

The `mc2d` package

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

Contents

1	Introduction	2
1.1	What is <code>mc2d</code> ?	2
1.2	What is Two-Dimensional Monte-Carlo Simulation (briefly)?	2
1.3	A basic example	4
2	Basic Principles and Functions	7
2.1	Preliminary Step	7
2.2	The <code>mcnode</code> Object	7
2.3	The <code>mc</code> Object	12
2.4	Analysing an <code>mc</code> Object	14
2.5	Other Functions and <code>mc</code> Objects	19
3	Multivariate Nodes	19
3.1	Multivariate Nodes for Multivariate Distributions	19
3.2	Multivariate Nodes as a Third Dimension for Multiple Options	21
3.3	Multivariate Nodes as a Third Dimension for Multiple Contaminants	22
4	Another Example: A QRA of Waterborne Cryptosporidiosis in France	23
4.1	Tap Water Consumption Model	23
4.2	The Dose-Response Model	26
4.3	The Model	26
	As a Conclusion	27

This documentation is intended for readers with:

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

The definitive reference for all function arguments remains the package documentation. It is recommended to try the examples while reading.

1 Introduction

1.1 What is mc2d?

mc2d means Two-Dimensional Monte-Carlo (*Monte-Carlo à 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, distribution-fitting tools were included; they have since been moved to the separate package `fitdistrplus` [3].

mc2d was built for QRA in the Food Safety domain but 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]. The principal reference remains [2].

A QRA should reflect the **variability** in the risk and take into account the **uncertainty** associated with the risk estimate. Variability represents temporal, geographical and/or individual heterogeneity of the risk across the population. Uncertainty stems from a lack of perfect knowledge about the QRA model structure and parameters.¹

A two-dimensional Monte-Carlo simulation separates variability and uncertainty by sampling their respective distributions independently [2]. The method proceeds as follows (see Figure 1):

1. Parameters are divided into four categories: *fixed*, *variable* (variability only), *uncertain* (uncertainty only), and *variable and uncertain*. For the last category, a hierarchical structure is required, e.g.:

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

where the normal distribution represents variability in r conditional on uncertain parameters a and b , with hyper-distributions such as $a \sim \text{Unif}(l_a, u_a)$ and $b \sim \text{Unif}(l_b, u_b)$.

2. A set of uncertain parameters is randomly sampled from their distributions.
3. The model is evaluated with a one-dimensional Monte-Carlo simulation of size N_v , treating uncertain parameters as fixed. Statistics (mean, SD, percentiles) are computed and stored.
4. Steps 2-3 are repeated N_u times.
5. The 50th percentile (median) of each statistic is the point estimate; the 2.5th and 97.5th percentiles constitute the 95% credible interval.

More formally, in its simplest version the framework is a chain of three random variables:

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

with marginal distribution $[p]$ and conditional distributions $[\pi \mid p]$ and $[Y \mid \pi]$, under the assumption $[p, \pi, Y] = [p][\pi \mid p][Y \mid \pi]$.

- Y is the variable of interest.
- π is the parameter governing the distribution of Y ; its uncertainty is characterised through $[\pi \mid p]$.
- p is the parameter governing the uncertainty of π ; its distribution $[p]$ is assumed known.

A two-dimensional Monte-Carlo simulation provides a bundle of N_u distributions $[Y \mid \pi = \pi_i]$ where $\pi_i, i = \{1, \dots, N_u\}$ are drawn from $[\pi \mid p]$.

mc2d uses arrays of (at least) two dimensions: the first dimension reflects variability, the second reflects uncertainty.

¹In the engineering risk community, these are called *aleatoric uncertainty* for variability and *epistemic uncertainty* for uncertainty.

