The mc2d package.

R. POUILLOT, M.-L. DELIGNETTE-MULLER, D.L. KELLY & J.-B. DENIS

July 17, 2023

This documentation is intended for readers with:

- A medium level of experience in R. Please refer to the Manual *An Introduction to R* available with R distribution if needed;
- Some knowledge about Monte-Carlo simulation (its basic principles and its use) and about Quantitative Risk Assessment (QRA).

This documentation will not describe all arguments of the functions. The definitive reference remains the documentation associated with the package. It is recommended to try the examples while reading the explanations.

Contents

1 Introduction

1.1 What is `mc2d`?

`mc2d` means Two-Dimensional Monte-Carlo (*Monte-Carlo ? 2 Dimensions*). This package provides:

- additional probability distributions;
- tools to construct One-Dimensional and Two-Dimensional Monte-Carlo Simulations;
- tools to analyse One-Dimensional and Two-Dimensional Monte-Carlo Simulations.

In a previous version, some tools to fit parametric distributions to data were included. Because these functions can be useful for other purposes, they have been moved to a separate package called `fitdistrplus` [3].

`mc2d` was built for QRA in the Food Safety domain but it can be used in other frameworks.

1.2 What is Two-Dimensional Monte-Carlo Simulation (briefly)?

The following text and Figure 1 are adapted from [6] and [7] where this method was used. The principal reference for Two-Dimensional Monte-Carlo simulation remains [2].

According to international recommendations, a QRA should reflect the variability in the risk and take into account the uncertainty associated with the risk estimate. The variability represents for instance temporal, geographical and/or individual heterogeneity of the risk for a given population. The uncertainty is understood as stemming from a lack of perfect knowledge about the QRA model structure and associated parameters[1].

In order to reflect the natural variability of a modelled risk, a Monte-Carlo simulation approach may be useful: the empirical distribution of the risk within the population may be obtained from the mathematical combination of distributions reflecting the variability of parameters across the population.

A two-dimensional (or second-order) Monte-Carlo simulation was proposed to superimpose the uncertainty in the risk estimates stemming from parameter uncertainty [2]. A two-dimensional Monte-Carlo simulation is a Monte-Carlo simulation where the distributions reflecting variability and the distributions representing uncertainty are sampled separately in the simulation, so that variability and uncertainty in the output may be assessed separately. It may be described as following (see Figure 1):

1. The parameters of the model should be divided into four categories: the fix parameters, the parameters whose distributions reflect variability only, hereinafter denoted as *variable parameters*, the parameters whose distributions reflect uncertainty only, denoted as *uncertain parameters* and the parameters whose distributions reflect both uncertainty and variability, denoted as *variable and uncertain parameters*. For this latter category, a hierarchical structure, using hyper-parameters, has to be specified: if a parameter is both uncertain and variable, one should be able to specify an empirical or parametric distribution representing variability. This distribution is conditional upon other parameters to which is associated the uncertainty. As an example, one should be able to specify a relationship such as

$$r \mid a, b \sim N(a, b)$$

[1] In the engineering risk community, these concepts are refered as aleatoric uncertainty for variability and epistemic uncertainty for uncertainty.

where the specified normal distribution represents variability in r conditional upon parameters a and b. Hyperdistributions, such as

$$a \sim Unif\,(l_a, u_a)$$

and

$$b \sim Unif\,(l_b, u_b)$$

represent the uncertainty in the parameters a and b with uniform distribution;

2. A set of uncertain parameters are randomly sampled from their respective distributions;

3. The model is evaluated using a classical (one-dimensional) Monte-Carlo simulation of size N_v, treating the uncertain parameters as fixed. This QRA takes into account the variability in all variable parameters, and leads to an empirical density function reflecting the variability of exposure/risk across the population, conditional upon the uncertain parameters. Various statistics (*e.g.* the mean, the standard deviation, percentiles) of the resulting empirical density function are evaluated and stored;

4. Steps 2) and 3) are performed a large number (N_u) of times , leading to N_u sets of statistics. The uncertain parameters are generated with respect to their uncertain distributions;

5. As output, the 50^{th} percentile (median) of each statistic is used as a point estimate of this statistic; the 2.5^{th} and 97.5^{th} percentiles of each statistic are used to establish a 95% credible interval (CI95) of this statistic. The median of the N_u estimated values for each of the 101 estimated percentiles allows us to display a variability cumulative distribution via a graph. This curve is surrounded by the 2.5th and 97.5th percentiles obtained from the N_u estimates of each of the 101 percentiles.

More formally, the two-dimensional Monte-Carlo simulation is a tool proposed to estimate the uncertainty of probability distributions of random variables of interest (and then some of its characteristics such as the mean or some percentiles). In its simplest version, an illustration of the basic framework could be written as a chain of three random variables:

$$p \rightarrow \pi \rightarrow Y$$

characterised by the marginal distribution of $p : [p]$ and the conditional distributions of $\pi : [\pi \mid p]$ and $Y : [Y \mid \pi]$, under the assumption that the joint distribution of these three random variables is the product:

$$[p, \pi, Y] = [p][\pi \mid p][Y \mid \pi].$$

The interpretation for each of these variables is:

Y is the variable of interest;

π is the parameter characterising the distribution of Y. π is not known precisely but the uncertainty associated to π can be characterised through a distribution $[\pi \mid p]$;

p is the parameter characterising the distribution of uncertainty of π. It is assumed that its distribution, $[p]$, is known.

A two-dimensional Monte-Carlo simulation provides a bundle of N_u distributions $[Y \mid \pi = \pi_i]$ where $\pi_i, i = \{1, \ldots, N_u\}$ are independent random draws from $[\pi \mid p]$, this later distribution characterising the uncertainty distribution of π.

`mc2d` is a set of R functions that will help to implement such two-dimensional Monte-Carlo simulations. The main point to understand is that `mc2d` uses arrays of (at least) two dimensions to derive the results: the first dimension will reflect variability, the second will reflect uncertainty. This document will not develop the method further, but will illustrate the practical application of `mc2d`, using a fictitious example.

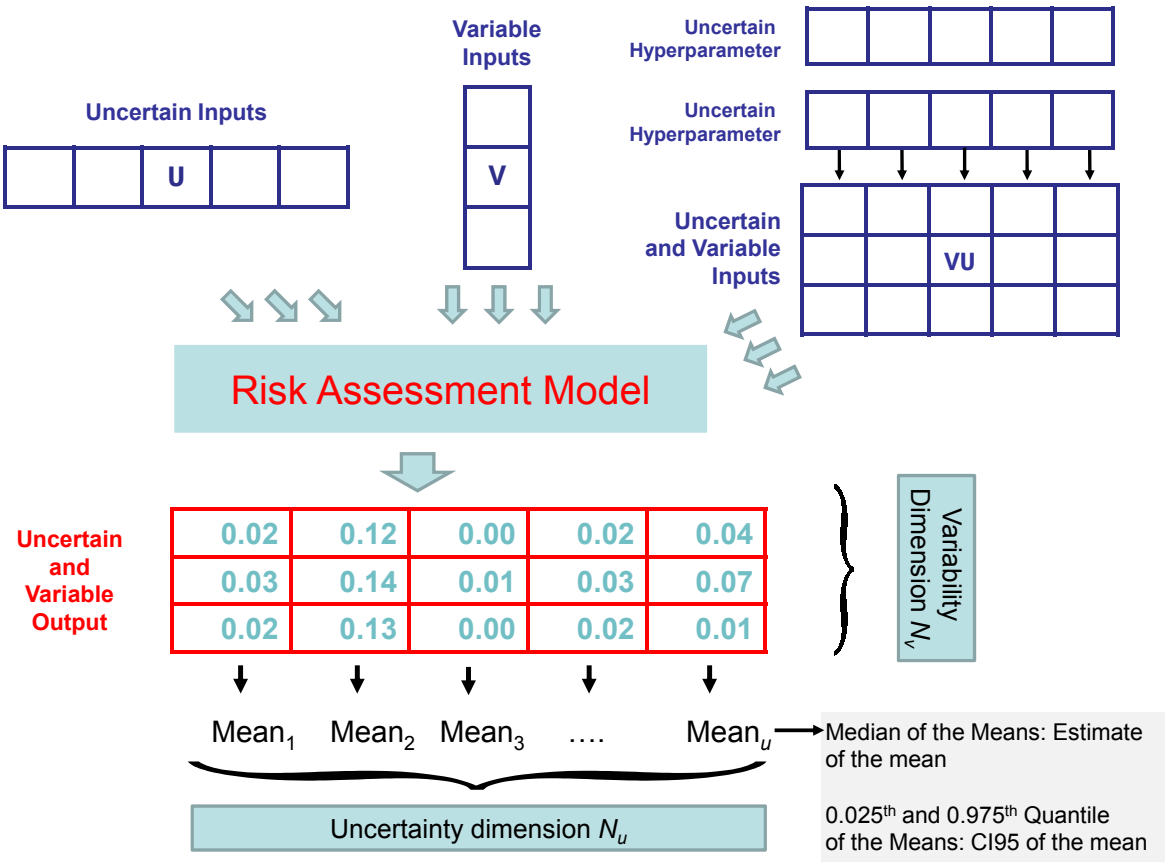


Figure 1: Schematic Representation of a Two-Dimensional Monte-Carlo Simulation.

1.3 A basic example

Quantitative Risk Assessment: *Escherichia coli* O157:H7 infection linked to the consumption of frozen ground beef in <3 year old children.

Warning: the data are fictitious and the model is over-simplified to better illustrate the use of the package: the results will not and should not be interpreted.

The model is built assuming that:

- in a given batch of ground beef, *E. coli* O157:H7 are randomly distributed with a mean concentration of $c = 10$ bacteria (cfu) per gram of product;
- no bacterial growth occurs in storage, since the product is kept frozen until it is cooked, just before consumption;
- 2.7% of consumers cook their beef rare, 37.3% medium and 60.0% well done;
- The following bacterial inactivation i is associated with these cooking practices:
  - No inactivation for rare cooking;
  - 1/5 surviving bacteria for a medium cooking;
  - 1/50 surviving bacteria for a well done cooking.
- The variability in steak serving sizes s for <3 year children can be described with a gamma distribution with parameters: $shape = 3.93$, $rate = 0.0806$.
- The dose-response relationship, describing the probability of illness, P, according to the dose is a one-hit model. The probability of illness per hit r is assumed to be constant with $r = 0.001$.

The question is: *What is the risk of illness in the population that consumed the contaminated lot?*

This distribution will be estimated using Monte-Carlo simulations performed with R via the `mc2d` package. First, the model will be developed in a one dimensional framework. Then, in order to include some uncertainties in the model, it will be derived in a two dimensional framework.

