Package 'distrSim'

July 22, 2025

Version 2.8.5

Date 2025-01-11

Title Simulation Classes Based on Package 'distr'

Description S4-classes for setting up a coherent framework for simulation within the distr family of packages.

Depends R(>= 3.4), methods, graphics, setRNG(>= 2006.2-1), distr(>= 2.5.2)

Suggests distrEx(>= 2.2)

Enhances distrTEst

Imports startupmsg(>= 1.0.0), stats, grDevices

ByteCompile yes

License LGPL-3

Encoding UTF-8

URL http://distr.r-forge.r-project.org/

LastChangedDate {$LastChangedDate: 2025-01-11 22:16:06 +0100 (Sa, 11 Jan 2025) $}

LastChangedRevision {$LastChangedRevision: 1489 $}

VCS/SVNRevision 1488

NeedsCompilation no

Author Florian Camphausen [ctb] (contributed as student in the initial phase --2005),
Matthias Kohl [aut, cph],
Peter Ruckdeschel [cre, cph],
Thomas Stabla [ctb] (contributed as student in the initial phase --2005)

Maintainer Peter Ruckdeschel <peter.ruckdeschel@uni-oldenburg.de>

Repository CRAN

Date/Publication 2025-01-13 19:00:02 UTC

Contents

`distrSim-package` *distrSim – S4-classes for Simulations Based on Package distr*

Description

distrSim is to provide a conceptual treatment of simulations by means of S4 classes. The package is based on our package **distr**, hence uses distribution classes as introduced there to describe the distributions under which simulations are performed.

Details

Package:	distrSim
Version:	2.8.5
Date:	2025-01-11
Depends:	R(>= 3.4), methods, graphics, setRNG(>= 2006.2-1), distr(>= 2.5.2)
Suggests:	distrEx(>= 2.2)
Imports:	startupmsg, stats, grDevices
LazyLoad:	yes
License:	LGPL-3
URL:	https://distr.r-forge.r-project.org/
VCS/SVNRevision:	1488

Classes

```
"SeqDataFrame"
      slots: data(list[of dataframes with common column structure])
"Dataclass"
      slots: [<name>(<class>)]
      filename(vectororNULL),
      name(character),
      Data(ArrayorNULLorVectororDataframeorSeqDataFrames),
      runs(numeric),
      samplesize(numeric),
      obsDim(numeric)
|>"Simulation"
      (additional) slots: [<name>(<class>)]
      seed(list), distribution(Distribution)
|>"Contsimulation"
      (additional) slots: [<name>(<class>)]
      seed(list), ind(vectororNULL), rate(numeric),
      distribution.id(Distribution),
      distribution.c(Distribution),
      Data.id(vectororNULL),
      Data.c(vectororNULL)
```

Methods

```
savedata                Methods to save the data slot
                       (for Simulation/Contsimulation)
simulate                Methods to fill the data slot
                       (for Simulation/Contsimulation)
plot-methods            Methods for Function plot
                       (for Simulation/Contsimulation)
print-, show-methods    Methods for Function print/show
                       (for Simulation/Contsimulation)
summary-methods         Methods for Function summary
                       (for Simulation/Contsimulation)
```

Functions

cload	loads just the comment slot (for Simulation/Contsimulation)

Slot accessors / -replacement functions

All slots are inspected / modified by corresponding accessors / -replacement functions, e.g. `rate(X)` for an object of class `"Contsimulation"`.

Start-up-Banner

You may suppress the start-up banner/message completely by setting `options("StartupBanner"="off")` somewhere before loading this package by `library` or `require` in your R-code / R-session. If option `"StartupBanner"` is not defined (default) or setting `options("StartupBanner"=NULL)` or `options("StartupBanner"="complete")` the complete start-up banner is displayed. For any other value of option `"StartupBanner"` (i.e., not in `c(NULL,"off","complete")`) only the version information is displayed. The same can be achieved by wrapping the `library` or `require` call into either `suppressStartupMessages()` or `onlytypeStartupMessages(.,atypes="version")`.

Far-reaching Change in Design

From version 1.8 of this package on, we have changed the format how data / simulations are stored: In order to be able to cope with multivariate distributions, regression distributions and (later) time series distributions, we have switched to the common array format: samplesize x obsDim x runs; you may check the version under which an object was generated by `getVersion`; for saved objects from earlier versions, we provide the functions `isOldVersion`, and `conv2NewVersion-methods` to check whether the object was generated by an older version of this package and to convert such an object to the new format, respectively.

Start-up-Banner

You may suppress the start-up banner/message completely by setting `options("StartupBanner"="off")` somewhere before loading this package by `library` or `require` in your R-code / R-session. If option `"StartupBanner"` is not defined (default) or setting `options("StartupBanner"=NULL)` or `options("StartupBanner"="complete")` the complete start-up banner is displayed. For any other value of option `"StartupBanner"` (i.e., not in `c(NULL,"off","complete")`) only the version information is displayed. As for general `packageStartupMessage`'s, you may also suppress all the start-up banner by wrapping the `library` or `require` call into `suppressPackageStartupMessages()` from **startupmsg**-version 0.5 on.