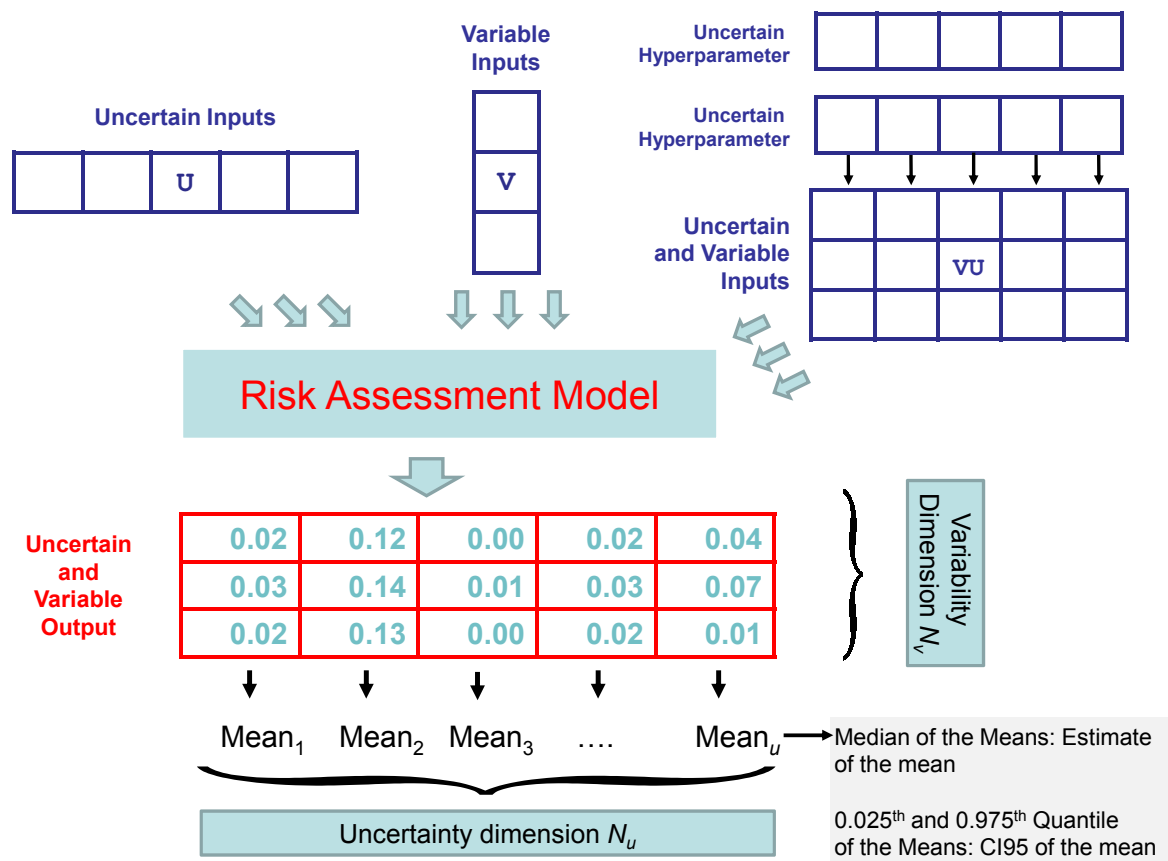


Figure 1: Schematic Representation of a Two-Dimensional Monte-Carlo Simulation.

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; results should not be interpreted.

The model assumes:

- *E. coli* O157:H7 are randomly distributed in a batch with mean concentration $c = 10$ CFU/g.
- No bacterial growth occurs (product is frozen until cooked just before consumption).
- 2.7% of consumers cook their beef rare, 37.3% medium, and 60.0% well done.
- Bacterial inactivation by cooking: no inactivation (rare), 1/5 surviving (medium), 1/50 surviving (well done).
- Serving size s for <3 year children follows a Gamma distribution: $shape = 3.93$, $rate = 0.0806$.
- The dose-response is a one-hit model with probability of illness per hit $r = 0.001$.

What is the risk of illness in the population that consumed the contaminated lot?

1.3.1 One-Dimensional Monte-Carlo Simulation

All distributions represent variability only:

$$c = 10, \quad i \sim \text{emp}(\{1, 1/5, 1/50\}, \{0.027, 0.373, 0.600\}), \quad s \sim \text{Gamma}(3.93, 0.0806)$$

$$n \sim \text{Poisson}(c \times i \times s), \quad r = 0.001, \quad P = 1 - (1 - r)^n$$

```
library(mc2d)
```

```
## Loading required package: mvtnorm
```

```
##
```

```
## Attaching package: 'mc2d'
```

```
## The following objects are masked from 'package:base':
```

```
##
```

```
##      pmax, pmin
```

```
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 shows for each node: the variable mode, `nsv` (variability dimension size), `nsu` (uncertainty dimension size), `nva` (number of variates), basic statistics (min, mean, median, max), number of missing values (Nas), the node type (0 = fixed, V = variable, U = uncertain, VU = variable and uncertain), and the output level outm.

This one-dimensional simulation gives a mean risk of approximately 5%, with 2.5% of the population having a risk greater than 20.3%.

1.3.2 Two-Dimensional Monte-Carlo Simulation

Adding uncertainty: the mean concentration c follows $N(10, 2)$ and the parameter r follows $\text{Unif}(0.0005, 0.0015)$:

$$c \sim N(10, 2), \quad i, s \text{ as before}, \quad n \sim \text{Poisson}(c \times i \times s), \quad r \sim \text{Unif}(0.0005, 0.0015), \quad P = 1 - (1 - r)^n$$