1.3.1 One Dimensional Monte-Carlo Simulation

As a first step, we assume that no uncertainty exists in our model. All distributions represent variability only. The model may be written as:

$$\begin{aligned} c &= 10. \\ i &\sim emp\left(\{1, 1/5, 1/50\}, \{0.027, 0.373, 0.600\}\right) \\ s &\sim gamma\left(3.93, 0.0806\right) \\ n &\sim Poisson\left(c \times i \times s\right) \\ r &= 0.001 \\ P &= 1 - (1 - r)^n \end{aligned}$$

where $emp\left(X, P\right)$ is an empirical distribution wherein each value X_i is associated with a probability P_i.We will use a classical one dimensional Monte-Carlo simulation, with 1,001 iterations. Using the `mc2d` package, the model may be written as:

```
> library(mc2d)
> ndvar(1001)

[1] 1001
```

```
> conc <- 10
> cook <- mcstoc(rempiricalD, values=c(1,1/5,1/50), prob=c(0.027,0.373,0.600))
> serving <- mcstoc(rgamma,shape=3.93,rate=0.0806)
> expo <- conc * cook * serving
> dose <- mcstoc(rpois,lambda=expo)
> r <- 0.001
> risk <- 1-(1-r)^dose
> EC1 <- mc(cook,serving,expo,dose,risk)
> print(EC1)
```

```
     node    mode  nsv nsu nva variate  min    mean  median     max Nas type outm
1    cook numeric 1001   1   1       1 0.02  0.1165  0.0200   1.000   0    V each
2 serving numeric 1001   1   1       1 5.17 48.3735 43.9192 219.976   0    V each
3    expo numeric 1001   1   1       1 1.03 56.0684 14.0630 987.649   0    V each
4    dose numeric 1001   1   1       1 0.00 56.1618 15.0000 962.000   0    V each
5    risk numeric 1001   1   1       1 0.00  0.0508  0.0149   0.618   0    V each
```

```
> summary(EC1)
```

```
cook :
       mean    sd  Min 2.5%  25%  50% 75% 97.5% Max  nsv Na's
NoUnc 0.117 0.176 0.02 0.02 0.02 0.02 0.2     1   1 1001    0

serving :
      mean   sd  Min 2.5%  25%  50%  75% 97.5% Max  nsv Na's
NoUnc 48.4 24.3 5.17 14.5 29.8 43.9 62.5   103 220 1001    0

expo :
      mean   sd  Min 2.5%  25%  50%  75% 97.5% Max  nsv Na's
NoUnc 56.1 96.2 1.03  3.1 8.17 14.1 79.7   259 988 1001    0

dose :
      mean   sd Min 2.5% 25% 50% 75% 97.5% Max  nsv Na's
NoUnc 56.2 96.1   0    2   7  15  79   262 962 1001    0

risk :
        mean     sd Min  2.5%     25%    50%   75% 97.5%   Max  nsv Na's
NoUnc 0.0508 0.0753   0 0.002 0.00698 0.0149 0.076 0.231 0.618 1001    0
```

The `print` output provides for each `node`: the `mode` of the variable (numeric or logical), `nsv`, the size of the variability dimension (here, 1001), `nsu`, the size of the uncertainty dimension (here, 1 since no uncertainty is considered), `nva`, the number of variate (here, all the variables are univariate), some statistics such as the `min`imum, the `mean`, the `median` and the `max`imum, evaluated after gathering the two dimensions, the number of missing data `Nas`, the `type` of the variable (`0` for fix, `V` for variable, `U` for uncertain or `VU` for variable and uncertain (here, all nodes are variable parameters only) and, finally, the level of output `outm` (used for multivariate parameters). The `summary` output provides additional statistics.

This One-Dimensional Monte-Carlo simulation provides an estimate of the mean risk (approximately 5%), as well as some quantiles of the risk distribution (2.5% of the population has a risk of illness greater than 20.3%).

1.3.2 Two dimensional Monte-Carlo Simulation

Assume now that:

- The mean concentration of bacteria in the batch is not known with certainty, but was only a point estimate. Microbiologists think that the uncertainty around this estimate can be represented via a normal distribution with parameters $\mu = 10$ and $\sigma = 2$;

- Epidemiological studies suggest that the r parameter is also uncertain. The uncertainty around the mean value of 0.001 can be represented with a uniform distribution between 0.0005 and 0.0015.

The model could then be written as:

$$\begin{aligned} c &\sim N(10, 2) \\ i &\sim emp(\{1, 1/5, 1/50\}, \{0.027, 0.373, 0.600\}) \\ s &\sim gamma(3.93, 0.0806) \\ n &\sim Poisson(c \times i \times s) \\ r &\sim Unif(0.0005, 0.0015) \\ P &= 1 - (1 - r)^n \end{aligned}$$

Note that the distributions of r and c represent uncertainty, while the distributions of i and s represent variability. n, which is a function of c, i and s, will become both variable *and* uncertain.

We will use a two-dimensional Monte-Carlo simulation, with 1,001 iterations in the variability dimension and 101 iterations in the uncertainty dimension. Using the `mc2d` package, the model may be written as:

```
> ndunc(101)

[1] 101

> conc <- mcstoc(rnorm,type="U",mean=10,sd=2)
> cook <- mcstoc(rempiricalD, type="V",values=c(1,1/5,1/50), prob=c(0.027,0.373,0.600))
> serving <- mcstoc(rgamma,type="V",shape=3.93,rate=0.0806)
> expo <- conc * cook * serving
> dose <- mcstoc(rpois,type="VU",lambda=expo)
> r <- mcstoc(runif,type="U",min=0.0005,max=0.0015)
> risk <- 1-(1-r)^dose
> EC2 <- mc(conc,cook,serving,expo,dose,r,risk)
> print(EC2,digits=2)

     node    mode  nsv nsu nva variate     min    mean median     max Nas type outm
1    conc numeric    1 101   1       1 5.91980 1.0e+01 10.045 1.6e+01   0    U each
2    cook numeric 1001   1   1       1 0.02000 1.1e-01  0.020 1.0e+00   0    V each
3 serving numeric 1001   1   1       1 2.66586 5.0e+01 44.942 1.6e+02   0    V each
4    expo numeric 1001 101   1       1 0.75130 5.4e+01 13.910 1.6e+03   0   VU each
5    dose numeric 1001 101   1       1 0.00000 5.4e+01 14.000 1.5e+03   0   VU each
6       r numeric    1 101   1       1 0.00052 9.8e-04  0.001 1.5e-03   0    U each
7    risk numeric 1001 101   1       1 0.00000 4.7e-02  0.014 8.4e-01   0   VU each

> summary(EC2)

conc :
       NoVar
median 10.05
mean   10.10
2.5%    6.03
97.5%  14.20

cook :
       mean    sd  Min 2.5%  25%  50% 75% 97.5% Max  nsv Na's
NoUnc 0.107 0.166 0.02 0.02 0.02 0.02 0.2   0.2   1 1001    0

serving :
```

```
      mean   sd  Min 2.5% 25%  50%  75% 97.5% Max  nsv Na's
NoUnc 49.6 24.9 2.67 13.6  31 44.9 63.9   110 161 1001    0

expo :
       mean    sd   Min 2.5%   25%   50%   75% 97.5%  Max  nsv Na's
median 53.7  95.6 1.275 3.36  8.10 14.14  75.8   243  960 1001    0
mean   54.0  96.2 1.282 3.38  8.15 14.22  76.2   244  966 1001    0
2.5%   32.2  57.4 0.765 2.02  4.86  8.48  45.5   146  576 1001    0
97.5%  75.9 135.2 1.802 4.75 11.45 19.98 107.1   343 1356 1001    0

dose :
       mean    sd    Min 2.5%   25%  50%   75% 97.5%  Max  nsv Na's
median 53.5  96.5 0.0000 2.00  8.00 14.0  76.0   245  946 1001    0
mean   54.1  96.5 0.0495 2.08  7.63 14.4  74.7   252  964 1001    0
2.5%   32.3  57.6 0.0000 1.00  4.50  9.0  45.0   152  592 1001    0
97.5%  76.4 136.3 1.0000 3.50 11.50 20.5 105.5   364 1407 1001    0

r :
          NoVar
median 0.001006
mean   0.000976
2.5%   0.000536
97.5%  0.001480

risk :
         mean     sd       Min     2.5%     25%     50%    75% 97.5%   Max  nsv Na's
median 0.0457 0.0714 0.00e+00 0.001925 0.00714 0.01340 0.0658 0.210 0.583 1001    0
mean   0.0473 0.0728 5.41e-05 0.002032 0.00744 0.01399 0.0702 0.216 0.588 1001    0
2.5%   0.0238 0.0397 0.00e+00 0.000653 0.00345 0.00657 0.0336 0.110 0.360 1001    0
97.5%  0.0794 0.1133 9.40e-04 0.003996 0.01372 0.02535 0.1234 0.359 0.807 1001    0
```

Note that the syntax is similar to the earlier model. However, a `type` argument is provided for each distribution, indicating whether the parameter distribution represents variability (`type=V`, by default), uncertainty (`type=U`), or both (`type=VU`).

The summary provides estimates of the variability distributions (in rows) but with a measure of their uncertainty, linked to the uncertainty around `conc` and `r`. The estimate of the mean risk is now uncertain. The median of the 101 simulations leads to a best estimate of 0.0457, with a 95% credible interval of [0.0238, 0.0794].

2 Basic Principles and Functions

A typical session of R using `mc2d` is as follows:

- From data, expert knowledge, *etc.* an empirical or parametric distribution is chosen for each parent parameter. The `fitdistrplus` [3] package is a convenient tool for assessing a parametric distribution from data;
- For each parameter, an `mcnode` object is constructed (key functions: `mcdata, mcstoc`);
- Various `mcnode` objects are grouped into an `mc` object (key function: `mc`).
- The `mc` object is studied through summaries, graphs, and sensitivity analysis (key functions: `summary.mc`, `plot.mc`, `tornado`, `tornadounc`).

2.1 Preliminary Step

The `mc2d` library should be loaded at the beginning of your R session (`library(mc2d)`).

The default size of the Monte-Carlo Simulation should be defined using the `ndvar()` function (dimension of variability) and the `ndunc()` function (dimension of uncertainty).

2.2 The `mcnode` Object as an Elementary Object.

2.2.1 `mcnode` Object Structure

An `mcnode` object is the basic element of an `mc` object. An `mcnode` is associated to one variable while an `mc` is a set of associated variables.

