The "mc2d" package.

Régis POUILLOT, Marie-Laure DELIGNETTE-MULLER & Jean-Baptiste DENIS

November 14, 2008

Contents

This documentation is intended for readers with:

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

This documentation will not described all arguments of the functions. The reference remains the documentation attached to the package.

1 Introduction

1.1 What is `mc2d`?

"`mc2d`" means Two-Dimensional Monte-Carlo (*"Monte-Carlo à Deux Dimensions"*). This package :

- provides additional distributions;
- provides tools to build One-Dimensional and Two-Dimensional Monte-Carlo Simulations;
- provides tools to study One-Dimensional and Two-Dimensional Monte-Carlo Simulations.

In a previous version, some tools to fit parametric distributions to data were included. These functions being useful for other purposes, they have been placed in a specific package called `specdist`. These packages are available at the URL `https://r-forge.r-project.org/projects/riskassessment/`.

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

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

The following text and Figure 1 are adapted from [4] and [5] where this method was used. The major reference for Two-Dimensional Monte-Carlo simulations remains [2].

According to international recommendations, a QRA should reflect the "variability" of the risk and estimate the "uncertainty" of the risk estimate. The "variability" represents the temporal, geographical and/or individual heterogeneity of the risk for a given population. The "uncertainty" is understood as the lack of perfect knowledge of the QRA model structure and parameters.

In order to estimate the natural "variability" of the risk, a Monte-Carlo simulation approach may be useful: the empirical distribution of the risk within the population may be estimated from the mathematical combination of distributions reflecting the variability of parameters in the population.

In order to estimate the "uncertainty" of the risk estimates issued from data uncertainty, a two-dimensional (or second-order) Monte-Carlo simulation was proposed [2]. A two-dimensional Monte-Carlo simulation is a Monte-Carlo simulation where the alea reflectiong "variability" and the alea reflecting "uncertainty" are transfered separately in the simulation, so that "variability"and "uncertainty" of the output may be estinated separately. It may be described as following (see Figure 1):

1. The parameters of the model should be divided in three categories: the parameters whose alea reflects "variability only", hereinafter denoted as "variable parameters", the parameters whose alea reflects "uncertainty only", denoted as "uncertain parameters" and the parameters whose alea reflects uncertainty and variability. For this latter category, a hierarchical structure, using "hyper-parameters", should be specified: if a parameter is uncertain and variable, one should be able to specify an empirical or parametric distribution reflecting variability only conditionally on other parameters known with uncertainty. As an example, one should be able to set a structure as $X\,|a,b \sim N\,(a,b)$, this normal distribution reflecting variability of x conditionally to a and b, with, e.g., $a \sim Unif\,(l_a,u_a)$and $b \sim Unif\,(l_b,u_b)$, these distributions reflecting the uncertainty around the parameters a and b;
2. A set of uncertain parameters are randomly sampled from their respective distributions;
3. The QRA is performed using a classical (one-dimensional) Monte-Carlo simulation of size N_v, conditionally to these uncertain parameters considered as fixed. This QRA takes into account the variability of all variable parameters, and leads to an empirical density function reflecting the variability of exposure/risk among the population conditionally to the uncertain parameters. Various statistics (*e.g.* the mean, the standard deviation, some percentiles) of the resulting empirical density are evaluated and stored;
4. Steps 2) and 3) are performed a large number of time (N_u times), leading to N_u set of statistics;
5. As output, the 50^{th} percentile (median) of each statistic is used to establish an 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 represent a "variability cumulative distribution" using 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.

"`mc2d`" is a set of R functions that will help to develop such two-dimensional Monte-Carlo simulations. The main difference with the procedure described above is that `mc2d` will use 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 further the method, but the practical use of `mc2d`, based on a fictive example.

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.

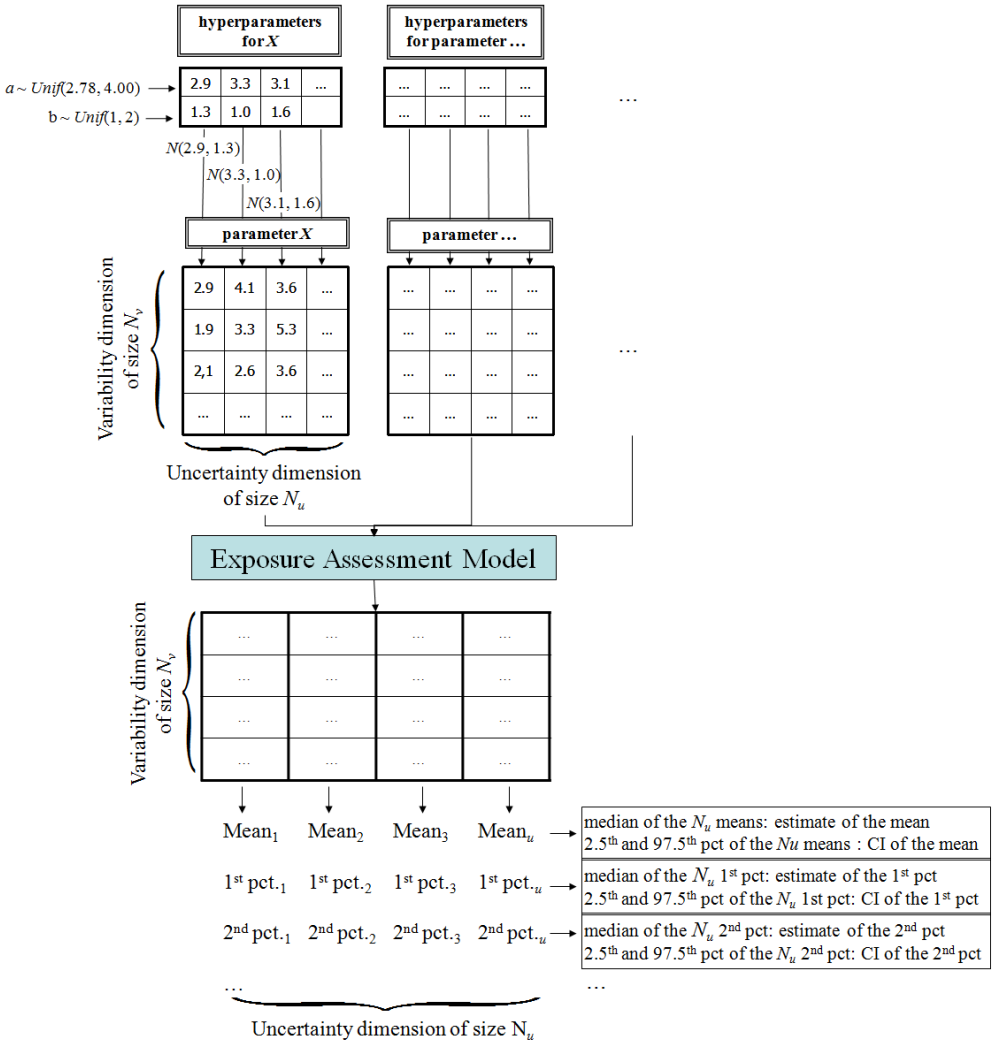


Figure 1: Shematic Representation of a Two-Dimensional Monte-Carlo Simulation.

- We assume 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;
- We assume that no bacterial growth occurs, since the product is kept frozen until cooked, just before consumption;
- 2.7% of consumers cook their beef "rare", 37.3% "medium" and 60.0% "well cooked";
- The following inactivation i is associated to 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 distribution of steak serving sizes s for <3 year children was estimated in a consumption survey. The "best fit" was obtained using 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 and $r = 0.001$.