```
ndunc(101)
```

```
## [1] 101
```

```
conc    <- mcstoc(rnorm,      type = "U", mean = 10, sd = 2)
cook    <- mcstoc(rempricalD, 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
```

The `type` argument indicates whether a distribution represents variability ("V", default), uncertainty ("U"), or both ("VU"). The median of the 101 simulations gives a best estimate of 0.0457 with a 95% credible interval of [0.0238, 0.0794].

2 Basic Principles and Functions

A typical `mc2d` session:

1. Choose an empirical or parametric distribution for each parameter (`fitdistrplus` [3] is a convenient fitting tool).
2. Construct an `mcnode` object for each parameter (`mcdata`, `mcstoc`).
3. Group `mcnode` objects into an `mc` object (`mc`).
4. Study the `mc` object via summaries, graphs, and sensitivity analysis (`summary.mc`, `plot.mc`, `tornado`, `tornadounc`).

2.1 Preliminary Step

Load `mc2d` at the start of your R session with `library(mc2d)`. Set the default Monte-Carlo dimensions with `ndvar()` (variability) and `ndunc()` (uncertainty).

2.2 The `mcnode` Object

2.2.1 `mcnode` Object Structure

An `mcnode` is the basic element of an `mc` object — it is associated with one variable, while an `mc` is a set of associated variables.

An `mcnode` is an array of dimension $(nsv \times nsu \times nvariables)$.² Four types exist:

- **V `mcnode`** (Variability): dimension $(nsv \times 1 \times nvariables)$.
- **U `mcnode`** (Uncertainty): dimension $(1 \times nsu \times nvariables)$.
- **VU `mcnode`** (Variability and Uncertainty): dimension $(nsv \times nsu \times nvariables)$.
- **0 `mcnode`** (Neither): dimension $(1 \times 1 \times nvariables)$. Not needed in the univariate context but useful for multivariate nodes (Section 3).

Ways to construct an `mcnode`:

1. `mcstoc` — from random number generating functions.

²In this section we only consider univariate nodes with $nvariables = 1$.

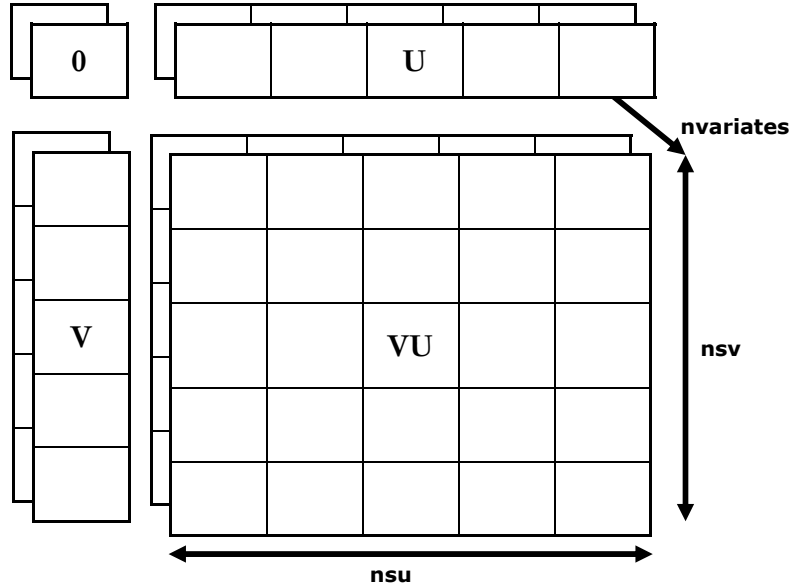


Figure 2: Structure of the various ‘mcnode’ objects.

2. `mcdata` — from data.
3. Direct operations on `mcnode` objects.
4. `mcprobtrees` — from a mixture of distributions (probability tree).
5. Functions such as `==`, `>`, `is.na`, `is.finite` applied to an existing `mcnode`.

2.2.2 The `mcstoc` function

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

- `func`: random-data function or its name as a character. Table 1 lists available distributions.
- `type`: type of `mcnode` to build. Default: "V".
- `...`: arguments passed to `func` (except the sample size). *All must be named.*
- `nsv`, `nsu`: number of samples in variability/uncertainty dimensions.
- `nvariables`: number of variates (Section 3).
- `outm`: default output for multivariate nodes (Section 3).
- `nsample`: name of the sample-size argument of `func` (usually "n"; use "nn" for `rhyper` and `rwilcox`).
- `seed`: random seed. If `NULL`, seed is unchanged.
- `rtrunc`: truncate the distribution between `linf` and `lsup`.
- `lhs`: use Latin Hypercube Sampling.

In our example, `mcstoc` specified `conc` (normal), `cook` (empirical discrete), `serving` (gamma), and `dose` (Poisson with `mcnode` argument `lambda`).

Table 1: Available distributions

Package	Distribution	Function	Size arg.	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
	neg. 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
	emp. discrete	‘rempiricalD‘	‘n‘	‘values, prob‘	Y	Y
	emp. cont.	‘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
	gen. beta	‘rbetagen‘	‘n‘	‘shape1,shape2,min,max,ncp‘	Y	Y
	multinomial	‘rmultinomial‘	‘n‘	‘size, prob‘	N	N
	Dirichlet	‘rdirichlet‘	‘n‘	‘alpha‘	N	N
	mv. normal	‘rmultinormal‘	‘n‘	‘mean, sigma‘	N	N
	beta subjective	‘rbetasubj‘	‘n‘	‘min, mode, mean, max‘	Y	Y
	min. info.	‘rmqi‘	‘n‘	‘mqi, mqi.quantile, ...‘	Y	Y

```

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

```

A normal distribution $N(2, 3)$ truncated on $[1.5, 2]$ with Latin Hypercube Sampling:³

```

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

```

Additional distributions in mc2d: `rbern` (Bernoulli), `rempiricalD` (empirical discrete), `rpert` [8], `rtriang`, `rdirichlet`, `rmultinomial`, `rmqi` (min. quantile info.). The multinomial distribution is vectorized as `rmultinomial` (use instead of `stats::rmultinom`).

2.2.3 The mcdata function

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

See the function documentation for accepted data sizes and types. Example — placing TRUE in a "U" node for half 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` objects support arithmetic operations with coherent type propagation (Table 2).⁴

- $0 + 0 = 0$; $0 + V = V$; $0 + U = U$; $0 + VU = VU$
- $V + V = V$; $V + U = VU^5$; $V + VU = VU^6$
- $U + U = U$; $U + VU = VU^7$; $VU + VU = VU$

³The mean and SD of the non-truncated distribution are not preserved after truncation.

⁴These rules differ from standard R recycling rules.

⁵U is recycled by row; V classically by column.

⁶V recycled by column.

⁷U recycled by row.

Table 2: ‘mcnode’ combination types

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