An `mcnode` is an array of dimension ($nsv \times nsu \times nvariates$) where nsv is the dimension of variability, nsu is the dimension of uncertainty and $nvariates$ is the number of variates of the `mcnode`[2]. Four types of `mcnode` exist:

- `V mcnode`, for *Variability*, is an array of dimension ($nsv \times 1 \times nvariates$). The distribution represents variability in the parameter;
- `U mcnode`, for *Uncertainty*, is an array of dimension ($1 \times nsu \times nvariates$). The distribution represents uncertainty in the parameter.
- `VU mcnode`, for *Variability and Uncertainty*, is an array of dimension ($nsv \times nsu \times nvariates$). The distribution represents both variability (in the first dimension) and uncertainty (in the second dimension) in the parameter.
- Additionally, a `0 mcnode` is also defined. `0` stands for Neither Variability or Uncertainty. Such nodes are arrays of dimension ($1 \times 1 \times nvariates$). No uncertainty or variability is considered for these nodes. A `0 mcnode` is not necessary in the univariate context (use a scalar instead) but is useful in constructing multivariate nodes (See section 3).

There are several ways to construct an `mcnode` object:

1. The `mcstoc` function constructs an `mcnode` from random number generating functions;
2. The `mcdata` function constructs an `mcnode` from data sets;
3. `An mcnode` can be constructed directly from operations on `mcnode` objects;
4. `mcprobtree` is a special function that constructs an `mcnode` from other `mcnodes` using a mixture of distribution, called "probability tree" in QRA;
5. Some functions, such as `==` or `>` , `is.na`, `is.finite` generate a new `mcnode` when applied to an existing `mcnode`.

2.2.2 The `mcstoc` function

The `mcstoc` function[3] is written as[4]:

```
mcstoc(func=runif, type=c(V, U, VU, 0), ..., nsv=ndvar(), nsu=ndunc(), nvariates=1,
outm=each, nsample='n', seed=NULL, rtrunc=FALSE, linf=-Inf, lsup=Inf, lhs=FALSE)
```

- `func` is a function providing random data or its name as a character. The table 1 provides available distributions from the `stats` and the `mc2d` libraries that can be used in `mcstoc`;
- `type` is the type of requested `mcnode`. By default, `mcstoc` constructs a `V mcnode`;

[2] In this section, we will only consider univariate `mcnodes`, that is `mcnodes` with $nvariates = 1$.

[3] from `stochastic`

[4] as is standard in R, most arguments have default values and will be infrequently modified.

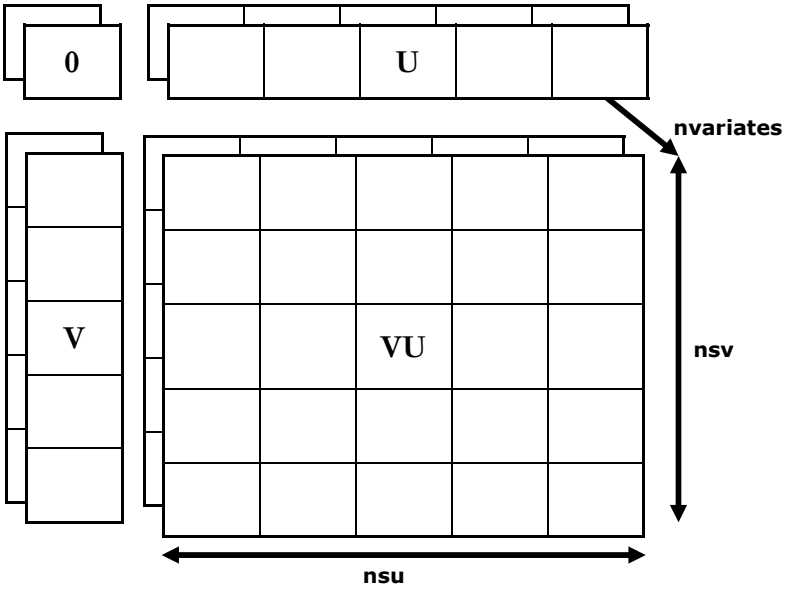


Figure 2: Structure of the various `mcnode` objects.

- ... are the arguments to be passed to the function `func`, with the exception of the argument providing the size of the sample. This latter is calculated by the function according to `func`, `type`, `nsv`, `nsu` and `nvariates`. *Note that all of the following arguments should be named*;
- `nsv` and `nsu` are the number of samples needed in the variability and uncertainty dimensions, respectively. By default, these values are the ones provided by `ndvar()` and `ndunc()`, respectively;
- `nvariates` is the desired number of variates in the `mcnode`. see section 3;
- `outm` is the default output for multivariate nodes. see section 3;
- `nsample` is the name of the argument of `func` specifying the size of the sample. It is usually `n`, with the notable exception of the `rhyper` and `rwilcox` where `nsample` should be changed in `nn`.
- `seed` optionally specifies a seed for the random number generator. If `NULL`, the seed is unchanged;
- `rtrunc` allows truncation of a distribution between `linf` and `lsup`. This option is not valid for every distribution (see table 1). See the `rtrunc` function help for further details;
- `lhs` allows Latin hypercube sampling of the node . This function is not valid for every distribution (see table 1). See the `lhs` function help for further details.

In our basic example, `mcstoc` was used to specify `conc` (a normal distribution), `cook` (an empirical discrete distribution), `serving` (a gamma distribution), and `dose` (a Poisson distribution). Note that the argument `lambda` of the Poisson distribution (node `dose`) is itself an `mcnode`.

```
> conc <- mcstoc(rnorm,type="U",mean=10,sd=2)
> cook <- mcstoc(rempiricalD, type="V",values=c(1,1/5,1/50), prob=c(0.027,0.373,0.600))
> serving <- mcstoc(rgamma,type="V",shape=3.93,rate=0.0806)
> ...
```

Table 1: Available distributions

Package	Distribution	function	Parameter `n`	Other Parameters	trunc	lhs
`stats`	beta	`rbeta`	`n`	`shape1, shape2, ncp`	Y	Y
	binomial	`rbinom`	`n`	`size, prob`	Y	Y
	Cauchy	`rcauchy`	`n`	`location, scale`	Y	Y
	chi-squared	`rchisq`	`n`	`df, ncp`	Y	Y
	exponential	`rexp`	`n`	`rate`	Y	Y
	F	`rf`	`n`	`df1, df2, ncp`	Y	Y
	gamma	`rgamma`	`n`	`shape, rate (or scale)`	Y	Y
	geometric	`rgeom`	`n`	`prob`	Y	Y
	hypergeometric	`rhyper`	`nn`	`m, n, k`	Y	Y
	lognormal	`rlnorm`	`n`	`meanlog, sdlog`	Y	Y
	logistic	`rlogis`	`n`	`location, scale`	Y	Y
	negative binomial	`rnbinom`	`n`	`size, prob (or mu)`	Y	Y
	normal	`rnorm`	`n`	`mean, sd`	Y	Y
	Poisson	`rpois`	`n`	`lambda`	Y	Y
	Student's t	`rt`	`n`	`df, ncp`	Y	Y
	uniform	`runif`	`n`	`min, max`	Y	Y
	Weibull	`rweibull`	`n`	`shape, scale`	Y	Y
	Wilcoxon	`rwilcox`	`nn`	`m,n`	Y	Y
`mc2d`	Bernoulli	`rbern`	`n`	`prob`	Y	Y
	empirical discrete	`rempiricalD`	`n`	`values, prob`	Y	Y
	empirical continuous	`rempiricalC`	n	`min, max, values, prob`	Y	Y
	PERT	`rpert`	`n`	`min, mode, max, shape`	Y	Y
	triangular	`rtriang`	`n`	`min, mode, max`	Y	Y
	generalised beta	`rbetagen`	`n`	`shape1,shape2,min,max,ncp`	Y	Y
	multinomial	`rmultinomial`	`n`	`n, size, prob`	N	N
	Dirichlet	`rdirichlet`	`n`	`alpha`	N	N
	multivariate normal	`rmultinormal`	`n`	`mean, sigma`	N	N
	beta subjective	`rbetasubj`	`n`	`min, mode, mean, max`	Y	U
	minimum information	`rmqi`	`n`	`mqi, mqi.quantile, ...`	Y	U

```
> dose <- mcstoc(rpois,type="VU",lambda=expo)
> r <- mcstoc(runif,type="U",min=0.0005,max=0.0015)
> ...
```

A normal distribution with parameters $mean = 2$, $sd = 3$, truncated on the interval [1.5, 2], with samples generated via Latin hypercube sampling could be written[5]:

```
> x <- mcstoc(rnorm, mean=2, sd=3, rtrunc=TRUE, linf=1.5, lsup=2, lhs=TRUE)
> summary(x)

node :
      mean    sd Min 2.5%  25%  50%  75% 97.5% Max  nsv Na's
NoUnc 1.75 0.144 1.5 1.51 1.63 1.75 1.88  1.99   2 1001    0
```

For convenience in using `mcstoc`, the following additional distributions have been implemented: the Bernoulli distribution (`rbern`), the empirical discrete distribution (`rempiricalD`), the PERT distribution (`rpert`)[8], the triangular distribution (`rtriang`), the Dirichlet distribution (`rdirichlet`) and the multivariate normal distribution (`rmultinormal`). The multinomial distribution has been adapted (vectorized) and `rmultinomial` (library `mc2d`) should be used in place of `rmultinom` (library `stats`). The empirical discrete (*e.g.* for bootstrap), the Dirichlet, the multinomial and the multivariate normal may be used with uncertain and/or variable parameters by specifying multivariate nodes. See section 3.

2.2.3 The `mcdata` function

Another way to construct a `mcnode` object is *via* the `mcdata` function, when data are available.

```
mcdata(data, type=c(V, U, VU, 0), nsv=ndvar(), nsu=ndunc(), nvariates=1, outm=each)
```

See the documentation associated with this function to see the size/type of data that can be used to construct an `mcnode`. The following example places a `TRUE` value in a `U` node in half of the simulations:

```
> nu <- ndunc()
> tmp <- (1:nu) > (nu/2)
> mcdata(tmp,type="U")

  node    mode nsv nsu nva variate min  mean median max Nas type outm
1    x logical   1 101   1       1   0 0.505      1   1   0    U each
```

2.2.4 Operations on an `mcnode`

`mcnode`s can be automatically constructed using operations on other `mcnode`s. Rules are used to transfer uncertainty and variability coherently within the model. Logically, the rules are as follows (illustrated here with a `+`, see table 2)[6]:

- `0` + `0` = `0`;
- `0` + `V` = `V`
- `0` + `U` = `U`;
- `0` + `VU`= `VU`;
- `V` + `V` = `V`;

[5] Note that the mean and the standard deviation of the non-truncated normal distribution are not preserved in the truncated distribution.

[6] These rules are not the standard R rules for recycling.

Table 2: mcnodes combinations

	0	V	U	VU
0	0	V	U	VU
V	V	V	VU	VU
U	U	VU	U	VU
VU	VU	VU	VU	VU

- V + U = VU [7];
- V + VU = VU [8];
- U + U = U;
- U + VU = VU [9];
- VU + VU = VU

Thus, in our example:

```
> ...
> expo <- conc * cook * serving
> ...
> risk <- 1-(1-r)^dose
```

expo is a function of a U and two V mcnodes: it is a VU mcnode with variability in the row dimension and uncertainty in the column dimension. risk is a function of a U and a VU mcnode: it is therefore a VU mcnode.