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

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

1.3.1 One Dimensional Monte-Carlo Simulation

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

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

where $emp\left(\boldsymbol{X}, \boldsymbol{P}\right)$ is the empirical distribution where each value X_i is associated to the probability P_i.We will use a "classical" one dimensional Monte-Carlo simulation, with 1000 iterations. Using the "`mc2d`" package, the model may be written as:

```
> library(mc2d)
> ndvar(1000)

[1] 1000

> conc <- 10
> cook <- mcstoc(rempiricalD, values = c(1, 1/5, 1/50), prob = c(0.027,
+     0.373, 0.6))
> 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
1    cook numeric 1000   1   1       1 0.02  0.1165  0.0200   1.000   0    V
2 serving numeric 1000   1   1       1 5.17 48.4451 44.0195 219.976   0    V
3    expo numeric 1000   1   1       1 1.03 56.2452 14.1530 935.189   0    V
4    dose numeric 1000   1   1       1 0.00 56.0520 15.0000 938.000   0    V
5    risk numeric 1000   1   1       1 0.00  0.0507  0.0149   0.609   0    V
  outm
1 each
2 each
3 each
4 each
5 each

> summary(EC1)

cook :
       mean    sd  Min 2.5%  25%  50% 75% 97.5% Max  nsv Na's
NoUnc 0.116 0.176 0.02 0.02 0.02 0.02 0.2     1   1 1000    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  44 62.6   103 220 1000    0

expo :
      mean   sd  Min 2.5%  25%  50%  75% 97.5% Max  nsv Na's
NoUnc 56.2 96.8 1.03  3.5 8.11 14.2 79.1   229 935 1000    0

dose :
      mean   sd Min 2.5% 25% 50% 75% 97.5% Max  nsv Na's
NoUnc   56 96.3   0    2   7  15  79   226 938 1000    0

risk :
        mean     sd Min  2.5%     25%    50%   75% 97.5%   Max  nsv Na's
NoUnc 0.0507 0.0755   0 0.002 0.00698 0.0149 0.076 0.203 0.609 1000    0
```

This One-Dimensional Monte-Carlo simulation provides an estimates of the mean risk (around 5%), as well as some quantiles of the risk (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. It was an estimate. Microbiologists think that the uncertainty around this estimate might be reflected through a normal distribution with parameters $\mu = 10$ and $\sigma = 2$;

- Epidemiological studies suggest that the r parameter is known with uncertainty. The uncertainty around the mean value 0.001 may be reflected through a uniform distribution bounded within 0.0005 and 0.0015.

The model could then be written as:

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

Nevertheless, the distributions of r and c do not reflect the same kind of alea then do the distributions of i and s. r and c are uncertain, while i and s are variable. n, as a function of c, i and s will be variable *and* uncertain.

We will use a two dimensional Monte-Carlo simulation, with 1000 iterations in the variability dimension and 100 iterations in the uncertainty dimension. Using the "`mc2d`" package, the model may be written as:

```
> ndunc(100)

[1] 100

> 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)
> expo <- conc * cook * serving
> dose <- mcstoc(rpois, type = "VU", lambda = expo)
> r <- mcstoc(runif, type = "U", min = 5e-04, max = 0.0015)
> risk <- 1 - (1 - r)^dose
> EC2 <- mc(conc, cook, serving, expo, dose, r, risk)
> EC2

     node    mode  nsv nsu nva variate      min     mean   median      max Nas
1    conc numeric    1 100   1       1 5.557709 9.94e+00 9.72e+00 1.75e+01   0
2    cook numeric 1000   1   1       1 0.020000 1.07e-01 2.00e-02 1.00e+00   0
3 serving numeric 1000   1   1       1 2.665858 4.97e+01 4.50e+01 1.61e+02   0
4    expo numeric 1000 100   1       1 0.705346 5.31e+01 1.37e+01 1.68e+03   0
5    dose numeric 1000 100   1       1 0.000000 5.31e+01 1.40e+01 1.71e+03   0
6       r numeric    1 100   1       1 0.000514 9.62e-04 9.02e-04 1.48e-03   0
7    risk numeric 1000 100   1       1 0.000000 4.55e-02 1.36e-02 8.40e-01   0
  type outm
1    U each
2    V each
3    V each
4   VU each
5   VU each
6    U each
7   VU each

> summary(EC2)
```

```
conc :
       NoVar
median  9.72
mean    9.94
2.5%    5.96
97.5%  14.46

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  0.22   1 1000    0

serving :
      mean   sd  Min 2.5% 25% 50%  75% 97.5% Max  nsv Na's
NoUnc 49.7 24.9 2.67 13.6  31  45 64.2   110 161 1000    0

expo :
       mean    sd   Min 2.5%   25%   50%   75% 97.5%  Max  nsv Na's
median 51.9  94.2 1.234 3.06  7.87 13.58  71.8   240  938 1000    0
mean   53.1  96.3 1.261 3.12  8.04 13.89  73.4   245  959 1000    0
2.5%   31.8  57.8 0.756 1.87  4.82  8.33  44.0   147  575 1000    0
97.5%  77.2 140.2 1.836 4.55 11.71 20.21 106.8   357 1396 1000    0

dose :
       mean    sd  Min 2.5%   25%  50%   75% 97.5%  Max  nsv Na's
median 51.9  94.7 0.00    2  7.00 14.0  71.8   242  958 1000    0
mean   53.1  96.7 0.04    2  7.53 14.2  73.4   245  964 1000    0
2.5%   31.7  57.8 0.00    1  4.47  9.0  43.5   146  573 1000    0
97.5%  77.6 140.7 1.00    3 11.00 20.5 107.8   355 1379 1000    0

r :
          NoVar
median 0.000902
mean   0.000962
2.5%   0.000525
97.5%  0.001459

risk :
         mean     sd      Min     2.5%     25%     50%    75%  97.5%   Max  nsv
median 0.0445 0.0703 0.00e+00 0.001713 0.00687 0.01298 0.0645 0.2027 0.589 1000
mean   0.0455 0.0706 3.88e-05 0.001902 0.00717 0.01347 0.0674 0.2061 0.582 1000
2.5%   0.0191 0.0324 0.00e+00 0.000583 0.00282 0.00538 0.0271 0.0841 0.290 1000
97.5%  0.0730 0.1057 7.08e-04 0.004115 0.01226 0.02242 0.1116 0.3259 0.788 1000
       Na's
median    0
mean      0
2.5%      0
97.5%     0
```

Note that the syntax is similar. Nevertheless, for each distribution, a "`type`" argument is provided, indicating if the parameter distribution reflects uncertainty (`type`="`U`"), variability (`type`="V"), or both (`type`="`VU`").

The summary now provides estimates of the variability distribution (in row) but with a measure of their uncertainty, linked to the uncertainty around `conc` and `r`. The estimate of the mean risk is now known with uncertainty. The median of the 100 simulations lead to a "best estimate"of 0.0445, with a "credible interval" of [0.191, 0.0730].

2 Basic Principles and Functions

A classical session of R using "mc2d" is as following:

- From data, expert knowledge, *etc.* empirical or parametric distributions are chosen for each "parents" parameters. For data fitting, the "specdist" package is recommended;
- For each parameter, an mcnode object is built (key functions: mcdata, mcstoc);
- Various mcnode objects are grouped in a mc object (key function: mc).
- The mc object is studied through summaries, graphs, sensitivity analysis (key functions: summary.mc, plot.mc, tornado, tornadounc).

2.1 Before All

The "mc2d" library should be loaded during 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. It 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[1]. Four types of mcnode exist:

- "V" mcnode, for "*Variability*", are arrays of dimension ($nsv \times 1 \times nvariates$). The alea in the data reflects variability of the parameter;
- "U" mcnode, for "*Uncertainty*", are arrays of dimension ($1 \times nsu \times nvariates$). The alea in the data reflects uncertainty of the parameter.
- "VU" mcnode, for "*Variability and Uncertainty*", are arrays of dimension ($nsv \times nsu \times nvariates$). The alea in the data reflects separated variability (in the first dimension) and uncertainty (in the second dimension) of the parameter.
- Additionally, "0" mcnode are defined for some use. "0" stand for "Neither Variability or Uncertainty". They are arrays of dimension ($1 \times 1 \times nvariates$). No alea is considered for these nodes. "0" mcnode are not necessary in the univariate context (use scalar instead) but are useful to build multivariate nodes (See section 3).