```
expo <- conc * cook * serving # U * V * V → VU
risk <- 1 - (1 - r)^dose      # U * VU → VU
```

2.2.5 The mcprobtrees function

mcprobtrees builds an mcnode as a mixture of distributions. If the microbiologists are 75% sure that $conc \sim N(10, 2)$ and 25% sure that $conc \sim U(8, 12)$:⁸

```
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 <- mcprobtrees(whichdist, list("0" = conc1, "1" = conc2), type = "U")
```

mcprobtrees can also generate samples from a mixture for variability.

2.2.6 Other functions for constructing an mcnode

Comparison operators (`==`, `<`, `<=`, `>=`, `>`) generate an mcnode when applied to one. Special functions `is.na`, `is.nan`, `is.finite`, `is.infinite` are also implemented.

```
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
```

```
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
```

⁸Alternatives for whichdist: `mcstoc(rempricalD, type="U", values=c(0,1), prob=c(75,25))` or `mcstoc(rbern, type="U", prob=0.25)`.

2.2.7 Specifying a correlation between mcnodes

A Spearman rank correlation structure between 2 or more nodes can be specified with `cornode`, which uses the Iman & Conover method [4].⁹

```
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
```

Correlations can be specified between V, U, or VU nodes, or between one V node and multiple VU nodes. A multivariate normal distribution (`rmultinormal`) is another way to specify correlations assuming normality.

2.3 The mc Object

Once `mcnode` objects are constructed, group them into an `mc` object for analysis. An `mc` is a list of `mcnodes`. It can be constructed with `mc()`, `evalmcmmod()`, or within `evalmccut()`.

2.3.1 The mc function

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

... are `mcnode` or `mc` objects. In our example:

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

2.3.2 The mcmmodel and evalmcmmod functions

`mcmmodel` wraps a model expression for later evaluation with `evalmcmmod`. Use this once the model is validated, to re-run it with different dimensions or seeds in one line.

```
modelEC3 <- mcmmodel({
  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)
```

⁹The resulting correlation (~0.4) is an approximation because a discrete distribution (`cook`: 3 categories) is correlated with a continuous distribution (`serving`).

```

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"

```

Notes:

- The model is wrapped between { and }.
- Any valid R code is allowed inside.¹⁰
- The model must end with an mc() call.

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

```

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

```

2.3.3 The mcmodelcut and evalmccut functions

For high-dimensional models that exceed R's memory limit, `evalmccut` evaluates the model in a loop over the uncertainty dimension, computing and storing statistics at each step.¹¹

```

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 → mc object

```

¹⁰The simulation dimensions can be referenced via `ndvar()` and `ndunc()` if needed.

¹¹Using a `tornado` inside the model should be avoided as it slows `evalmccut` considerably.

```

{
  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: statistics
{
  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

2.4.1 The summary function

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

Quantiles in `probs` are evaluated in the variability dimension; then the median and `lim` quantiles are evaluated across those 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

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

Provides histograms of each `mcnode` in the `mc` object. Note: variability and uncertainty are collapsed, so the histogram may be approximate.

```
hist(EC2)
```

From `mc2d` version 0.2-0, `gghist` provides a `ggplot2`-based histogram, returning a `ggplot2` object that can be further customised with `+` operators.