2.2.5 The mcprobtree function

The mcprobtree function can be used if a mixture of distribution is needed to construct an mcnode. Assume that the distribution representing the uncertainty on conc was not itself certain, and that the microbiologists suggest that they are 75% sure that $conc \sim N(10, 2)$ but that they are 25% sure that $conc \sim U(8, 12)$. This could be written using mcprobtree as[10]:

```
> conc1 <- mcstoc(rnorm,type="U",mean=10,sd=2)
> conc2 <- mcstoc(runif,type="U",min=8,max=12)
> whichdist <- c(0.75,0.25)
> concbis <- mcprobtree(whichdist,list("0"=conc1,"1"=conc2),type="U")
```

mcprobtree can also be used to generate samples from a mixture distribution for variability .

2.2.6 Other functions for constructing an mcnode

The functions ==, <, <=, >=, >, generate an mcnode when applied to another mcnode.

Special functions is.na(x), is.nan(x), is.finite(x), is.infinite(x) are implemented to test if any values are NA (missing data), NaN (*Not A Number*), finite or infinite .

```
> cook < 1

  node    mode  nsv nsu nva variate min  mean median max Nas type outm
1    x logical 1001   1   1       1   0 0.975      1   1   0    V each
```

[7] the U mcnode is recycled by row, the V mcnode is recycled in the standard manner by column.

[8] the V mcnode is recycled in the standard manner by column.

[9] the U mcnode is recycled by row.

[10] two alternatives for whichdist are whichdist <- mcstoc(rempiricalD, type=U, values=c(0,1), prob=c(75,25)) or whichdist <- mcstoc(rbern,type=U,prob=0.25)

```
> suppressWarnings(tmp <- log(mcstoc(runif,min=-1,max=1)))
> tmp

  node    mode  nsv nsu nva variate   min  mean median    max Nas type outm
1    x numeric 1001   1   1       1 -7.08 -1.08 -0.777 -0.001 481    V each

> is.na(tmp)

  node    mode  nsv nsu nva variate min  mean median max Nas type outm
1    x logical 1001   1   1       1   0 0.481      0   1   0    V each
```

2.2.7 Specifying a correlation between mcnodes

Structural links between sets of parameters may be very important in QRA. In mc2d, a Spearman rank correlation structure for 2 or more nodes may be specified with the cornode function. This function uses the method of Iman & Conover to generate correlated samples [4]. Assume that a study suggests that people who consume rare ground beef also consume larger serving sizes. We could specify this relation using[11]:

```
> cornode(cook,serving,target=0.5,result=TRUE)

output Rank Correlation per variates
variates: 1
[1] 1.000 0.396 0.396 1.000
$cook
  node    mode  nsv nsu nva variate  min  mean median max Nas type outm
1    x numeric 1001   1   1       1 0.02 0.107   0.02   1   0    V each

$serving
  node    mode  nsv nsu nva variate  min mean median max Nas type outm
1    x numeric 1001   1   1       1 2.67 49.6   44.9 161   0    V each
```

It is possible to create such correlations between V nodes, between U nodes, between VU nodes, or between one V node and multiple VU nodes.

The use of a multivariate normal distribution (rmultinormal) is another way to specify correlations among nodes, assuming that the individual nodes are normally distributed.

2.3 The mc Object

Once the mcnode objects are constructed, one should group them into a single object in order to analyse the Monte-Carlo results. The mc object is a list of mcnodes. There are three ways to construct an mc object: using the mc function, using the evalmcmod function, or within the evalmccut function.

2.3.1 The mc Function

```
mc(..., name=NULL, devname=FALSE)
```

... are mcnodes or mc objects to be gathered into an mc object. mc value is an mc object with specific methods, e.g. print or summary. In our example, we used:

```
> ...
> EC2 <- mc(conc,cook,serving,expo,dose,r,risk)
> print(EC2)
> summary(EC2)
```

[11] Note that the resulting correlation (around 0.4) is obviously an approximation to the desired value of 0.5, because a discrete distribution (cook: 3 categories) is correlated with a continuous distribution (serving).

2.3.2 The `mcmodel` and the `evalmcmod` Functions

A model may be written in one step using `mcmodel` (just a wrapper of your model in a function), and then evaluated using `evalmcmod`. These functions may be used once your model is correct and has been tested using a small number of iterations. For our example:

```
> modelEC3 <- mcmodel({
+   conc <- mcstoc(rnorm,type="U",mean=10,sd=2)
+   cook <- mcstoc(rempiricalD, type="V",values=c(1,1/5,1/50),
+     prob=c(0.027,0.373,0.600))
+   serving <- mcstoc(rgamma,type="V",shape=3.93,rate=0.0806)
+   r <- mcstoc(runif,type="U",min=0.0005,max=0.0015)
+   expo <- conc * cook * serving
+   dose <- mcstoc(rpois,type="VU",lambda=expo)
+   risk <- 1-(1-r)^dose
+   mc(conc,cook,serving,expo,dose,r,risk)
+ })
> modelEC3

expression({
    conc <- mcstoc(rnorm, type = "U", mean = 10, sd = 2)
    cook <- mcstoc(rempiricalD, type = "V", values = c(1, 1/5,
        1/50), prob = c(0.027, 0.373, 0.6))
    serving <- mcstoc(rgamma, type = "V", shape = 3.93, rate = 0.0806)
    r <- mcstoc(runif, type = "U", min = 5e-04, max = 0.0015)
    expo <- conc * cook * serving
    dose <- mcstoc(rpois, type = "VU", lambda = expo)
    risk <- 1 - (1 - r)^dose
    mc(conc, cook, serving, expo, dose, r, risk)
})
attr(,"class")
[1] "mcmodel"
```

Note that:

- the model is wrapped between `{` and `}`;
- any (valid) R code may be placed in the model[12];
- The model should end with an `mc()` function.

The model is then evaluated using the `evalmcmod` function:

```
evalmcmod(expr, nsv=ndvar(), nsu=ndunc(), seed=NULL)
```

One can re-run the model with different dimensions or random seeds in one line:

```
> EC3 <- evalmcmod(modelEC3,nsv=100,nsu=10,seed=666)
> EC4 <- evalmcmod(modelEC3,nsv=100,nsu=1000,seed=666)
```

2.3.3 The `mcmodelcut` and the `evalmccut` Functions

When evaluating a high-dimensional model, R may exceed its memory limit. To overcome this drawback, `evalmccut` evaluates a 2-dimensional Monte-Carlo model (written with the `mcmodelcut` function) using a loop, and calculates and stores statistics in the uncertainty dimension for further analysis. Readers should refer to the corresponding documentation for further details. Our example would be written as[13]:

[12] If needed, it is possible to make reference to the simulation dimensions using `ndvar()` and/or `ndunc()`.

[13] Note that the use of a `tornado` function in the model should be avoided as it slows the `evalmccut` function considerably.

```
> modEC4 <- mcmodelcut({
+ ## First block: unidimensional nodes
+ {cook <- mcstoc(rempiricalD, type = "V", values = c(0, 1/5, 1/50),
+                 prob = c(0.027, 0.373, 0.6))
+  serving <- mcstoc(rgamma, type = "V", shape = 3.93, rate = 0.0806)
+  conc <- mcstoc(rnorm, type = "U", mean = 10, sd = 2)
+  r <- mcstoc(runif, type = "U", min = 5e-04, max = 0.0015)
+ }
+ ## Second block: two dimensional nodes
+ {expo <- conc * cook * serving
+  dose <- mcstoc(rpois, type = "VU", lambda = expo)
+  risk <- 1 - (1 - r)^dose
+  res <- mc(conc, cook, serving, expo, dose, r, risk)     }
+ ## Third block: Outputs
+ {list(
+   sum = summary(res),
+   plot = plot(res, draw=FALSE),
+   minmax = lapply(res,range),
+   tor=tornado(res),
+   et =  sapply(res,sd))
+ }
+ })
> res <- evalmccut(modEC4, nsv = 10001, nsu = 101, seed = 666)
> summary(res)
```

2.4 Analysing an `mc` Object

As a reminder, the `print` function provides a very basic summary of the `mc` object. It has a `digits` argument (default: 3). Other more informative functions are provided in the `mc2d` package.

2.4.1 The `summary` Function

The `summary` function provides statistics on an `mc` object:

```
summary(object, probs=c(0,0.025,0.25,0.5,0.75,0.975,1), lim=c(0.025,0.975), ...)
```

The mean, the standard deviation and the quantiles provided in the `probs` arguments are evaluated on the variability dimension. Then, the median and the quantiles provided in the `lim` argument are evaluated on these statistics.

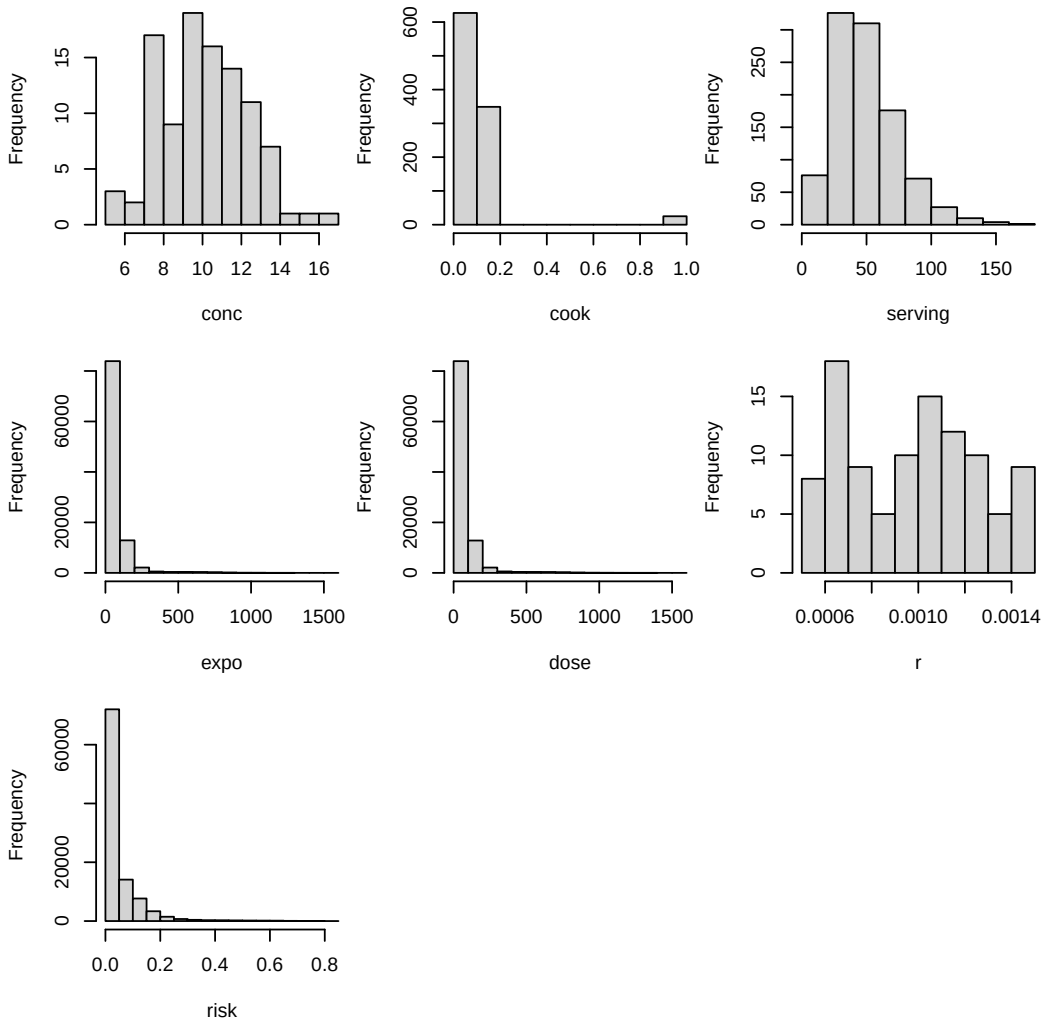
```
> tmp <- summary(EC2,probs=c(0.995,0.999),digits=12)
> tmp$risk
```

```
         mean     sd 99.5% 99.9%  nsv Na's
median 0.0457 0.0714 0.464 0.525 1001    0
mean   0.0473 0.0728 0.465 0.523 1001    0
2.5%   0.0238 0.0397 0.265 0.318 1001    0
97.5%  0.0794 0.1133 0.681 0.761 1001    0
attr(,"type")
[1] "VU"
```

