σ`=sd)` $\mu \in \mathbb{R};\ \sigma \in \mathbb{R}^+$ $x \in \mathbb{R}$	$\dfrac{1}{\sigma\sqrt{2\pi}}\ \exp\left(-\dfrac{(x-\mu)^2}{2\sigma^2}\right)$

Truncated Normal	`tnorm(`μ`=mean,`σ`=sd,`
	`a=lower,b=upper)`
	$\mu \in \mathbb{R};\ \sigma \in \mathbb{R}^+$
	$x \in \mathbb{R}$
		$\dfrac{\frac{1}{\sigma}\phi\left(\frac{x-\mu}{\sigma}\right)}{\Phi\left(\frac{b-\mu}{\sigma}\right) - \Phi\left(\frac{a-\mu}{\sigma}\right)}$

Student t	`t(n=df)` $n \in \mathbb{N}$ $x \in \mathbb{R}$	$\dfrac{\Gamma\left(\frac{n+1}{2}\right)}{\sqrt{n\pi}\,\Gamma(n/2)}\left(1 + \dfrac{x^2}{n}\right)^{-(n+1)/2}$
Triangular[a]	`triang(a=min,b=mode,c=max)` $a \leq c \leq b$ $a \leq x \leq b$	$\begin{cases} \dfrac{2(x-a)}{(b-a)(c-a)}, & \text{for } a \leq x \leq c \\[2mm] \dfrac{2(b-x)}{(b-a)(b-c)}, & \text{for } c < x \leq b \end{cases}$
Uniform	`unif(a=min,b=max)` $a \in \mathbb{R};\ b \in \mathbb{R}$ $a \leq x \leq b$	$\dfrac{1}{b-a}$
Weibull	`weibull(a=shape,s=scale)` $a \in \mathbb{R}^+;\ s \in \mathbb{R}^+$ $x \in \mathbb{R}^+$	$\dfrac{s\,x^{s-1}}{a^s}\ \exp\left(-\left(\frac{x}{a}\right)^s\right)$

[a]Distribution provided by `rrisk`.

References

Natalia Belgorodski, Matthias Greiner, and Alexander Engelhardt. *rriskBayes: Predefined Bayes models fitted with Markov chain Monte Carlo (MCMC) (related to the 'rrisk' project)*, 2012a. URL http://www.bfr.bund.de/cd/52158. R package version 2.4.

Natalia Belgorodski, Matthias Greiner, Kristin Tolksdorf, and Katharina Schueller. *rriskDistributions: Fitting distributions to given data or known quantiles*, 2012b. URL http://www.bfr.bund.de/cd/52158. R package version 1.9.

Alexander Engelhardt, Natalia Belgorodski, Kristin Tolksdorf, Christine Mueller-Graf, and Matthias Greiner. rriskdistributions: Estimating distribution parameters based on empirical quantiles and data (submitted). *R Journal*, 2012.

W.R. Gilks, S. Richardson, and D.J. Spiegelhalter. *Markov chain Monte Carlo in practice*. Chapman and Hall, London, 1996.

Matthias Greiner, Natalia Belgorodski, Alexander Engelhardt, and Christine Mueller-Graf. rriskbayes: Predefined bayes models fitted with markov chain monte carlo (mcmc) (submitted). *R Journal*, 2012.

R. Pouillot, ML Delignette-Muller, DL Kelly, and JB Dennis. The mc2d package. Technical report, 2009. URL http://riskassessment.r-forge.r-project.org/docmcEnglish.pdf.

R Development Core Team. R: A language and environment for statistical computing. Technical report, Vienna, Austria, 2009. URL http://www.R-project.org. ISBN 3-900051-07-0.

J. H. Smid, D. Verloo, G. C. Barker, and A. H. Havelaar. Strengths and weaknesses of monte carlo simulation models and bayesian belief networks in microbial risk assessment. *Int J Food Microbiol*, 139 Suppl 1:S57–S63, May 2010. doi: 10.1016/j.ijfoodmicro.2009.12.015. URL http://dx.doi.org/10.1016/j.ijfoodmicro.2009.12.015.

A. Thomas, B. O'Hara, U. Ligges, and S. Sturtz. Making BUGS open. *R.News*, 6:12–17, 2006.

J.P. Van der Sluijs, M. Craye, S. Funtowicz, P. Kloprogge, J. Ravetz, J. Risbey, V.E. Fermi, and C. Campus. Combining quantitative and qualitative measures of uncertainty in model-based environmental assessment: the NUSAP system. *Risk Analysis*, 25(2):481–492, 2005.

WHO. Harmonization project document no. 6: Part 1: Guidance document on characterizing and communicating uncertainty in exposure assessment. Technical report, World Health Organization, 2008. URL http://www.who.int/ipcs/publications/methods/harmonization/exposure_assessment.pdf.

WHO. Risk characterization of microbiological hazards in food. Technical report, World health Organization (WHO), 2009. URL https://www.who.int/foodsafety/publications/micro/MRA17.pdf.