Package versions

Note: The first two numbers of package versions do not necessarily reflect package-individual development, but rather are chosen for the distrXXX family as a whole in order to ease updating "depends" information.

Note

Global options controlling the plots and summaries of Dataclass and Simulation/Contsimulation objects may be inspected / set by `distrSimoptions()` and `getdistrSimOption()`.

Author(s)

Thomas Stabla <statho3@web.de>,
Florian Camphausen <fcampi@gmx.de>,
Peter Ruckdeschel <peter.ruckdeschel@uni-oldenburg.de>,
Matthias Kohl <Matthias.Kohl@stamats.de>
Maintainer: Peter Ruckdeschel <peter.ruckdeschel@uni-oldenburg.de>

References

P. Ruckdeschel, M. Kohl, T. Stabla, F. Camphausen (2006): S4 Classes for Distributions, *R News*, *6*(2), 2-6. https://CRAN.R-project.org/doc/Rnews/Rnews_2006-2.pdf A vignette for packages **distr**, **distrSim**, **distrTEst**, and **distrEx** is included into the mere documentation package **distrDoc** and may be called by `require("distrDoc");vignette("distr")`. A homepage to this package is available under
https://distr.r-forge.r-project.org/

See Also

distr-package setRNG

---

cload *cload*

Description

loads the comment file from a saved `Dataclass` object

Usage

```
cload(filename)
```

Arguments

`filename`	the filename which was declared at the instantiaton of the Dataclass

Details

Uses function `load` to load the comment file from a saved `Dataclass` object.

Value

no value is returned

Author(s)

Thomas Stabla <statho3@web.de>,
Florian Camphausen <fcampi@gmx.de>,
Peter Ruckdeschel <peter.ruckdeschel@uni-oldenburg.de>,
Matthias Kohl <Matthias.Kohl@stamats.de>

See Also

Dataclass-class load savedata-methods

Examples

```
# see Dataclass and Evaluation for examples
## The function is currently defined as
function(filename){
  eval.parent(parse(text=paste("load(\"",filename,".comment\")", sep = "")))
  }
```

Contsimulation-class *Class "Contsimulation"*

Description

In an object of type Contsimulation data can be simulated in any distribution and size. One part (usually the largest) of the random numbers stems from an ideal distribution, the rest from a contaminating distribution.

Objects from the Class

Objects can be created by calls of the form Contsimulation(filename, samplesize, runs, seed, distribution.id,distribution.c, rate) (observation dimension is deduced from slot distribution.id). A Contsimulation-object includes a filename, the number of runs, the size of the sample, the seed, the distribution of the ideal and the contaminated data and the contamination rate. The slot Data stays empty until the method simulate has been used. The same goes for slots Data.id and Data.c.

Slots

ind: Object of class "MatrixorNULLorVector": Indicator of the same length as the data; saves whether each element of the data vector is contaminated or not

Data.id: Object of class "ArrayorNULLorVector": – the ideal data

Data.c: Object of class "ArrayorNULLorVector": – the contaminated data

rate: Object of class "numeric": the contamination rate, so the probability for each random number to be contaminated

distribution.c: Object of class "UnivariateDistribution": the distribution of the ideal data

distribution.id: Object of class "UnivariateDistribution": the distribution of the contaminated data

seed: Object of class "list": the seed the simulation has been generated with

name: Object of class "character": a name for the Contsimulation

filename: Object of class "character": the filename the Contsimulation shall be saved

Data: Object of class "ArrayorNULLorVector": the simulated data

samplesize: Object of class "numeric": the size of the sample, so the dimension of the data

obsDim: Object of class "numeric": the observation dimension of the data

runs: Object of class "numeric": the number of runs of the data

Extends

Class "Dataclass", directly.

Methods

Data.c signature(object = "Contsimulation"): returns the contaminated data

Data.id signature(object = "Contsimulation"): returns the ideal data

Data<- signature(object = "Contsimulation"): ERROR: A modification of simulated data is not allowed.

filename signature(object = "Contsimulation"): returns the the filename

filename<- signature(object = "Contsimulation"): changes the the filename

name signature(object = "Contsimulation"): returns the the name

name<- signature(object = "Contsimulation"): changes the the name

distribution.c signature(object = "Contsimulation"): returns the distribution of the contaminated data

distribution.c<- signature(object = "Contsimulation"): changes the distribution of the contaminated data

distribution.id signature(object = "Contsimulation"): returns the distribution of the ideal data

distribution.id<- signature(object = "Contsimulation"): changes the distribution of the ideal data

seed signature(object = "Contsimulation"): returns the seed

seed<- signature(object = "Contsimulation"): changes the seed

ind signature(object = "Contsimulation"): returns the indicator which saves which data is contaminated

initialize signature(.Object = "Contsimulation"): initialize method

rate signature(object = "Contsimulation"): returns the contamination rate

rate<- signature(object = "Contsimulation"): changes the contamination rate

obsDim signature(object = "Contsimulation"): returns the dimension of the observations

getVersion signature(object = "Contsimulation"): returns the version of this package, under which this object was generated

runs signature(object = "Contsimulation"): returns the number of runs

runs<- signature(object = "Contsimulation"): changes the number of runs

samplesize signature(object = "Contsimulation"): returns the size of the sample

samplesize<- signature(object = "Contsimulation"): changes the size of the sample

savedata signature(object = "Contsimulation"): saves the object without the data in the directory of R. (After loading the data can be reproduced by using simulate.)