2.4.2 The `hist` Function

The `hist` provides a histogram of the different `mcnodes` making up the `mc` object (cf. Figure 3).

Figure 3: Function `hist`.




```
hist (x, griddim = NULL, xlab = names(x), ylab = Frequency, main = , ...)
```

In the current version, uncertainty and variability distributions are collapsed. Thus, the resulting histogram may be meaningless.

```
> hist(EC2)
```

Note that, from `mc2d version 0.2-0`, the `gghist` function draws an histogram within the `ggplot2` framework.

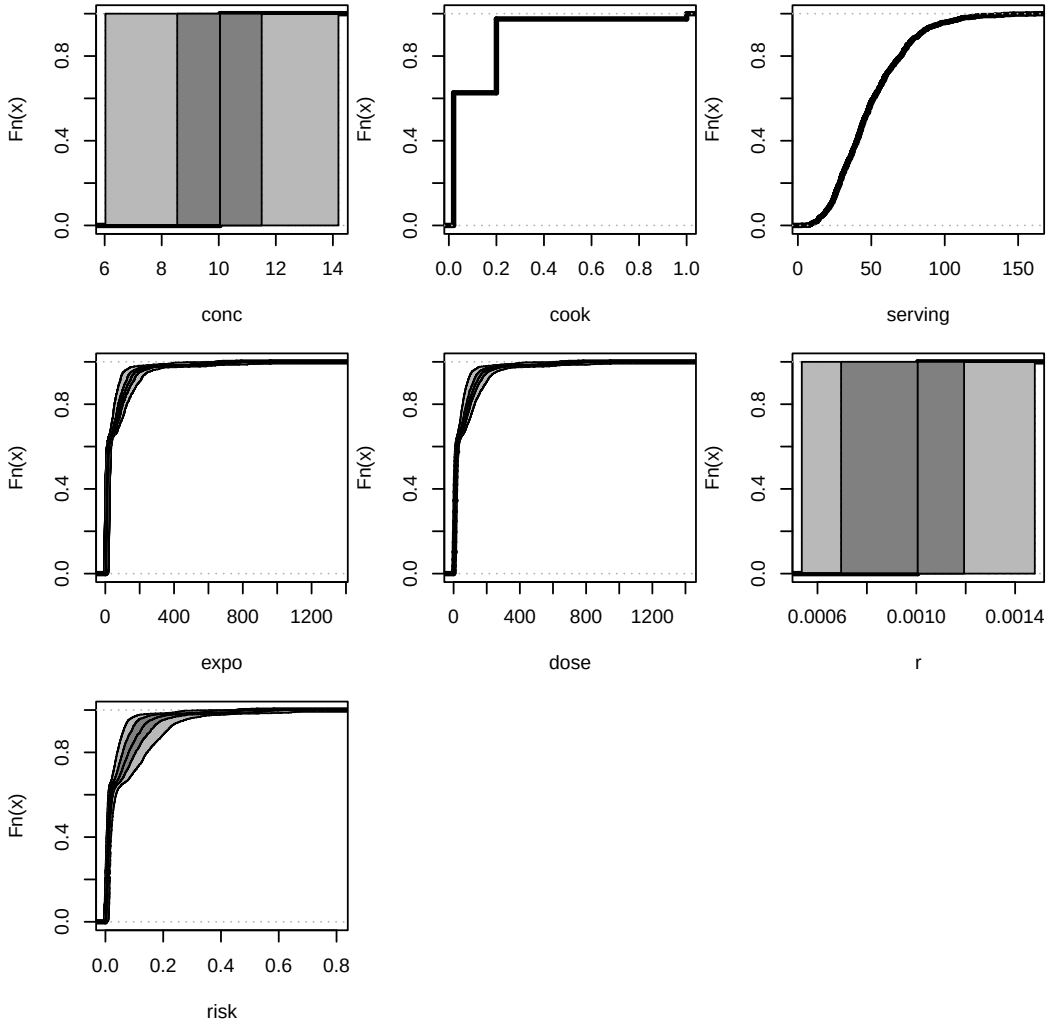
2.4.3 The `plot` function

The `plot` function provides a graph of the cumulative empirical distribution function of the estimate (mean or median) of the quantiles.

```
plot(x, prec = 0.001, stat = c("median", "mean"), lim = c(0.025, 0.25, 0.75, 0.975), na.rm =
TRUE, griddim = NULL, xlab = NULL, ylab = "Fn(x)", main = "", draw = TRUE, paint = TRUE, ...)
```

For our example, see Figure 4, a default graph. The 0.25 and 0.75 quantiles (default values of `lim`) in the uncertainty dimension of the quantiles (variability dimension) are used as the envelope.

Figure 4: `plot` Function .




```
> plot(EC2)
```

Note that, from `mc2d version 0.2-0`, the `ggplotmc` function draws an histogram within the `ggplot2` framework.

```
> ggplotmc(EC2)
```

The `spaghetti` function (and its `ggplot2` version `ggspaghetti`) allow to plot a spaghetti graph, rather than the ecdf with envelope.

Note that `mcnode` objects have the same methods `print`, `summary`, `plot`, and `hist`. Running a `ggplotmc`, a `gghist` or a `ggspaghetti` function on an `mcnode` is handy to post-process the graph.

2.4.4 The `tornado` function

The `tornado` function calculates the Spearman (default) rank correlation between nodes of the `mc` object.

```
tornado(x, output=length(x), use=all.obs, method=c(spearman, kendall,pearson), lim=c(0.025,
0.975))
```

Figure 5: `ggplotmc` Function .

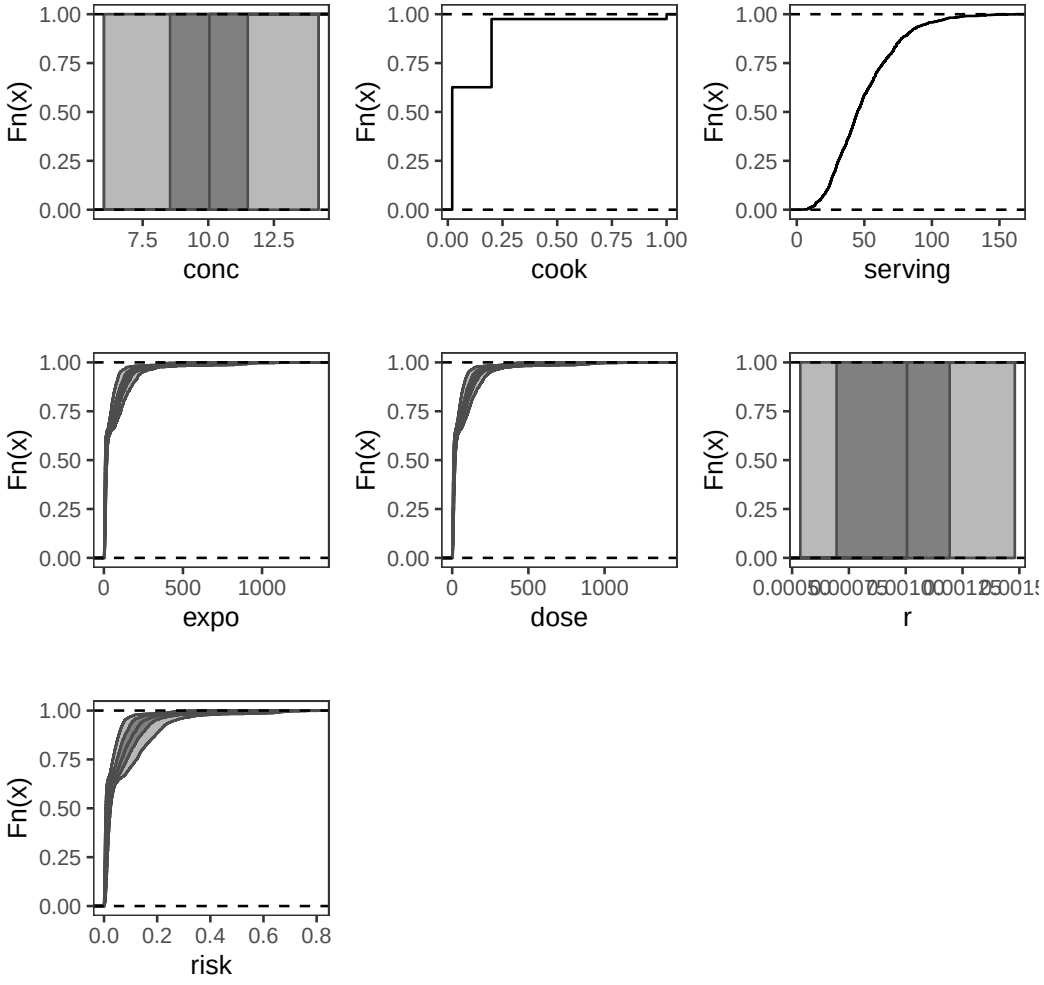
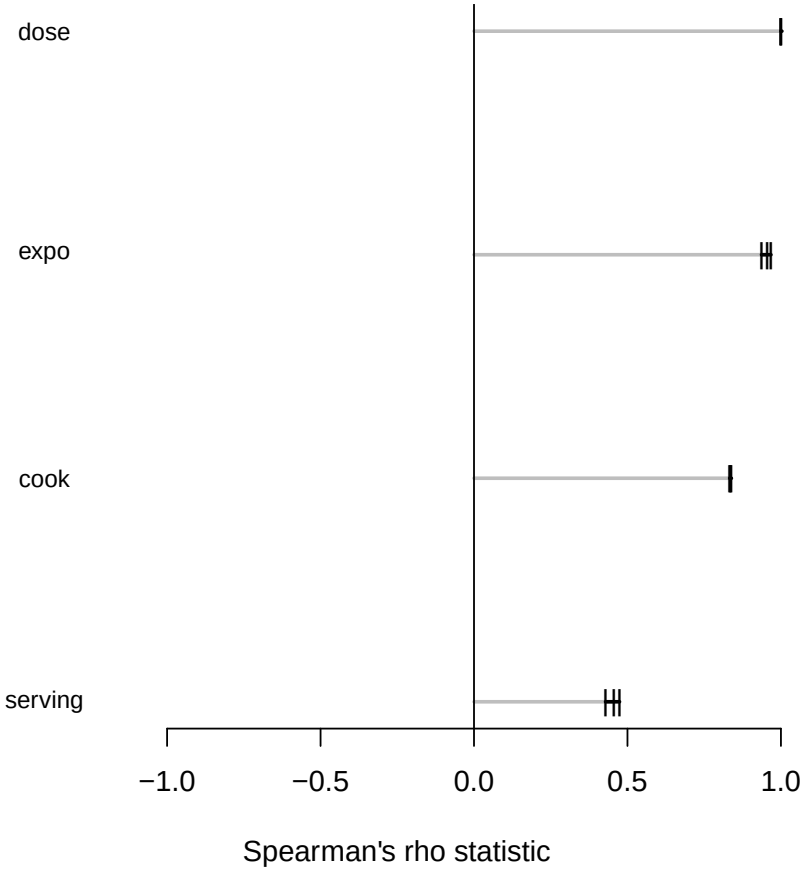