There are 5 ways to build a mcnode object:

1. The mcstoc function builds mcnode from random generating functions;
2. The mcdata function fills mcnode from data sets;
3. mcnode are built directly from operations on mcnode objects;
4. mcprobtree is a special function that builds an mcnode from various mcnode using a probability tree;
5. Some functions, as "==" or ">" , is.na, is.finite build an mcnode when applied to an mcnode.

[1] In this section, we will only consider mcnode with $nvariates = 1$.

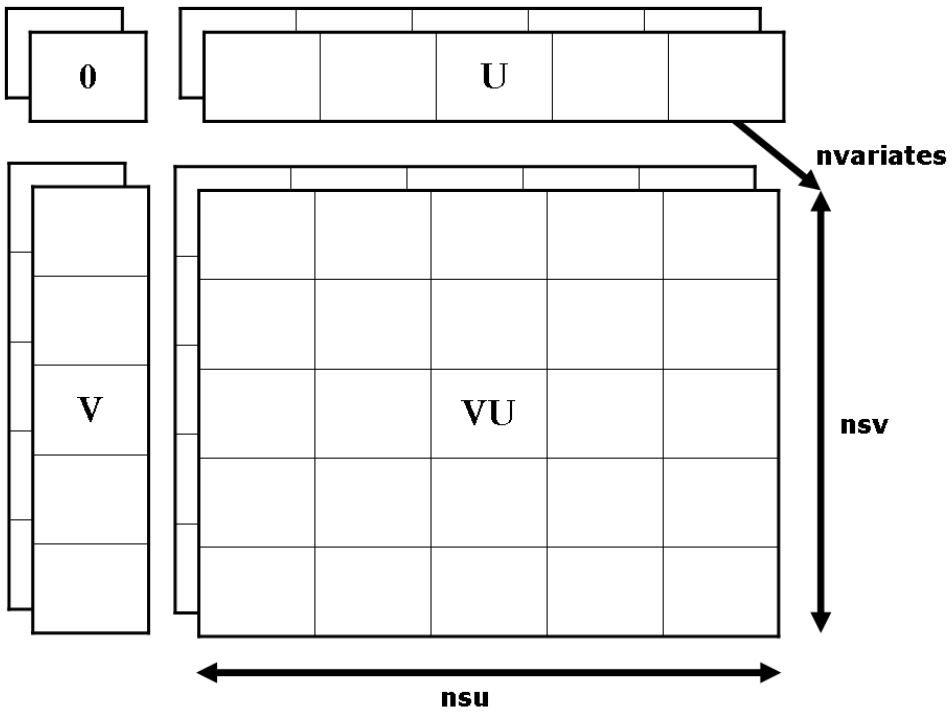


Figure 2: Structure of the various `mcnode` objects.

2.2.2 The `mcstoc` function

The `mcstoc` function is written as[2]:

```
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 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` builds a "`V`" `mcnode`;
- ... 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`. If the name of the argument specifying the size of the sample is not `n` (*e.g.* functions `rhyper` and `rwilcox`, see table 1), the name of this parameter should be provided in the `nsample` argument. *Note that all following arguments should be named*;
- `nsv` and `nsu` are the number of values needed in the variability and the uncertainty dimension, respectively. By default, these values are the one provided by `ndvar()` and `ndunc()`, respectively;
- `nvariates` is the desired number of variates in the `mcnode`;
- `outm` is the default output for multivariate nodes;
- `seed` optionally specify a seed for the random generator;

[2]as classically in R, most of arguments have logical default values and will be infrequently modified.

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
	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
	multinormal	`rmultinormal`	`n`	`mean, sigma`	N	N

- `rtrunc` allows to truncate the distribution between `linf` and `lsup`. This function is not valid for all distributions (see table 1). See the `rtrunc` function help for further details;
- `lhs` allows to sample the node in a Latin Hypercube Sampling framework. This function is not valid for all distributions (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 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, 0.6))
> serving <- mcstoc(rgamma, type = "V", shape = 3.93, rate = 0.0806)
> ...
> dose <- mcstoc(rpois, type = "VU", lambda = expo)
> r <- mcstoc(runif, type = "U", min = 5e-04, max = 0.0015)
> ...
```

A normal distribution with parameters $mean = 2$, $sd = 3$, truncated on [1.5, 2] with a Latin Hypercube Sampling could be written[3]:

[3]Note that the mean and the standard deviation of the Gaussian distribution are not kept due to the truncation.

```
> 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 1000    0
```

For your use in `mcstoc`, additionnal distributions have been implemented: the Bernoulli distribution (`rbern`), the empirical discrete distribution (`rempiricalD`), the PERT distribution (`rpert`)[6], the triangular distribution (`rtriang`), the Dirichlet distribution (`rdirichlet`) and the multinormal distribution (`rmultinormal`). The multinomial distribution has been adapted (vectorized): `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 multinormal may be used with uncertain and/or variable parameters using multivariate nodes. See section 3.

2.2.3 The `mcdata` function

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

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

See the documentation associated to this function to see the size/mode of data that can be used to specify an `mcnode`. The following example place 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   1 100   1       1   0  0.5    0.5   1   0    U each
```

2.2.4 Operations on `mcnode`

`mcnode`s are automatically built using operations on `mcnode`. Rules are built to transfer coherently uncertainty and variability within the model. Logically, the rules are as following (illustrated here with a "+")[4]:

- "`0`" + "`0`" = "`0`";
- "`0`" + "`V`" = "`V`"
- "`0`" + "`U`" = "`U`";
- "`0`" + "`VU`" = "`VU`";
- "`V`" + "`V`" = "`V`";
- "`V`" + "`U`" = "`VU`": the "`U`" mcnode is recycled by row, the "`V`" mcnode is recycled classically by column;
- "`V`" + "`VU`" = "`VU`": the "`V`" mcnode is recycled classically by column;

[4] These rules are not classical R rules of recycling.

- "U" + "U" = "U";
- "U" + "VU" = "VU": the "U" mcnode is recycled by row;
- "VU" + "VU" = "VU"

Thus, in our example:

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

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

2.2.5 The mcprobtree function

The mcprobtree function should be used if a "probability tree" is needed to build an mcnode. Assume that the distribution reflecting the uncertainty on conc was not sure, and that the microbiologists suggest that they are 75% confident that $conc \sim N(10, 2)$ but that they are 25% confident that $conc \sim U(8, 12)$. This could be written using mcprobtree as[5]:

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

mcprobtree could also be used to provide a mixture distribution in the variability dimension.

2.2.6 Other functions to build mcnode

The functions "==", "<", "<=", ">=", ">", provides an mcnode when applied on a 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 not.

```
> cook < 1

  node    mode  nsv nsu nva variate min  mean median max Nas type outm
1    x logical 1000   1   1       1   0 0.975      1   1   0    V each

> 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 1000   1   1       1 -8.19 -1.03 -0.699 -0.00167 512    V each

> is.na(tmp)

  node    mode  nsv nsu nva variate min  mean median max Nas type outm
1    x logical 1000   1   1       1   0 0.512      1   1   0    V each
```

[5] an alternative for whichdist could be whichdist <- mcstoc(rempiricalD, type="U", values=c(0,1), prob=c(75,25))

2.2.7 Building correlation between mcnode

Structural links between set of parameters may be very important in QRA. In mc2d, a correlation structure (in the sense of Spearman) implying 2 or more nodes may be built with the cornode function. This function use the Iman & Conover method [3]. Assume that a study suggests that people that eat their ground beef "rare" eat bigger serving sizes. We could build this relation using:

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

output Rank Correlation per variates
variates: 1
[1] 1.0000000 0.3796997 0.3796997 1.0000000
$cook
  node    mode  nsv nsu nva variate  min  mean median max Nas type outm
1    x numeric 1000   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 1000   1   1       1 2.67 49.7     45 161   0    V each
```

Note that the resulting correlation (around 0.4) is obviously an approximation in this case, where a discrete distribution (cook: 3 categories) is correlated to a continuous distribution (serving).