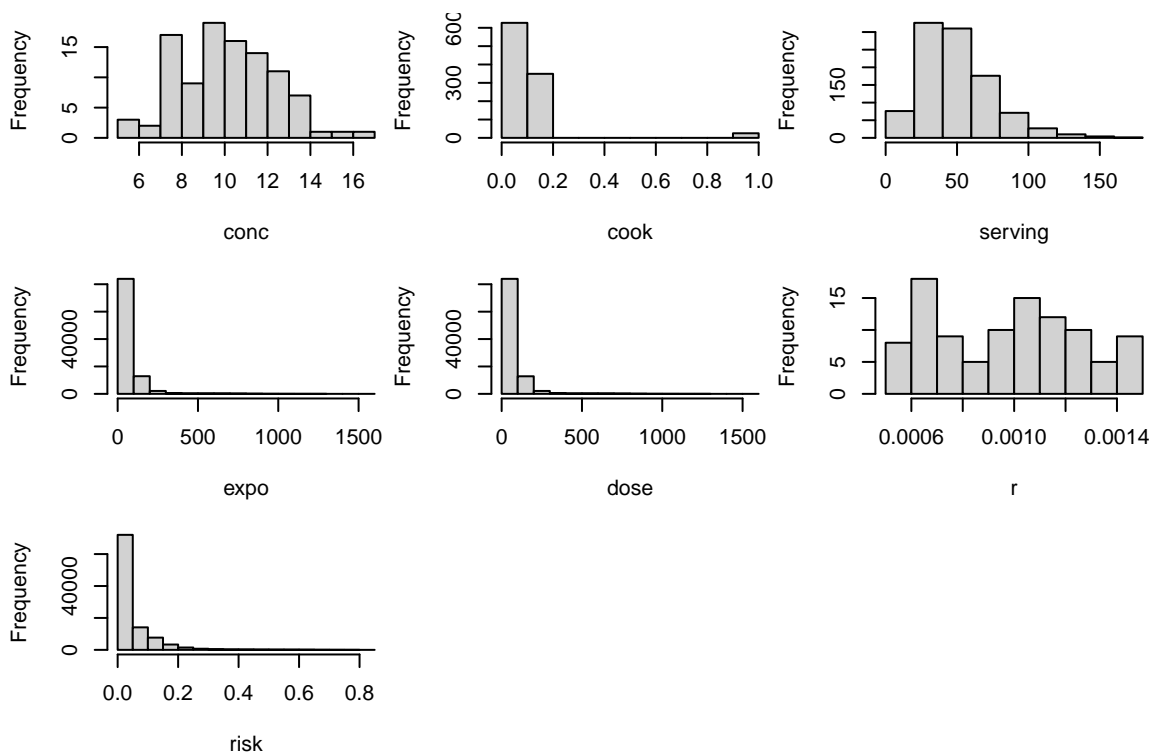


Figure 3: The `hist` function.

2.4.3 The `plot` function

```
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, ...)
```

Plots the empirical CDF with uncertainty envelope. The 0.25 and 0.75 quantiles (default `lim`) form the inner envelope.

```
plot(EC2)
```

From `mc2d` version 0.2-0, `ggplotmc` provides a `ggplot2`-based version.

`spaghetti` draws individual empirical CDF step-functions, one per uncertainty sample, instead of the summary envelope produced by `plot`. The `maxlines` argument caps the number of lines drawn.

```
spaghetti(x, griddim=NULL, xlab=names(x), ylab="F(n)", main="", maxlines=100, ...)
```

```
spaghetti(EC2)
```

`ggspaghetti` provides the same plot via `ggplot2`.

All `mcnode` objects support the same `print`, `summary`, `plot`, and `hist` methods. `ggplotmc`, `gghist`, and `ggspaghetti` on an `mcnode` allow post-processing of the graph.

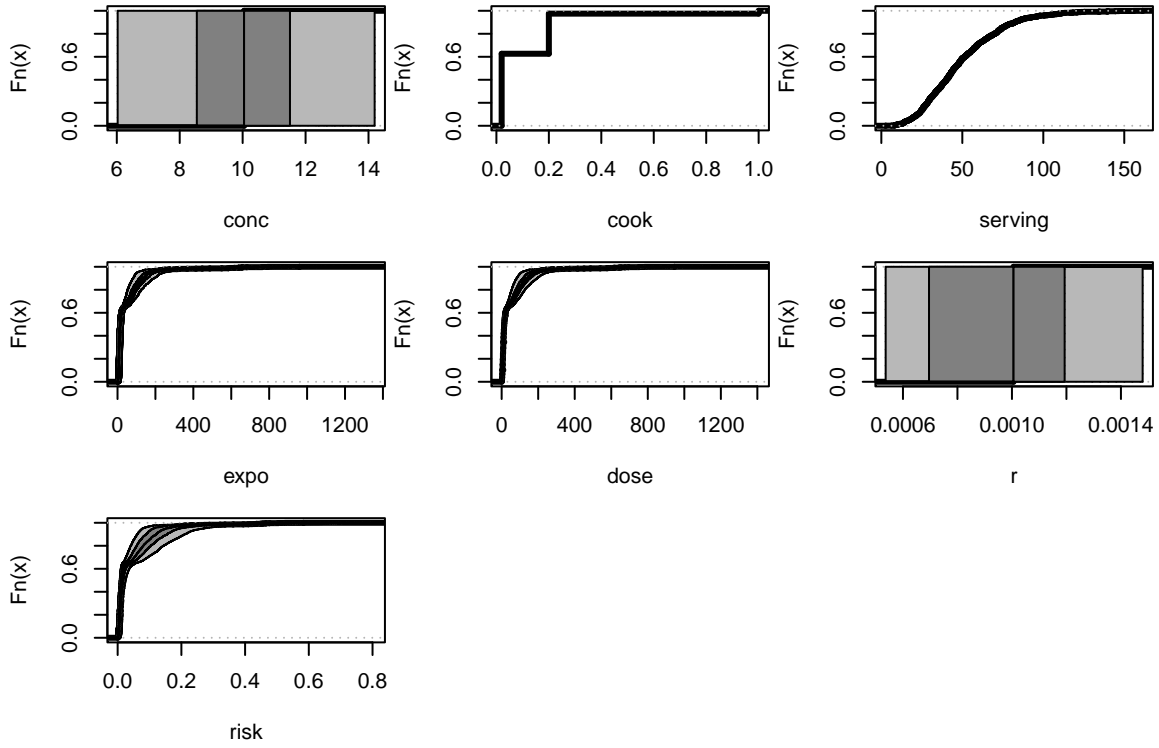


Figure 4: The plot function.