Figure 6: `plot.tornado` Function .



where `output` is the `mcnode` (name or rank) of the output (default: the last `mcnode`). Missing data are treated using the `use` arguments (see the reference documentation). `tornado` creates a `tornado` object with a `plot` method (*cf.* Figure 6).

```
> torEC2 <- tornado(EC2)
> plot(torEC2)
```

Note that, from `mc2d version 0.2-0`, the `ggplottornado` function draws an histogram within the `ggplot2` framework.

2.4.5 The `tornadounc` function

The `tornadounc` function examines the impact of the uncertainty on the estimate of an output. It calculates the Spearman (default) rank correlation between statistics of the `mc` object in the variability dimension.

```
tornadounc(mc,output = length(mc), quant=c(0.5,0.75,0.975), use = all.obs,
method=c(spearman,kendall,pearson), ...)
```

The quant argument indicates which quantiles should be used in the variability dimension. tornadounc creates a tornadounc object with a plot method

```
> tornadounc(EC2, output="risk", quant=.99)

Tornado on uncertainty
Spearman's rho statistic
Output:  risk
$risk
           conc mean expo sd expo 99% expo mean dose sd dose 99% dose     r
mean risk 0.612     0.612   0.612    0.612     0.610   0.609    0.614 0.813
sd risk   0.611     0.611   0.611    0.611     0.609   0.608    0.613 0.813
99% risk  0.611     0.611   0.611    0.611     0.609   0.608    0.616 0.811
```

The output shows the impact of the uncertain nodes (type U nodes) and some statistics (mean, median and, here, the 99^{th}percentile) calculated on the variability dimension (type VU nodes) of some output statistics .

2.4.6 The mcratio function

The mcratio function provides measures of variability, uncertainty, and both combined propose by [5] for an mc or an mcnode object. Given:

A the median of uncertainty for the median of variability;

B the median of uncertainty for the 97.5th percentile of variability;

C the 97.5th percentile of uncertainty for the median percentile of variability;

D the 97.5th percentile of uncertainty for the 97.5th percentile of variability.

The following ratio are estimated:

- Variability Ratio: B / A
- Uncertainty Ratio: C / A
- Overall Uncertainty Ratio: D / A

```
> mcratio(risk)

          A    B      C     D VariabilityR UncertaintyR Over.Unc.R
risk 0.0134 0.21 0.0254 0.359         15.6         1.89       26.8
```

2.5 Other Functions and mc Objects

mc objects are simply lists of three dimensional arrays; within each array, values in a given column represent variability in the parameter.

Knowing the structure of the mc and the structure of the mcnode objects, it is straightforward to apply any R function to these objects. The $ function is helpful for extracting an mcnode from an mc object. The unmc function removes all attributes, classes, and dimensions equal to one, providing a list of vectors, matrices and/or arrays.

Here is an example building a linear model (in fact 100 linear models) between the risk and the dose within each uncertainty dimension and estimating some statistics for the coefficients. This example is here only to illustrate that the entire spectrum of R functionality is available for your analysis.

```
> tmp <- unmc(EC2, drop=TRUE)
> dimu <- ncol(tmp$risk)
> coef <- sapply(1:dimu, function(x) lm(tmp$risk[,x] ~ tmp$dose[,x])$coef)
> apply(coef,1,summary)
```

```
        (Intercept) tmp$dose[, x]
Min.        0.00117      0.000461
1st Qu.     0.00335      0.000593
Median      0.00592      0.000773
Mean        0.00702      0.000752
3rd Qu.     0.00966      0.000861
Max.        0.02083      0.001140
```

3 Multivariate Nodes

The dimension `nvariates` is the third dimension of the `mcnode`. One can ignore it while using `mc2d` . Nevertheless, its use is mandatory to handle some multivariate distributions, and it may be useful in other circumstances. Constructing multivariate nodes is straightforward. We note that the following code:

```
> mcstoc(runif, nvariates=3, min=c(1,2,3),max=4)

  node    mode  nsv nsu nva variate  min mean median max Nas type outm
1    x numeric 1001   1   3       1 1.04 3.02   3.21   4   0    V each
2    x numeric 1001   1   3       2 1.00 3.00   3.18   4   0    V each
3    x numeric 1001   1   3       3 1.01 2.97   3.14   4   0    V each
```

will logically not provide a node with 3 variates, each having a different limit. The classical R recycling rule implies that the vector `c(1, 2, 3)` will be recycled in the first dimension, i.e. the variability dimension. Use instead:

```
> lim <- mcdata(c(1,2,3), type="0", nvariates=3)
> mcstoc(runif, nvariates=3, min=lim,max=4)

  node    mode  nsv nsu nva variate min mean median max Nas type outm
1    x numeric 1001   1   3       1   1 2.50   2.51   4   0    V each
2    x numeric 1001   1   3       2   2 2.95   2.90   4   0    V each
3    x numeric 1001   1   3       3   3 3.51   3.51   4   0    V each
```

to let `mc2d` knows that the values should be considered for a multivariate node.

3.1 Multivariate Nodes for Multivariate Distributions

The basic usage of multivariate nodes (and the reason why they have been implemented) is for multivariate distributions such as the Dirichlet distribution, the multinomial distribution, the multivariate normal distribution and, possibly, the empirical distribution

As an example, assume that 3-member families buy 500 g of ground beef. The proportions of steak eaten by the baby, his older brother and his mother follow a Dirichlet (uncertainty) distribution with (vector) parameter $\alpha = (2, 3, 5)$. We want to derive the distribution (variability) of steak masses eaten by 500 babies sampled from such families.

```
> (p <- mcstoc(rdirichlet, type="U", nvariates=3, alpha=c(2,3,5)))

  node    mode nsv nsu nva variate     min  mean median   max Nas type outm
1    x numeric   1 101   3       1 0.00389 0.184  0.173 0.486   0    U each
2    x numeric   1 101   3       2 0.04085 0.290  0.281 0.593   0    U each
3    x numeric   1 101   3       3 0.21786 0.526  0.519 0.851   0    U each

> s <- mcstoc(rmultinomial,type="VU", nvariates=3, size=500, prob=p)
> summary(s)
```

```
node :
[[1]]
       mean   sd   Min 2.5%  25%  50% 75% 97.5% Max  nsv Na's
median 92.0 51.9 0.000 13.0 53.0 86.0 121   216 259 1001    0
mean   92.0 51.8 0.228 13.4 52.6 86.1 121   216 260 1001    0
2.5%   91.5 51.2 0.000 11.5 51.0 84.0 119   212 249 1001    0
97.5%  92.5 52.4 1.500 15.0 54.0 88.0 122   219 274 1001    0

[[2]]
       mean   sd  Min 2.5%  25% 50% 75% 97.5% Max  nsv Na's
median  145 70.9 13.0 32.0 85.0 141 198   291 315 1001    0
mean    145 70.9 13.5 32.3 84.8 141 198   291 315 1001    0
2.5%    144 70.4 10.0 30.0 83.0 138 196   287 308 1001    0
97.5%   146 71.5 18.0 34.5 87.0 144 200   294 327 1001    0

[[3]]
       mean   sd  Min 2.5% 25% 50% 75% 97.5% Max  nsv Na's
median  263 74.0 93.0  117 211 261 316   401 437 1001    0
mean    263 74.1 92.4  117 211 261 316   401 437 1001    0
2.5%    262 73.4 83.0  114 209 259 314   397 429 1001    0
97.5%   264 74.8 99.0  120 214 264 318   404 444 1001    0
```

As a second example, assume that each member of these families eats a normal distribution (variability) of steak with mean 100, 150 and 250 g. There is a positive correlation between the servings of the children, and a negative one with the serving of the mother. We want to derive the distribution (variability) of steak eaten by 500 babies.

```
> sigma <- matrix(c(10,2,-5,2,10,-5,-5,-5,10), ncol=3)
> (x <- mcstoc(rmultinormal,type="V", nvariates=3, mean=c(100,150,250),
+                                sigma=as.vector(sigma)))

  node    mode  nsv nsu nva variate   min mean median max Nas type outm
1    x numeric 1001   1   3       1  89.3  100    100 110   0    V each
2    x numeric 1001   1   3       2 140.6  150    150 160   0    V each
3    x numeric 1001   1   3       3 239.9  250    250 262   0    V each

> cor(x[,1,])

       [,1]   [,2]   [,3]
[1,]  1.000  0.166 -0.475
[2,]  0.166  1.000 -0.521
[3,] -0.475 -0.521  1.000
```

In this example, `mean` could be variable or uncertain, as well as `sigma`[14]. You could have used, for an uncertain mean:

```
> m <- mcdata(c(100,150,250), type="0", nvariates=3)
> mun <- mcstoc(rnorm, type="U", nvariates=3, mean=m, sd=20)
> x <- mcstoc(rmultinormal, type="VU", nvariates=3, mean=mun, sigma=as.vector(sigma))
> cor(x[,1,])

       [,1]   [,2]   [,3]
[1,]  1.000  0.187 -0.467
[2,]  0.187  1.000 -0.543
[3,] -0.467 -0.543  1.000
```

[14] `rmultinormal` is a vectorized version of `rmvnorm` (library `mvtnorm`).