It is possible to create such correlation between "V" nodes, between " U" nodes, between "VU" nodes or between one "V" node and some "VU" nodes.

The use of a multinormal distribution (rmultinormal) is another way to create such relationship between nodes.

2.3 The mc Object

Once your mcnode objects are built, one should group them in a single object to study the Monte-Carlo results. The "mc" object is a list of mcnode. There are three ways to build a mc object: using the mc function, the evalmcmod function or within the evalmccut function.

2.3.1 The mc Function

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

... are mcnode or mc objects to be gathered in a 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)
```

2.3.2 The mcmodel and the evalmcmod Functions

A model may be written in one step using mcmodel (just a wrap of your model in a function), and then evaluated using evalmcmod. These functions may be used once your model is correct and 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, 0.6))
+     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)
+ })
> 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[6];
- The model should end by an `mc()` function.

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

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

The interest lay in the possibility to re-run the model with various 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

If you want to evaluate a high dimension model, R may reach its memory limit. `evalmccut` evaluates a 2-dimensional Monte-Carlo model (written with the `mcmodelcut` function) using a loop, 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:

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

```
> modEC4 <- mcmodelcut({
+     {
+         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)
+     }
+     {
+         expo <- conc * cook * serving
+         dose <- mcstoc(rpois, type = "VU", lambda = expo)
+         risk <- 1 - (1 - r)^dose
+         res <- mc(zero, conc, cook, serving, expo, dose, r, risk)
+     }
+     {
+         list(sum = summary(res), plot = plot(res, draw = FALSE),
+             minmax = lapply(res, range), tor = tornado(res),
+             et = sapply(res, sd))
+     }
+ })
> evalmccut(modEC4, nsv = 10001, nsu = 101, seed = 666, progress.bar = TRUE)
```

Note that the use of a `tornado` function in the model should be avoided since it slows considerably the `evalmccut` function. The `tornado` function will be rewritten in the near future.

2.4 Studying 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). Obviously, other more informative functions are provided in the `mc2d` package.

2.4.1 The `summary` Function

The `summary` function provide statistics on the `mc` object:

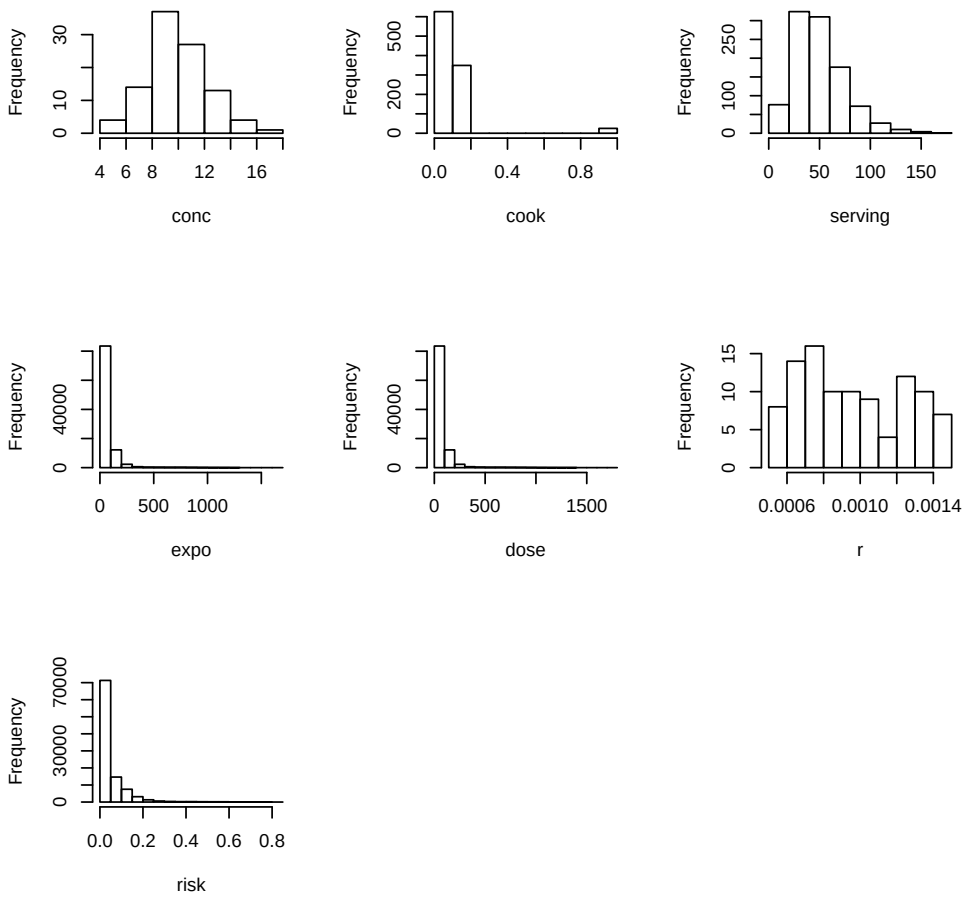
```
summary(object, probs=c(0,0.025,0.25,0.5,0.75,0.975,1), lim=c(0.025,0.975), ...)
```

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. Of course, these arguments should be changed if other quantiles are needed.

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

```
             mean         sd     99.5%     99.9%  nsv Na's
median 0.04446930 0.07028198 0.5016035 0.5573356 1000    0
mean   0.04554518 0.07058057 0.4979376 0.5522955 1000    0
2.5%   0.01914973 0.03243724 0.2380268 0.2771793 1000    0
97.5%  0.07297994 0.10573336 0.7113664 0.7565299 1000    0
attr(,"type")
[1] "VU"
```

Figure 3: Function `hist`.



2.4.2 The `hist` Function

The `hist` provides an histogram of the different `mcnode` of the `mc` object (cf. Figure 3).

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

In the current version, uncertainty and variability distributions are collapsed. The histogram might be meaningless.

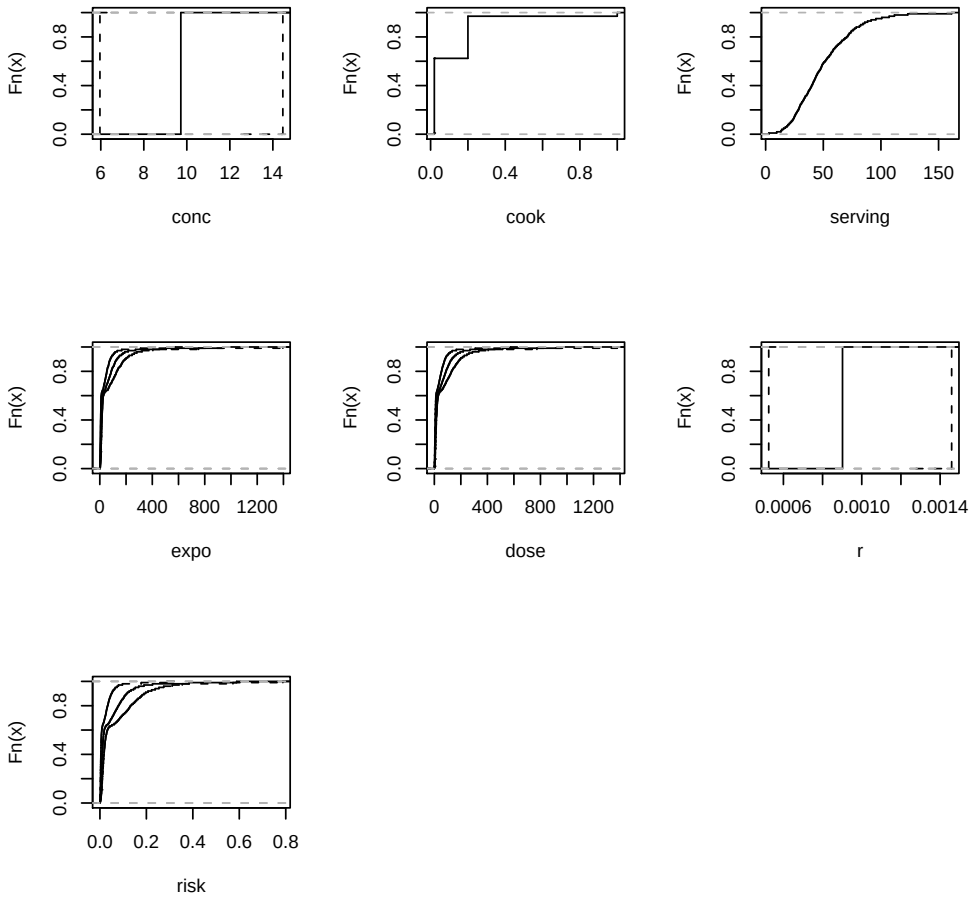
```
> hist(EC2)
```

2.4.3 The `plot` function

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

```
plot (x, prec = 0.01, stat = c("median", "mean"), lim = c(0.025,0.975), na.rm = TRUE,
griddim = NULL, xlab = NULL, ylab = "Fn(x)", main = "", draw = TRUE, ...)
```

Figure 4: `plot` Function .



For our example, see the Figure 4 as a default graph.