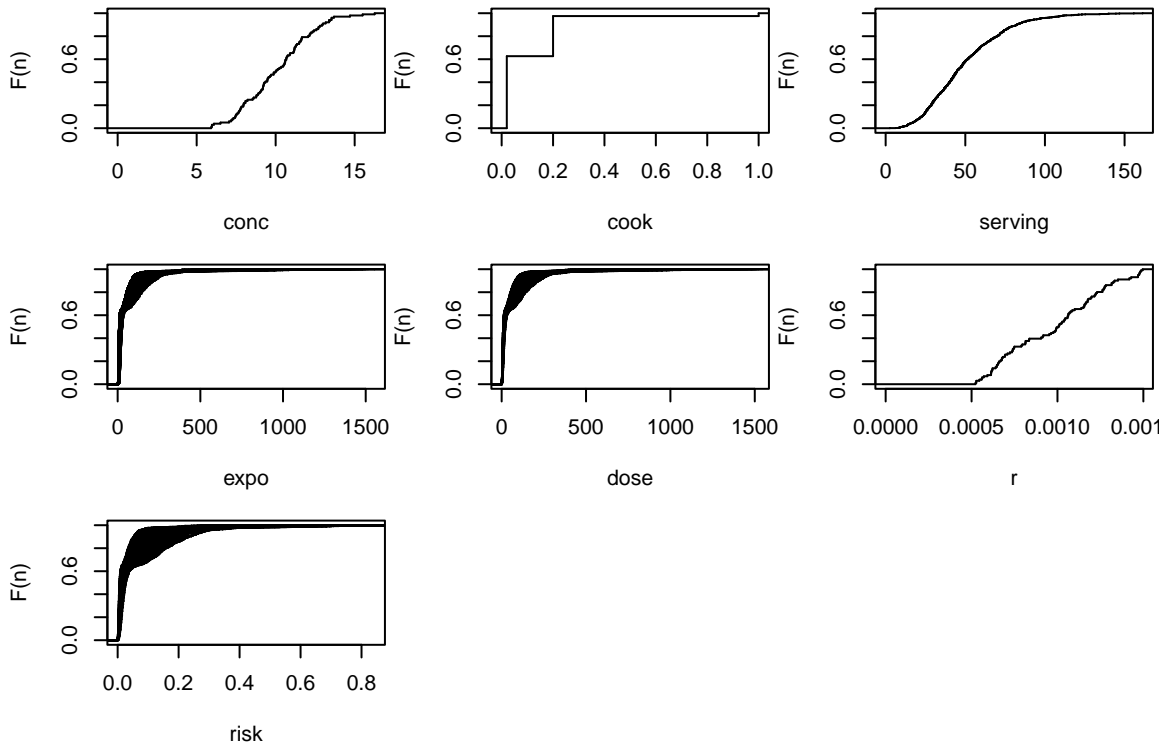


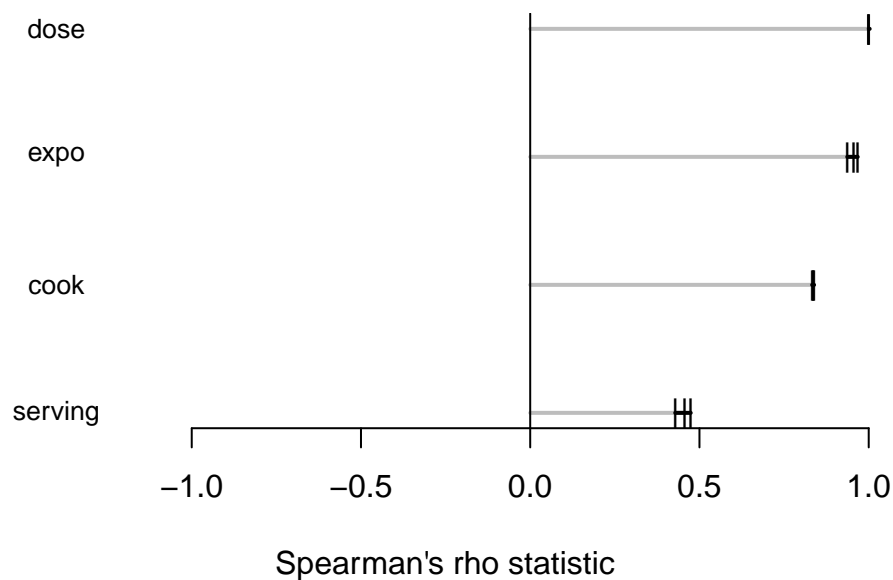
Figure 5: The spaghetti function.

2.4.4 The tornado function

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

Computes Spearman (default) rank correlation between nodes. `output` specifies the output node (default: last). `tornado` creates a `tornado` object with a `plot` method.

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



From `mc2d` version 0.2-0, `ggtornado` provides a `ggplot2`-based version.

2.4.5 The tornadounc function

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

Examines the impact of uncertainty on an output estimate. Computes Spearman rank correlation between statistics of `mcnode` objects in the variability dimension. `quant` specifies which quantiles to use.