The correlation is preserved, but the mean of each category is uncertain.

As a third example, multivariate nodes may be useful to derive a nonparametric bootstrap. Assume that, based on a study, you obtained 6 individuals who eat 100 g, 12 individuals who eat 150 g, 6 individuals who eat 170 g and 6 individuals who eat 200 g of ground beef. You want to use a nonparametric bootstrap to derive uncertainty [2], and then select samples from the empirical distribution.

```
> val <- c(100,150,170,200)
> pr <-  c(6,12,6,6)
> out <- c('min','mean','max')
> (x <- mcstoc(rempiricalD, type="U", outm=out, nvariates=30,
+               values=val,prob=pr))

  node    mode nsv nsu nva variate min mean median max Nas type outm
1    x numeric   1 101  30      NA 100  100    100 100   0    U  min
2    x numeric   1 101  30      NA 137  155    155 169   0    U mean
3    x numeric   1 101  30      NA 200  200    200 200   0    U  max

> mcstoc(rempiricalD,type="VU", values=x)

  node    mode  nsv nsu nva variate min mean median max Nas type outm
1    x numeric 1001 101   1       1 100  155    150 200   0   VU each
```

Printing the statistics of the 30 variates of `x` is of no interest. Instead, we use the `outm` option, which allows us to specify which output we want (`none` for none, `each`, the default, for a series of statistics for each variate, or, as in the example, a vector of function names that are applied over all the 30 variates).

3.2 Multivariate Nodes as a Third Dimension for Multiple Options in a Model

The recycling rules in `mc2d` regarding the `nvariate` dimension are as follows: if needed, the recycling will be done from `nvariates=1` to `nvariates=n` with $n > 1$. This allows you to use multivariates nodes as a third dimension, in case you want to test various alternatives.

Assume, as in section 2.2.5, that the distribution representing uncertainty in `conc` was not certain, and that the microbiologists suggest that $conc \sim N(10, 2)$ is possible, but that $conc \sim U(8, 12)$ is also possible. We can i) build a bivariate node reflecting these two independent options; ii) transfer these options into the final risk estimate. We obtain a bivariate node for the risk, one using the first hypothesis, the second the second hypothesis.

```
> conc1 <- mcstoc(rnorm, type="U", mean=10, sd=2)
> conc2 <- mcstoc(runif, type="U", min=8, max=12)
> conc <- mcdata(c(conc1,conc2),type="U",nvariates=2)
> cook <- mcstoc(rempiricalD, type="V", values=c(1,1/5,1/50), prob=c(0.027,0.373,0.600))
> serving <- mcstoc(rgamma,type="V",shape=3.93,rate=0.0806)
> expo <- conc * cook * serving
> dose <- mcstoc(rpois,type="VU",nvariates=2,lambda=expo)
> r <- mcstoc(runif,type="U",min=0.0005,max=0.0015)
> risk <- 1-(1-r)^dose
> EC5 <- mc(conc,risk)
> summary(EC5)

conc :
[[1]]
       NoVar
median  9.99
mean    9.87
2.5%    6.11
```

```
97.5%  13.07

[[2]]
       NoVar
median  9.99
mean   10.01
2.5%    8.19
97.5%  11.91


risk :
[[1]]
         mean     sd Min     2.5%     25%     50%    75% 97.5%   Max  nsv Na's
median 0.0463 0.0716   0 0.001433 0.00650 0.01317 0.0667 0.229 0.672 1001    0
mean   0.0472 0.0720   0 0.001658 0.00689 0.01365 0.0687 0.235 0.659 1001    0
2.5%   0.0261 0.0428   0 0.000635 0.00372 0.00713 0.0370 0.133 0.440 1001    0
97.5%  0.0760 0.1082   0 0.003346 0.01174 0.02345 0.1150 0.369 0.856 1001    0

[[2]]
         mean     sd       Min     2.5%     25%     50%    75% 97.5%   Max  nsv Na's
median 0.0485 0.0743 0.00e+00 0.001774 0.00697 0.01402 0.0691 0.241 0.693 1001    0
mean   0.0481 0.0733 1.12e-05 0.001760 0.00705 0.01395 0.0700 0.240 0.671 1001    0
2.5%   0.0259 0.0428 0.00e+00 0.000563 0.00351 0.00701 0.0364 0.134 0.458 1001    0
97.5%  0.0769 0.1094 0.00e+00 0.002908 0.01247 0.02305 0.1155 0.377 0.856 1001    0
```

(Do not forget to transfer the number of variates you want in `mcstoc`... (see the definition of `dose`). `mc2d` cannot guess...)

3.3 Multivariate Nodes as a Third Dimension for Multiple Vectors/Contaminants

The recycling rules in `mc2d` also allow you to use multivariate nodes as a third dimension for multiple vectors/Contaminants.

Assume in our ground beef example that we have two contaminants: one has a mean concentration that follows an uncertainty distribution $conc \sim N(10, 2)$, the second one follows $conc \sim N(14, 2)$. We can *i*) build a bivariate node reflecting these two concentrations[15] ; *ii*) transfer these options into the final dose; *iii*) sum the dose over the variates (using `mcapply`). The behavior of contaminants is transferred in the model.

```
> mconc <- mcdata(c(10,14), type="0", nvariates=2)
> conc <- mcstoc(rnorm, nvariates=2, type="U", mean=mconc, sd=2)
> cook <- mcstoc(rempiricalD, type="V", values=c(1,1/5,1/50), prob=c(0.027,0.373,0.600))
> serving <- mcstoc(rgamma,type="V",shape=3.93,rate=0.0806)
> expo <- conc * cook * serving
> dose <- mcstoc(rpois,type="VU",nvariates=2,lambda=expo)
> dosetot <- mcapply(dose, margin="variates", fun=sum)
> r <- mcstoc(runif,type="U",min=0.0005,max=0.0015)
> risk <- 1-(1-r)^dosetot
> EC6 <- mc(conc,risk)
> summary(EC6)

conc :
[[1]]
       NoVar
```

[15] Note that we could simulate a correlation between both contaminants using a multivariate normal distribution.

```
median 10.25
mean   10.00
2.5%    5.61
97.5%  13.68

[[2]]
       NoVar
median  14.1
mean    14.0
2.5%    10.3
97.5%   17.8


risk :
         mean     sd     Min    2.5%     25%    50%    75% 97.5%   Max  nsv Na's
median 0.1017 0.1245 0.00116 0.00588 0.01782 0.0326 0.1671 0.415 0.845 1001    0
mean   0.1051 0.1265 0.00117 0.00617 0.01865 0.0347 0.1734 0.421 0.828 1001    0
2.5%   0.0611 0.0798 0.00000 0.00314 0.00997 0.0187 0.0955 0.259 0.638 1001    0
97.5%  0.1525 0.1729 0.00276 0.00970 0.02912 0.0530 0.2590 0.581 0.951 1001    0
```

As a conclusion, this third dimension is highly flexible...

4 Another Example: A QRA of Waterborne Cryptosporidiosis in France

This example is adapted from [6]. The aim is to evaluate the risk of infection with *Cryptosporidium parvum* from consumption of tap water, given that n oocysts /100 l. have been observed in a storage reservoir. The study is simplified here for illustration purpose and the reference for the subject remains [6].

4.1 Tap Water Consumption Model

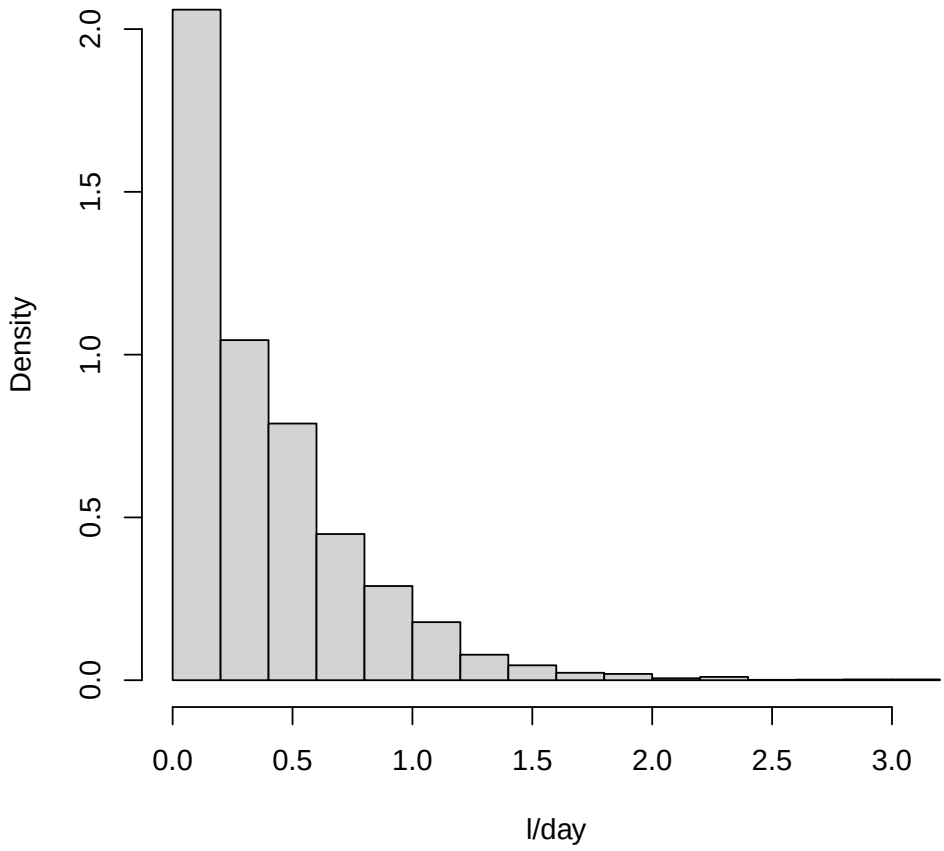
We have raw data of daily consumption of tap water from 1,180 tap water consumers (var `inca`, see Figure 7). We could choose to use this empirical distribution to evaluate the variability in the tap water consumption:

```
> ndvar(1001)

[1] 1001

> ndunc(101)

[1] 101

> mcstoc(rempiricalD,type="V",values=inca)

  node    mode  nsv nsu nva variate min mean median max Nas type outm
1    x numeric 1001   1   1       1   0  0.4    0.3 3.2   0    V each
```

but we will use the `fitdistrplus` [3] library. `inca` includes a lot of `0` nodes, corresponding to days when individuals do not drink tap water (possibly they drink bottled water on those days). We could try a mixture of distributions, with `0` and non-`0` data.

```
> library(fitdistrplus)
> pzero <- sum(inca==0)/length(inca)
> inca_non_0 <- inca[inca!=0]
> descdist(inca_non_0)
```