```
> plot(EC2)
```

Note that `mcnode` objects have the same methods `print`, `summary`, `plot`, and `hist`.

2.4.4 The `tornado` function

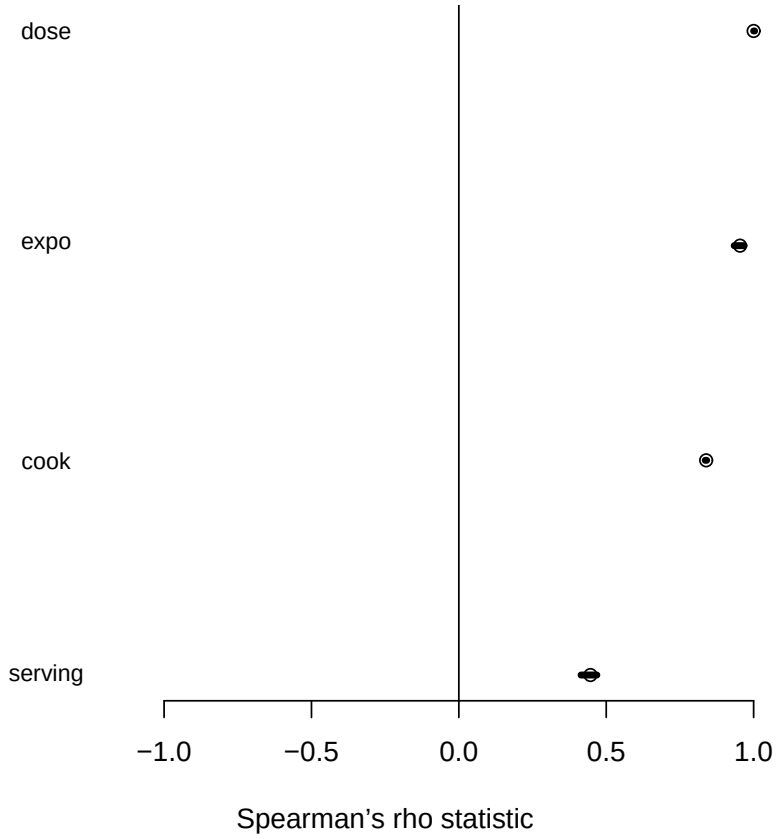
The `tornado` function builds the Spearman (default) rank correlation between nodes of the `mc` object.

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

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 5).

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

Figure 5: `plot.tornado` Function .



2.4.5 The `tornadounc` function

The `tornadounc` explore the impact of the uncertainty on the uncertainty of an output. It builds the Spearman (default) rank correlation between statistics of the `mc` object calculated 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 = 0.99)

Tornado on uncertainty
Spearman's rho statistic
Output:  risk
$risk
               conc mean expo   sd expo  99% expo mean dose   sd dose  99% dose
mean risk 0.5244044 0.5244044 0.5244044 0.5244044 0.5220402 0.5235644 0.4966712
sd risk   0.5233123 0.5233123 0.5233123 0.5233123 0.5210321 0.5229043 0.4958971
99% risk  0.5233603 0.5233603 0.5233603 0.5233603 0.5214761 0.5233003 0.5088284
                  r
mean risk 0.7727933
sd risk   0.7728413
99% risk  0.7607921
```

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

2.5 Other Functions and `mc` Objects

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

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

Here is a (silly) example building a linear model (in fact `ndunc()` linear models) between the `risk` and the `dose` within each uncertainty dimension and studying some statistics on the coefficients. This example is here only to show you that the whole power of R 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.      0.0007991     0.0004028
1st Qu.   0.0038060     0.0005948
Median    0.0064130     0.0007084
Mean      0.0072600     0.0007334
3rd Qu.   0.0092290     0.0008837
Max.      0.0206100     0.0011200
```

3 Multivariate Nodes

The dimension `nvariates` is the third dimension of the `mcnode`. One can use `mc2d` ignoring it. Nevertheless, its use is mandatory to deal with some multivariate distributions, and it may be useful in some circonstances. Building multivariate nodes is direct. We just have to say that the following code:

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

will logically not provide a nodes with 3 variates, each having a different limit. The recycling rule tells you that `c(1, 2, 3)` will be used 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 1000   1   3       1 1.00 2.54   2.58   4   0    V each
2    x numeric 1000   1   3       2 2.00 3.00   3.00   4   0    V each
3    x numeric 1000   1   3       3 3.00 3.52   3.52   4   0    V each
```

3.1 Multivariate Nodes for Multivariate Distributions

The basic usage of multivariate nodes (and the reason why it has been implemented) is for multivariate distributions such as the dirichlet distribution, the multinomial distribution, the multinormal distribution and, possibly, the empirical distribution

As an example, assume that 3-member families buy 500 g of ground beef. The proportion of steak eaten by the baby, his older brother and his mom follow a Dirichlet (uncertainty) distribution of parameter $\alpha = (2, 3, 5)$. You want to derive the distribution (variability) of steak eaten by 500 babies issued from these 500 families.

```
> (p <- mcstoc(rdirichlet, type = "U", nsu = 100, nvariates = 3,
+     alpha = c(2, 3, 5)))

  node    mode nsv nsu nva variate    min  mean median   max Nas type outm
1    x numeric   1 100   3       1 0.0198 0.196  0.170 0.647   0    U each
2    x numeric   1 100   3       2 0.0389 0.297  0.283 0.685   0    U each
3    x numeric   1 100   3       3 0.1968 0.507  0.512 0.846   0    U each

> s <- mcstoc(rmultinomial, type = "VU", nsv = 500, nsu = 100,
+     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  85.0  8.34  60.50  69.00  79.0  85.0  90.5 101.8 109.5 500    0
mean    98.1  8.16  74.28  82.60  92.5  98.0 103.5 114.2 123.8 500    0
2.5%    15.7  3.68   6.47   8.95  13.4  15.4  17.9  23.4  28.3 500    0
97.5%  249.1 11.29 216.65 226.78 241.5 249.0 256.5 270.8 281.0 500    0

[[2]]
        mean    sd   Min  2.5%   25%   50%   75% 97.5%   Max nsv Na's
median 141.3 10.06 113.0 121.7 135.0 141.5 148.0 160.3 173.5 500    0
```

```
mean   148.5  9.55 120.2 130.3 141.9 148.5 154.9 167.1 178.2 500    0
2.5%    24.4  4.89  11.4  15.6  20.9  23.9  27.4  34.4  38.3 500    0
97.5%  319.8 11.34 289.8 298.9 311.9 320.4 327.9 341.6 351.7 500    0

[[3]]
       mean    sd   Min  2.5% 25% 50% 75% 97.5% Max nsv Na's
median  256 10.78 221.0 234.7 248 256 264   278 290 500    0
mean    253 10.70 221.5 232.7 246 253 261   274 286 500    0
2.5%    114  9.04  88.8  96.7 108 114 121   134 148 500    0
97.5%   380 11.67 347.0 360.3 374 381 387   399 409 500    0
```

Assume that each member of these families eat a "normal" distribution (variability) of steak with mean 100, 150 and 250 g. There is a positive correlation between the serving of the children, and a negative one with the one of the mother. You want to derive the distribution (variability) of steak eaten by 500 babies.