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

```
## Tornado on uncertainty  
## Spearman's rho statistic  
## Output: risk
```

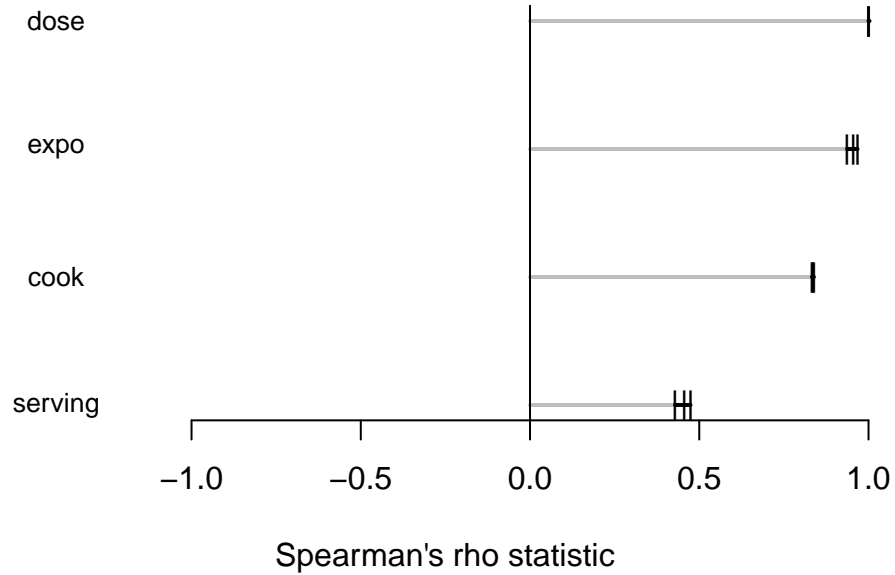


Figure 6: The `plot.tornado` function.

```
## $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 uncertain U nodes and statistics (mean, median, 99th percentile) computed in the variability dimension of VU nodes.

2.4.6 The `mcratio` function

Provides variability, uncertainty, and combined measures [5]. Given:

- **A** = median of uncertainty for the median of variability
- **B** = median of uncertainty for the 97.5th percentile of variability
- **C** = 97.5th percentile of uncertainty for the median of variability
- **D** = 97.5th percentile of uncertainty for the 97.5th percentile of variability

Ratios: Variability = B/A ; Uncertainty = C/A ; Overall Uncertainty = 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. `$` extracts an `mcnode`. `unmc` removes attributes and collapses unit dimensions to return vectors, matrices, or arrays.

```
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 `nvariables` dimension is the third dimension of an `mcnode`. It is mandatory for multivariate distributions, and useful in other situations.

Note that:

```
mcstoc(runif, nvariables = 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
```

does **not** produce a node with 3 variates each having a different lower limit — the vector `c(1,2,3)` is recycled along the variability dimension (standard R behaviour). Use instead:

```
lim <- mcddata(c(1, 2, 3), type = "0", nvariables = 3)
mcstoc(runif, nvariables = 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
```

3.1 Multivariate Nodes for Multivariate Distributions

The primary use of multivariate nodes is for multivariate distributions: Dirichlet, multinomial, multivariate normal, and empirical.

Example 1. Three-member families each buy 500 g of ground beef. The proportions eaten by the baby, older brother, and mother follow a Dirichlet ($\alpha = (2, 3, 5)$) distribution.

```
(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
```

Example 2. Each family member eats a normal distribution of steak (100, 150, 250 g mean) with positive correlation between the children's servings and negative with the mother's.

```
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
```

Both mean and sigma can be variable or uncertain.¹² Example with uncertain mean:

```
m <- mcdata(c(100, 150, 250), type = "O", 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
```

Example 3. Non-parametric bootstrap: 6 individuals eat 100 g, 12 eat 150 g, 6 eat 170 g, and 6 eat 200 g [2].

```
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
```

The outm option controls which statistics to show: "none", "each" (default), or a vector of function names applied across all 30 variates.

3.2 Multivariate Nodes as a Third Dimension for Multiple Options

Assume uncertainty in conc: 75% sure $conc \sim N(10, 2)$, 25% sure $conc \sim U(8, 12)$. Build a bivariate node and propagate both hypotheses simultaneously.

```
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)
```

¹²`rmultinormal` is a vectorized version of `rmvnorm` from `mvtnorm`.

```
## 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
```

(Remember to specify `nvariables = 2` in `mcstoc` for dose — `mc2d` cannot infer it.)

3.3 Multivariate Nodes as a Third Dimension for Multiple Contaminants

Two contaminants with uncertain concentrations $N(10, 2)$ and $N(14, 2)$:¹³

```
mconc <- mcdata(c(10, 14), type = "0", nvariables = 2)
conc  <- mcstoc(rnorm,      type = "U", nvariables = 2, mean = mconc, sd = 2)
cook  <- mcstoc(rempricalD, 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", nvariables = 2, lambda = expo)
dosetot <- mcapply(dose, margin = "variables", 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 :
```

¹³A correlation between contaminants could be specified via `rmultinormal`.

```
## [[1]]
##          NoVar
## 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
```

4 Another Example: A QRA of Waterborne Cryptosporidiosis in France

Adapted from [6]. Goal: evaluate the risk of infection with *Cryptosporidium parvum* from consumption of tap water given n oocysts/100 l observed in a storage reservoir (simplified model).

4.1 Tap Water Consumption Model

We have raw data on daily tap water consumption from 1,180 consumers (*inca*, histogram below). We could use the empirical distribution directly:

```
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
```

Instead, we use `fitdistrplus` [3]. The data include many zeros (days without tap water consumption). We model them as a mixture.