Figure 7: Histogram of daily tap water intake



```
summary statistics
------
min:  0.0221   max:  3.2
median:  0.48
mean:  0.566
estimated sd:  0.385
estimated skewness:  1.75
estimated kurtosis:  7.99
```

Following the `descdist` function (See figure 8), let us try the lognormal distribution.

```
> Adj_water <- fitdist(inca_non_0,"lnorm",method="mle")
> meanlog <- Adj_water$est[1]
> sdlog   <- Adj_water$est[2]
> summary(Adj_water)

Fitting of the distribution ' lnorm ' by maximum likelihood
Parameters :
        estimate Std. Error
meanlog   -0.784    0.00891
sdlog      0.674    0.00630
Loglikelihood:  -1374   AIC:  2752   BIC:  2765
Correlation matrix:
          meanlog     sdlog
meanlog  1.00e+00 -6.39e-12
sdlog   -6.39e-12  1.00e+00
```

The fit seems correct, and better than the one obtained using a gamma distribution (results not shown). We can now rebuild our mixture. We could consider uncertainty around the maximum likelihood estimates using the `bootdist` function of the `fitdistrplus` [3] package, using something like:

```
> Boot <- bootdist(Adj_water, bootmethod="param", niter=ndunc())
> Mean_conso <- mcdata(Boot$estim$meanlog, type="U")
> Sd_conso <- mcdata(Boot$estim$sdlog, type="U")
> conso1 <- mcstoc(rlnorm, type="VU", meanlog= Mean_conso, sdlog= Sd_conso)
```

But for simplicity, we will not consider uncertainty around the estimates.

We will use the `mcprobtree` function to construct a mixture of 0 and non-0 distributions:

```
> conso0 <- mcdata(0,type="V")
> conso1 <- mcstoc(rlnorm, type="V", meanlog=meanlog, sdlog=sdlog)
> v <- mcprobtree(c(pzero,1-pzero), list("0"=conso0,"1"=conso1), type = "V")
> summary(v)

node :
       mean    sd Min 2.5% 25%   50%   75% 97.5%  Max  nsv Na's
NoUnc 0.394 0.419   0    0   0 0.313 0.597  1.41 2.98 1001    0
```

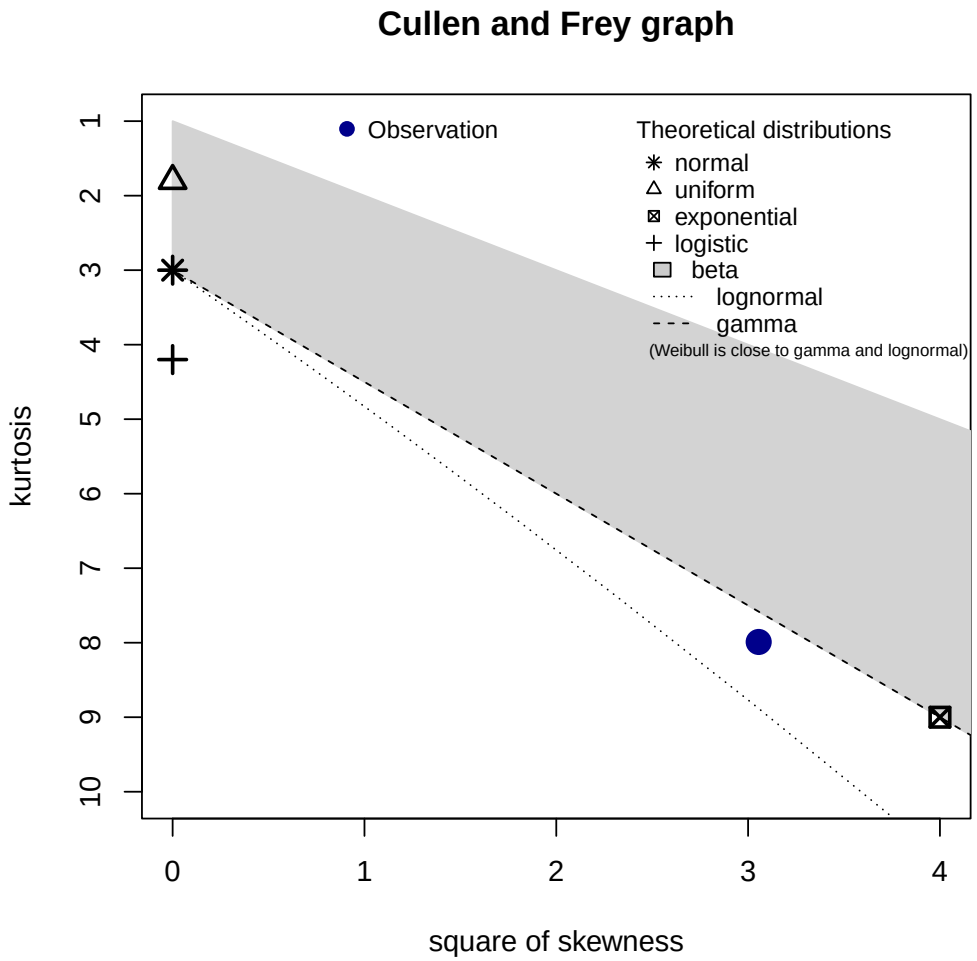
4.2 The Dose-Response Model

We propose a bootstrap from data (`datDR`) derived from [1]. We first define a function `DR` with an `n` argument for the size of the sample to draw. This function may then be used in a `mcstoc` function:

```
> datDR <- list(   dose=c(30,100,300,500,1000,10000,100000,1000000),
+                  pi=c(2,4,2,5,2,3,1,1),
```

Figure 8: Graph from the `descdist` function.

```
summary statistics
------
min:  0.0221   max:  3.2
median:  0.48
mean:  0.566
estimated sd:  0.385
estimated skewness:  1.75
estimated kurtosis:  7.99
```



```
+                  ni=c(5,8,3,6,2,3,1,1))
> estDR <- function(pos,ref){
+    suppressWarnings(
+         -glm(cbind(ref$ni-pos,pos) ~ ref$dose + 0,
+                 binomial(link="log"))$coefficients)}
> ml <- 1-exp(-estDR(datDR$pi, datDR) * datDR$dose)
> DR <- function(n){
+    boot <- matrix(rbinom(length(datDR$dose)*n,datDR$ni,ml),nrow=length(datDR$dose))
+    apply(boot,2,estDR,ref=datDR)}
> r <- mcstoc(DR, type="U")
> summary(r)

node :
         NoVar
median 0.00536
mean   0.00600
2.5%   0.00274
97.5%  0.01113
```

4.3 The Model

Deriving the final model is straightforward. We construct the `mcnode` corresponding to the recovery rate (Uncertainty, `Rr`), the probability for an oocyst to be infective (Variability, `w`):

```
> Rr <- mcstoc(rbeta, type="U", shape1=2.65, shape2=3.64)
> w <- mcstoc(rbeta, type="V", shape1=2.6, shape2=3.4)
```

Given that $O_o = 2$ oocysts are observed in 100 l of water, the expected number of oocysts in the sample is `l`:

```
> Oo <- 2
> l <- (Oo + mcstoc(rnbinom, type="U", size=Oo+1, prob=Rr))/100
```

The expected number of oocysts drunk by the individuals is `Or` and the risk ($\times 10000$) is estimated by:

```
> Or <- l * v * w
> P <- 10000 * (1-exp(-r*Or))
> summary(P)

node :
        mean    sd Min 2.5% 25%   50%   75%  97.5%   Max  nsv Na's
median 0.486 0.572   0    0   0 0.321 0.713  2.026  3.97 1001    0
mean   0.687 0.809   0    0   0 0.454 1.008  2.864  5.61 1001    0
2.5%   0.162 0.191   0    0   0 0.107 0.238  0.675  1.32 1001    0
97.5%  2.411 2.837   0    0   0 1.594 3.538 10.047 19.67 1001    0
```

This result can be compared (roughly since there are some differences in the model variability) to the results shown in Table 2 in [6].

Improvement: the results for $O_o = \{0, 1, 2, 5, 10, 20, 50, 100, 1000\}$ can be obtained in one step using:

```
> Oo <- mcdata(c(0,1,2,5,10,20,50,100,1000),type="0",nvariates=9)
```

As a Conclusion

We think and hope that `mc2d` could help risk assessors to construct and analyse their models, and that it may help in developing two-dimensional simulations. *Please report any bugs you get to rpouillot@yahoo.fr.*

If you would like to improve `mc2d`, join us at

http://riskassessment.r-forge.r-project.org/

References

[1] C.L. Chappell, P.C. Okhuysen, C.R. Sterling, and H.L. DuPont. *Cryptosporidium parvum*: intensity of infection and oocyst excretion patterns in healthy volunteers. *Journal of Infectious Diseases*, 173(1):232–236, 1996.

[2] A.C. Cullen and H.C. Frey. *Probabilistic techniques in Exposure assessment.* Plenum Press, New York, 1999.

[3] Marie Laure Delignette-Muller and Christophe Dutang. fitdistrplus: An R package for fitting distributions. *Journal of Statistical Software*, 64(4):1–34, 2015.

[4] R.L. Iman and W.J. Conover. A distribution-free approach to inducing rank correlation among input variables. *Communication in Statistics*, B11(3):311–334, 1982.

[5] H. Ozkaynak, H.C. Frey, J. Burke, and R.W. Pinder. Analysis of coupled model uncertainties in source-to-dose modeling of human exposures to ambient air pollution: A pm2.5 case study. *Atmospheric environment*, 43(9):1641–1649, 2009.

[6] R. Pouillot, P. Beaudeau, J.-B. Denis, F. Derouin, and AFSSA Cryptosporidium Study Group. A quantitative risk assessment of waterborne cryptosporidiosis in france using second-order monte carlo simulation. *Risk Anal*, 24(1):1–17, 2004.

[7] R. Pouillot, N. Miconnet, A.-L. Afchain, M.-L. Delignette-Muller, A. Beaufort, L. Rosso, J.-B. Denis, and M. Cornu. Quantitative risk assessment of listeria monocytogenes in french cold-salmon : I. quantitative exposure assessment. *Risk Analysis*, 27(3):683–700, 2007.

[8] D. Vose. *Risk Analysis, A quantitative guide, 2nd Edition.* Wiley and Sons, Chichester, 2nd edition, 2000.