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

  node    mode  nsv nsu nva variate   min mean median max Nas type outm
1    x numeric 1000   1   3       1  88.4  100    100 110   0    V each
2    x numeric 1000   1   3       2 141.3  150    150 160   0    V each
3    x numeric 1000   1   3       3 239.0  250    250 260   0    V each

> cor(x[, 1, ])

           [,1]       [,2]       [,3]
[1,]  1.0000000  0.1822931 -0.4950757
[2,]  0.1822931  1.0000000 -0.4884462
[3,] -0.4950757 -0.4884462  1.0000000
```

In this example, `mean` could be variable or uncertain, as well as `sigma`[7]. 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 = c(10, 2, -5, 2, 10, -5, -5, -5, 10))
> cor(x[, 1, ])

           [,1]       [,2]       [,3]
[1,]  1.0000000  0.1817660 -0.5168595
[2,]  0.1817660  1.0000000 -0.4903274
[3,] -0.5168595 -0.4903274  1.0000000
```

The correlation is preserved, but the mean of each categories is known with uncertainty.

Multivariate nodes may finally be useful to derive non parametric bootstrap. Assume that, from a study, you obtained 6 individuals that eat 100 g, 12 individuals that eat 150 g, 6 individuals that eat 170 g and 6 individuals that eat 200 g of ground beef. You want to derive a non parametric bootstrap to derive uncertainty [2], and then pick in the empirical distribution.

[7] Caution: the use of a varying `sigma` would be very slow.

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

  node    mode nsv nsu nva variate min mean median max Nas type outm
1    x numeric   1 100  30      NA 100  100    100 100   0    U  min
2    x numeric   1 100  30      NA 143  154    154 168   0    U mean
3    x numeric   1 100  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 1000 100   1       1 100  154    150 200   0   VU each
```

Printing the statistics of the 30 variates of `x` has no interest. Instead, we use the "`outm`" option which allows to specify which output we want ("`none`" for none, "`each`", the default, for a series of statistics for each variates, or, as in the example, a vector of functions 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 is as following: the recycling will be done from `nvariates=1` to `nvariates=n` with $n > 1$. This allows to use the multivariates nodes as a third dimension, in case you want to test various alternatives.

Assume as in section 2.2.5 that the distribution reflecting the uncertainty on `conc` was not sure, 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 options; ii) transfer these options until 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, 0.6))
> 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 = 5e-04, max = 0.0015)
> risk <- 1 - (1 - r)^dose
> EC5 <- mc(conc, cook, serving, expo, dose, r, risk)
> summary(EC5)

conc :
[[1]]
       NoVar
median  9.96
mean    9.86
2.5%    6.12
97.5%  13.65

[[2]]
       NoVar
```

```
median  9.95
mean    9.92
2.5%    8.08
97.5%  11.82


cook :
       mean    sd  Min 2.5%  25%  50% 75% 97.5% Max  nsv Na's
NoUnc 0.122 0.182 0.02 0.02 0.02 0.02 0.2     1   1 1000    0

serving :
      mean   sd  Min 2.5%  25%  50%  75% 97.5% Max  nsv Na's
NoUnc 48.8 25.9 5.88 13.2 29.5 44.3 61.9   112 169 1000    0

expo :
[[1]]
       mean    sd  Min 2.5%   25%   50%   75% 97.5%  Max  nsv Na's
median 59.3 100.7 1.17 3.16  7.95 15.00  83.0   312 1000 1000    0
mean   58.7  99.7 1.16 3.13  7.87 14.85  82.1   309  990 1000    0
2.5%   36.5  61.9 0.72 1.94  4.89  9.22  51.0   192  615 1000    0
97.5%  81.3 138.0 1.60 4.33 10.90 20.56 113.7   428 1370 1000    0

[[2]]
       mean    sd  Min 2.5%  25%  50%  75% 97.5%  Max  nsv Na's
median 59.3 100.7 1.17 3.16 7.95 15.0 82.9   312  999 1000    0
mean   59.1 100.4 1.17 3.15 7.92 14.9 82.7   311  996 1000    0
2.5%   48.1  81.7 0.95 2.56 6.45 12.2 67.3   253  811 1000    0
97.5%  70.4 119.5 1.39 3.75 9.44 17.8 98.5   370 1187 1000    0


dose :
[[1]]
       mean    sd  Min 2.5%   25%  50%   75% 97.5%  Max  nsv Na's
median 59.4 101.4 0.00 2.00  8.00 16.0  82.0   312  998 1000    0
mean   58.7 100.1 0.04 1.88  7.61 15.8  81.1   314  990 1000    0
2.5%   36.3  62.1 0.00 1.00  5.00 10.0  49.1   198  633 1000    0
97.5%  81.2 138.0 1.00 3.00 11.00 22.0 110.8   426 1363 1000    0

[[2]]
       mean    sd  Min 2.5%  25%  50%  75% 97.5%  Max  nsv Na's
median 59.3 101.0 0.00 2.00 8.00 16.0 82.2   317 1020 1000    0
mean   59.1 100.7 0.02 1.95 7.63 16.0 81.4   316 1002 1000    0
2.5%   47.8  81.8 0.00 1.00 6.00 13.0 66.1   255  791 1000    0
97.5%  70.7 120.4 0.00 3.00 9.00 19.0 97.0   378 1196 1000    0


r :
          NoVar
median 0.001004
mean   0.001036
2.5%   0.000568
97.5%  0.001435

risk :
```

```
[[1]]
         mean     sd      Min     2.5%     25%    50%    75% 97.5%   Max  nsv
median 0.0546 0.0808 0.000000 0.001936 0.00752 0.0162 0.0811 0.278 0.630 1000
mean   0.0543 0.0796 0.000042 0.001968 0.00787 0.0163 0.0805 0.274 0.622 1000
2.5%   0.0257 0.0408 0.000000 0.000604 0.00357 0.0074 0.0362 0.131 0.358 1000
97.5%  0.0854 0.1175 0.000858 0.003666 0.01292 0.0263 0.1325 0.413 0.812 1000
       Na's
median    0
mean      0
2.5%      0
97.5%     0

[[2]]
         mean     sd      Min     2.5%     25%     50%    75% 97.5%   Max  nsv
median 0.0538 0.0796 0.00e+00 0.001948 0.00763 0.01598 0.0795 0.272 0.638 1000
mean   0.0544 0.0799 2.37e-05 0.001999 0.00783 0.01639 0.0803 0.276 0.630 1000
2.5%   0.0308 0.0483 0.00e+00 0.000896 0.00411 0.00882 0.0439 0.159 0.429 1000
97.5%  0.0802 0.1120 0.00e+00 0.003280 0.01230 0.02501 0.1220 0.394 0.801 1000
       Na's
median    0
mean      0
2.5%      0
97.5%     0
```

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

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

The recycling rules in `mc2d` also allows to use the 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 $conc \sim N(14, 2)$. We can *i*) build a "bivariate" node reflecting these two concentrations[8] ; *ii*) transfer these options until 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, 0.6))
> 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 = 5e-04, max = 0.0015)
> risk <- 1 - (1 - r)^dosetot
> EC6 <- mc(conc, cook, serving, expo, dose, dosetot, r, risk)
> summary(EC6)
```

[8]Note that we could simulate a correlation between both contaminants using a multinormal distribution.