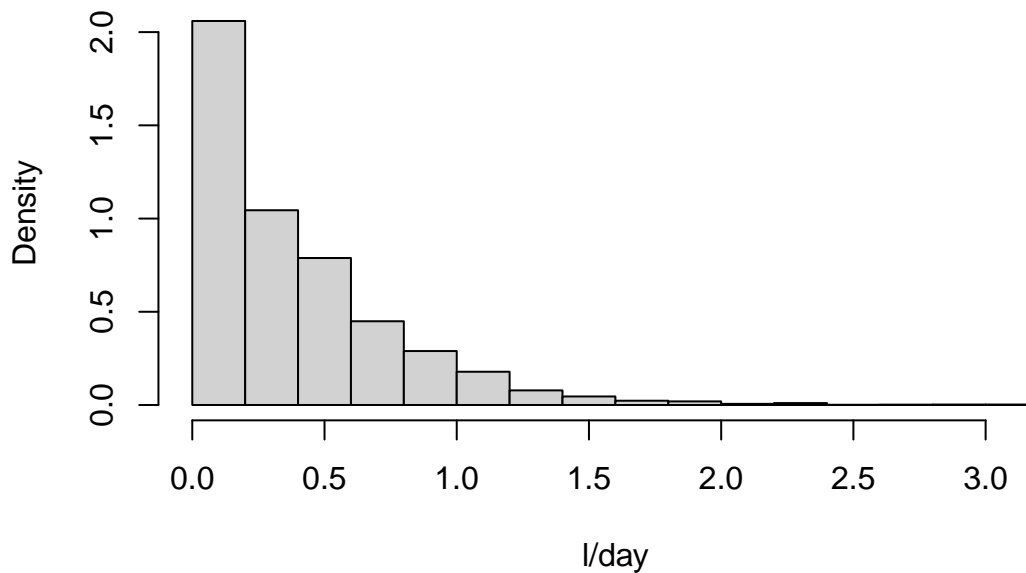


Figure 7: Histogram of daily tap water intake.

```
library(fitdistrplus)

## Loading required package: MASS

## Loading required package: survival

pzero <- 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
## estimated sd: 0.385
## estimated skewness: 1.75
## estimated kurtosis: 7.99
```

From the shape descriptor plot above, we try a lognormal distribution:

Cullen and Frey graph

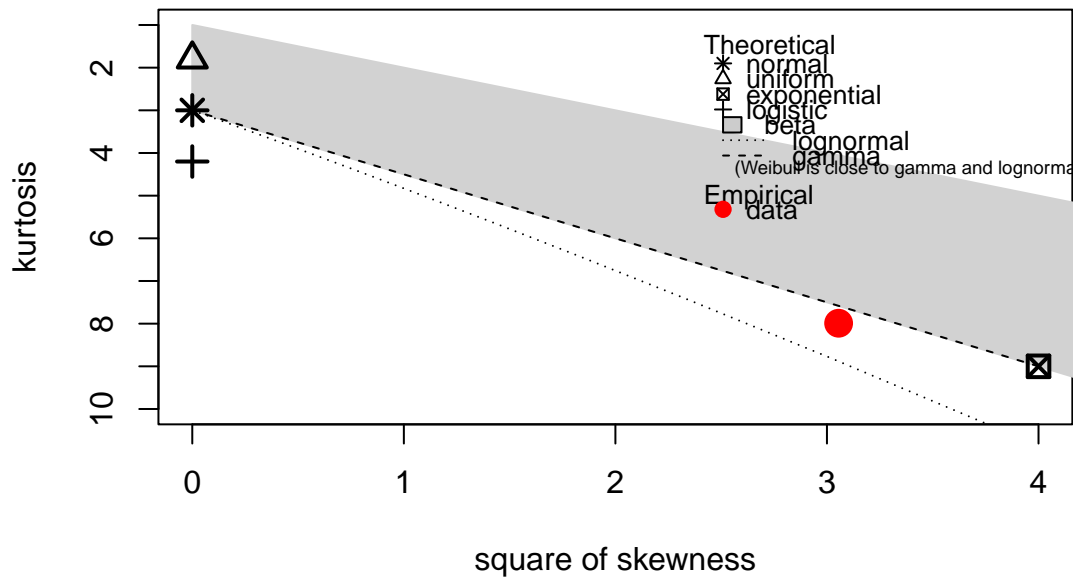


Figure 8: Graph from descdist.

```
Adj_water <- fitdlist(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 -2.23e-12
## sdlog    -2.23e-12  1.00e+00
```

The fit is satisfactory. Uncertainty around the MLE estimates could be incorporated via `bootdist`:

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

For simplicity, we ignore that uncertainty and use `mcprobtrees` to build the mixture:

```

conso0 <- mcdata(0, type = "V")
conso1 <- mcstoc(rlnorm, type = "V", meanlog = meanlog, sdlog = sdlog)
v      <- mcprobtrees(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

Bootstrap from data `datDR` [1]. A function `DR` with argument `n` is defined and passed to `mcstoc`:

```

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

```

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

```

Given $O_o = 2$ oocysts observed in 100 l, the expected number of oocysts in the sample 1:

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

The expected number drunk by individuals and the risk ($\times 10000$):

```
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 can be compared to Table 2 in [6]. 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 = "O", nvariates = 9)
```

As a Conclusion

We hope that mc2d will help risk assessors to construct and analyse their models, and to develop two-dimensional simulations. *Please report any 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):232–236, 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.