```
conc :
[[1]]
       NoVar
median  9.79
mean    9.77
2.5%    5.96
97.5%  14.83

[[2]]
       NoVar
median  14.0
mean    14.1
2.5%    10.8
97.5%   18.3


cook :
       mean    sd  Min 2.5%  25%  50% 75% 97.5% Max  nsv Na's
NoUnc 0.112 0.169 0.02 0.02 0.02 0.02 0.2     1   1 1000    0

serving :
      mean   sd  Min 2.5%  25%  50%  75% 97.5% Max  nsv Na's
NoUnc   49 24.7 5.58 13.3 30.9 45.5 61.8   108 171 1000    0

expo :
[[1]]
       mean    sd   Min 2.5%   25%  50%   75% 97.5%  Max  nsv Na's
median 55.6  94.2 1.092 2.74  7.76 13.1  77.4   258 1031 1000    0
mean   55.5  93.9 1.090 2.74  7.75 13.1  77.2   257 1028 1000    0
2.5%   33.8  57.3 0.665 1.67  4.72  8.0  47.1   157  627 1000    0
97.5%  84.2 142.6 1.654 4.16 11.76 19.9 117.2   390 1561 1000    0

[[2]]
        mean  sd  Min 2.5%   25%  50%   75% 97.5%  Max  nsv Na's
median  79.4 134 1.56 3.92 11.08 18.8 110.5   368 1471 1000    0
mean    80.0 135 1.57 3.95 11.17 18.9 111.4   371 1483 1000    0
2.5%    61.6 104 1.21 3.04  8.59 14.5  85.7   285 1141 1000    0
97.5%  103.9 176 2.04 5.13 14.51 24.6 144.6   482 1926 1000    0


dose :
[[1]]
       mean    sd   Min 2.5%   25%  50%   75% 97.5%  Max  nsv Na's
median 55.4  94.1 0.000 2.00  7.00 14.0  78.5   262 1038 1000    0
mean   55.5  94.2 0.030 1.76  7.29 14.2  78.5   263 1029 1000    0
2.5%   33.7  57.3 0.000 1.00  4.00  9.0  48.0   158  614 1000    0
97.5%  84.0 142.5 0.525 3.00 11.00 21.0 118.0   400 1539 1000    0

[[2]]
        mean  sd  Min 2.5%  25%  50% 75% 97.5%  Max  nsv Na's
median  79.5 135 0.00 3.00 10.5 20.0 112   372 1478 1000    0
mean    80.1 136 0.23 2.88 10.7 20.0 113   377 1484 1000    0
2.5%    61.8 105 0.00 2.00  8.0 15.7  86   285 1120 1000    0
97.5%  104.3 177 1.00 4.00 14.0 25.3 146   493 1941 1000    0
```

```
dosetot :
       mean  sd  Min 2.5%  25%  50% 75% 97.5%  Max  nsv Na's
median  136 230 1.00 6.00 18.0 33.2 192   635 2491 1000    0
mean    136 230 1.10 5.69 18.3 33.3 191   634 2514 1000    0
2.5%    107 181 0.00 4.00 15.0 27.0 152   510 1972 1000    0
97.5%   164 279 2.52 7.52 23.0 40.0 234   779 3075 1000    0

r :
          NoVar
median 0.001000
mean   0.000994
2.5%   0.000546
97.5%  0.001452

risk :
         mean    sd     Min    2.5%     25%    50%    75% 97.5%   Max  nsv Na's
median 0.1050 0.138 0.00104 0.00564 0.01750 0.0311 0.1647 0.450 0.909 1000    0
mean   0.1076 0.139 0.00110 0.00563 0.01806 0.0326 0.1721 0.459 0.894 1000    0
2.5%   0.0633 0.091 0.00000 0.00292 0.00936 0.0171 0.0954 0.289 0.729 1000    0
97.5%  0.1582 0.188 0.00293 0.00893 0.02943 0.0514 0.2613 0.634 0.981 1000    0
```

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

4 Another Example: A QRA of Waterborne Cryptosporidiosis in France

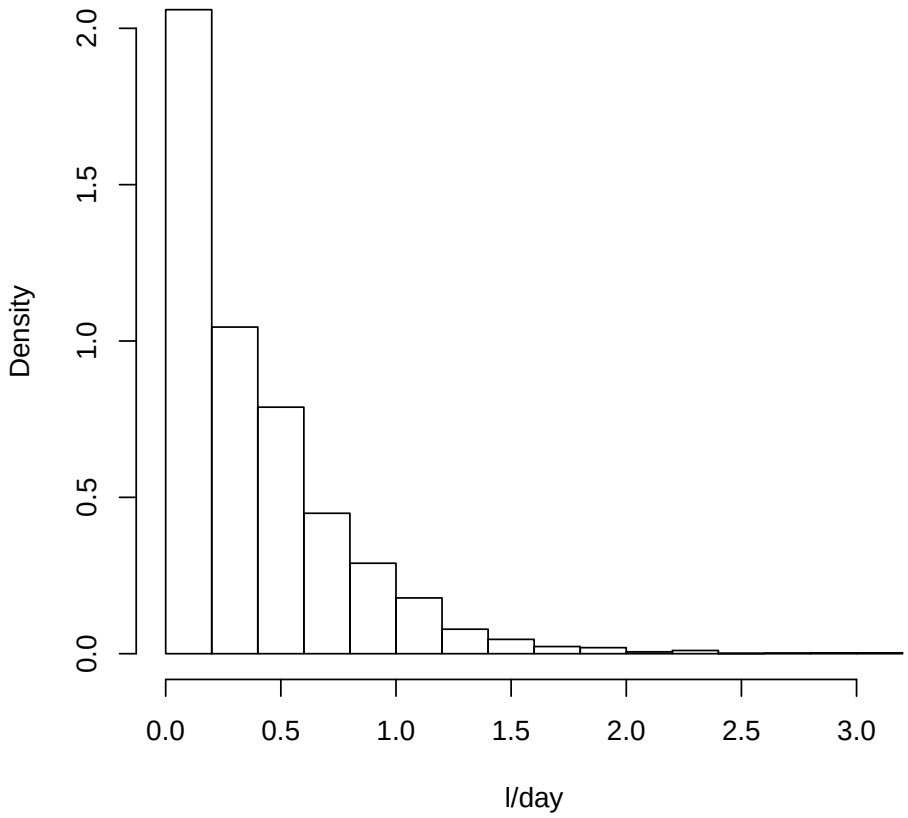
This example is adapted from [4]. The aim is to evaluate the risk of infection with *Cryptosporidium parvum* from consumption of tap-water, given that n oocysts /100 l. have been observed in a storage reservoir.

4.1 Tap Water Consumption Model

We own row data of daily cunsomption of tap water from 1,180 tap water consumers (var `inca`, see Figure 6). We could choose to use this empirical distribution:

```
> ndvar(1001)

[1] 1001

> ndunc(1001)

[1] 1001

> 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 0.41   0.36   3   0    V each
```

Figure 6: Histogram of daily tap water intake



but we will use the "fitdistrplus" library. inca includes a lot of 0, corresponding to days when individuals do not drink tap water (possibly bottled water). We could try a mixture of distributions, with 0 and non-0.

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

summary statistics
------
min:  0.0221   max:  3.2
median:  0.48
mean:  0.566
sample sd:  0.385
sample skewness:  1.75
sample kurtosis:  7.98
```

Following the descdist function (See figure 7), 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
------
GOODNESS-OF-FIT STATISTICS

_____________ Chi-squared_____________
Chi-squared statistic:  3081
Degree of freedom of the Chi-squared distribution:  23
Chi-squared p-value:  0

!!! For continuous distributions, Kolmogorov-Smirnov and
      Anderson-Darling statistics should be prefered !!!

_____________ Kolmogorov-Smirnov_____________
Kolmogorov-Smirnov statistic:  0.0643
Kolmogorov-Smirnov test:  rejected
!!! The result of this test may be too conservative as it
     assumes that the distribution parameters are known !!!

_____________ Anderson-Darling_____________
Anderson-Darling statistic:  18.8
Anderson-Darling test:  rejected

> plot(Adj_water)
```

Figure 7: Graph from the `descdist` function.

```
summary statistics
------
min:  0.0221   max:  3.2
median:  0.48
mean:  0.566
sample sd:  0.385
sample skewness:  1.75
sample kurtosis:  7.98
```

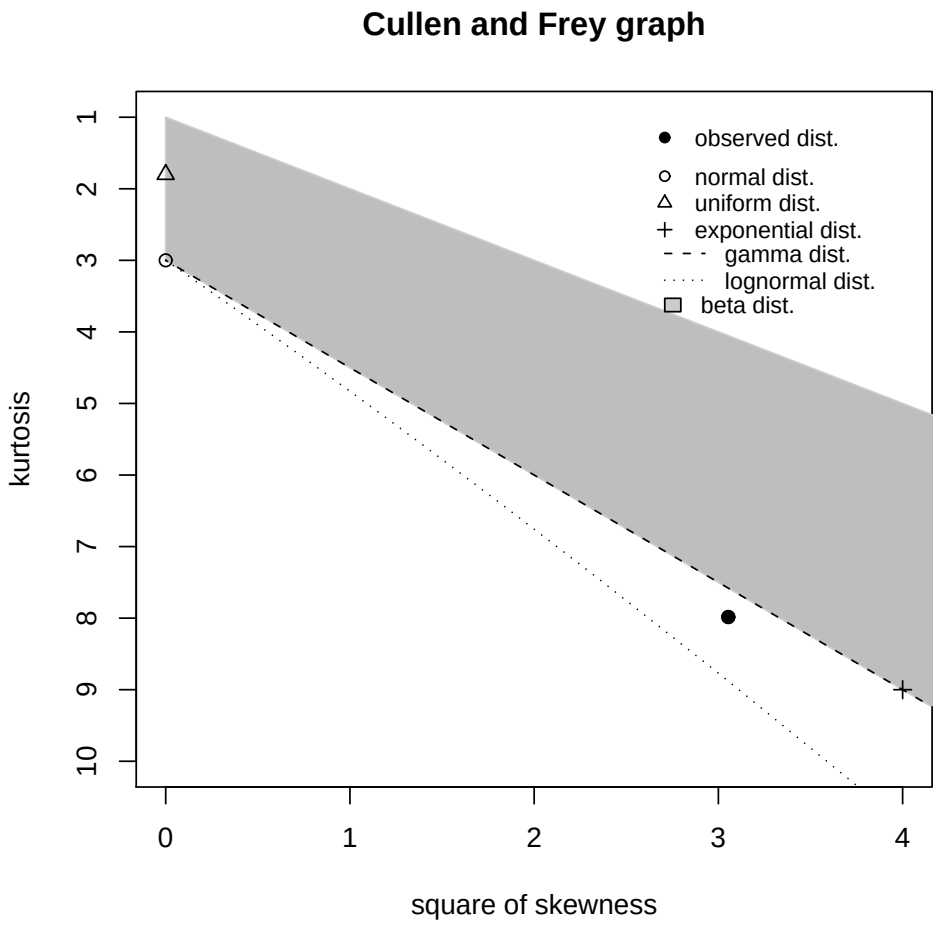
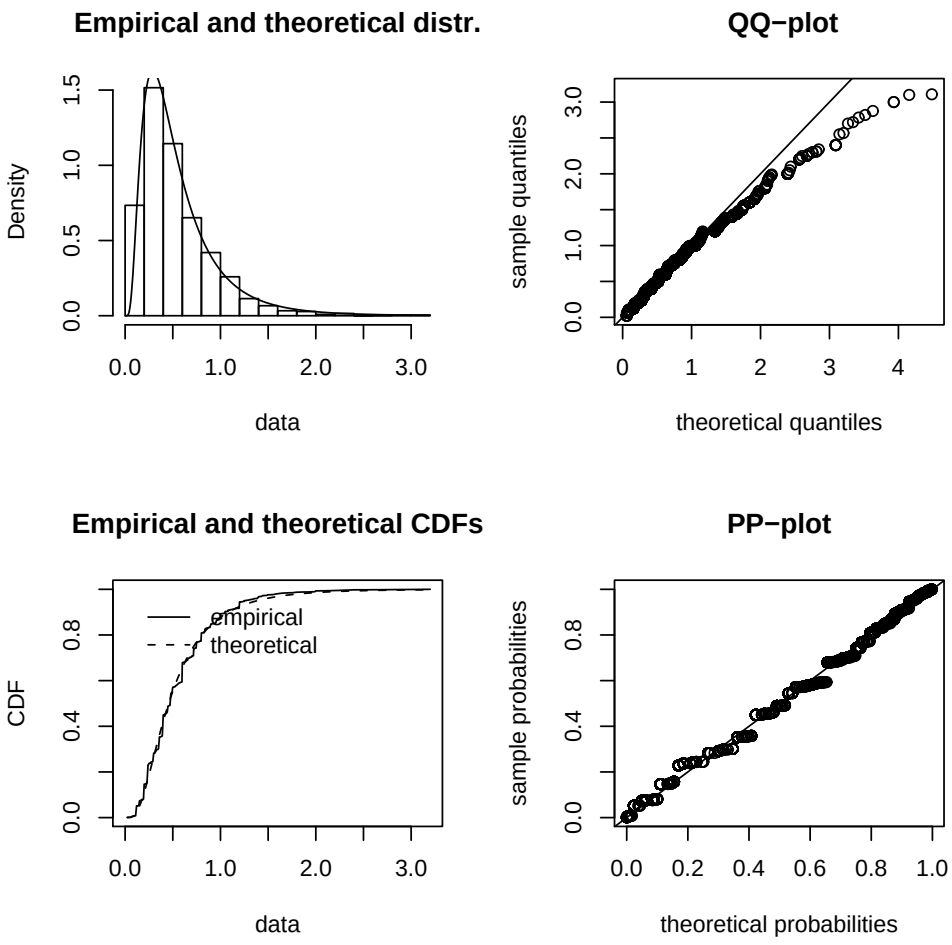


Figure 8: Graph from the `descdist` function.



Not so bad (See Figure 8), and better than 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` package, using something like.

```
> Boot <- bootdist(Ajust_lnorm, bootmethod = "param", niter = ndunc())
> Mean_conso <- mcdata(Boot$bootestim$meanlog, type = "U")
> Sd_conso <- mcdata(Boot$bootestim$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 thus use:

```
> conso0 <- mcdata(0, type = "V")
> conso1 <- mcstoc(rlnorm, type = "V", meanlog = meanlog, sdlog = sdlog)
> mcswitch <- mcstoc(rbern, type = "V", prob = pnonzero)
> v <- mcprobtree(mcswitch, 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 0.418 0.496   0    0   0 0.31 0.624  1.64 7.08 1001    0
```

4.2 The Dose-Response Model

We propose a boostrap from data (`datDR`) issued from [1]. We first define a function "`DR`" with a `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, 1e+05,
+     1e+06), pi = c(2, 4, 2, 5, 2, 3, 1, 1), ni = c(5, 8, 3, 6,
+     2, 3, 1, 1))
> estDR <- function(pos, ref) {
+     -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 0.00532
mean   0.00571
2.5%   0.00296
97.5%  0.01031
```

4.3 The Model

Deriving the final model is direct. We build 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 $n = 2$ 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 0.558 0.787   0    0   0 0.3411 0.789  2.39 12.13 1001    0
mean   0.883 1.244   0    0   0 0.5396 1.248  3.79 19.15 1001    0
2.5%   0.142 0.200   0    0   0 0.0868 0.201  0.61  3.09 1001    0
97.5%  3.349 4.714   0    0   0 2.0463 4.732 14.36 72.54 1001    0
```

To be compared to the results obtained in the Table 2 in [4].

Improvement: the results for $n = \{0, 1, 2, 5, 10, 20, 50, 100, 1000\}$ 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 build or study their models, and that it may help developing the use of "Two-dimensional" simulation. Nevertheless, "`mc2d`" is currently under development:

CHECK CAREFULLY YOUR MODEL AND RESULTS TO TRACK THE BUGS

and, if you would like to improve it

JOIN US AT `http://riskassessment.r-forge.r-project.org/`

Please refer any commentary or bugs to `rpouillot@yahoo.fr`.

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):232–6., 1996.

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

[3] 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.

[4] 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.

[5] 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.

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