simulate signature(x = "Contsimulation"): generates the random numbers for the simulation

plot signature(x = "Contsimulation"): produces a plot of the real data matrix; ; for details confer plot-methods

print `signature(x = "Contsimulation")`: returns filename, seed, the observation dimension, the number of runs, the size of the sample, the contamination rate and the distributions, and, if from a version > 1.8, also the package version under which the object was generated

summary `signature(object = "Contsimulation")`: returns filename, seed, number of runs, the size of the sample, the rate and a statistical summary for each run of the real data

Note

Changing distributions, seed, runs, samplesize or rate deletes possibly simulated data, as it would not fit to the new parameters.

Author(s)

Thomas Stabla <statho3@web.de>,
Florian Camphausen <fcampi@gmx.de>,
Peter Ruckdeschel <peter.ruckdeschel@uni-oldenburg.de>,
Matthias Kohl <Matthias.Kohl@stamats.de>

See Also

Dataclass-class Simulation-class savedata-methods plot-methods simulate-methods summary-methods getVersion-methods

Examples

```
N <- Norm() # N is a standard normal distribution.
C <- Cauchy() # C is a Cauchy distribution
cs <- Contsimulation(filename = "csim",
                     runs = 10,
                     samplesize = 3,
                     seed = setRNG(),
                     distribution.id = N,
                     distribution.c = C,
                     rate = 0.1)
simulate(cs)
# Each of the 30 random numbers is ideal (N-distributed) with
# probability 0.9 and contaminated (C-distributed) with
# probability = 0.1
Data(cs)
Data.id(cs)
Data.c(cs)
ind(cs)
summary(cs)
Data(cs) # different data
savedata(cs) # saves the object in the working directory of R...
load("csim") # loads it again...
Data(cs) # ...without the data - use simulate to return it!
## clean up
unlink("csim")
```

Data-methods *Methods for Function Data in Package 'distrSim'*

Description

Methods to access and change the Data-slot

Methods

Data `signature(object = "Dataclass")`: returns the data

Data<- `signature(object = "Dataclass")`: changes the data (does not work with a simulation or a contsimulation object)

Data<- `signature(object = "Simulation")`: ERROR: A change of the data is not allowed.

Data<- `signature(object = "Contsimulation")`: ERROR: A change of the data is not allowed.

Data.c-methods *Methods for Function Data.c in Package 'distrSim'*

Description

Data.c-methods

Methods

Data.c `signature(object = "Dataclass")`: returns the contaminated data

Data.id-methods *Methods for Function Data.id in Package 'distrSim'*

Description

Data.id-methods

Methods

Data.id `signature(object = "Contsimulation")`: returns the ideal data

Dataclass *Generating function for "Dataclass"*

Description

Generates an object of class "Dataclass".

Usage

```
Dataclass(Data, ...)
## S4 method for signature 'ArrayorNULLorVector'
Dataclass(Data, filename = NULL, name = "Data-Set")
## S4 method for signature 'array'
Dataclass(Data, filename = NULL, name = "Data-Set")
## S4 method for signature 'matrix'
Dataclass(Data, filename = NULL, name = "Data-Set")
## S4 method for signature 'DataframeorSeqDataFrames'
Dataclass(Data, filename = NULL, name = "Data-Set")
```

Arguments

Data	the data to be filled in
filename	Object of class "character": the filename the data shall be saved
name	Object of class "character": a name for the Data
...	additional parameters.

Author(s)

Peter Ruckdeschel <peter.ruckdeschel@uni-oldenburg.de>,

See Also

Dataclass-class

Dataclass-class *Class "Dataclass"*

Description

In an object of type "Dataclass" data can be saved containing any number of runs in any dimension. All information about the data is stored in a unified way.

Objects from the Class

Objects can be created by calls to the generating function, method `Dataclass(Data, filename = "Data-set", name = "Data-Set")`. A Dataclass-object includes, aside from the actual data, a filename and the size of the sample, the observation dimension, and the number of runs, which give the number of rows and columns (and, if more than one run, slices) of the data array.

Slots

`filename`**:** Object of class `"character"`: the filename the data shall be saved

`name`**:** Object of class `"character"`: a name for the Data

`Data`**:** Object of class `"ArrayorNULLorVectororDataframeorSeqDataFrames"`: the actual data, either of type "NULL" (means no data) or "vector" or "array" or "Dataframe" or "SeqDataFrames"

`obsDim`**:** Object of class `"numeric"`: the observation dimension of the data

`runs`**:** Object of class `"numeric"`: the number of runs of the data

`samplesize`**:** Object of class `"numeric"`: the size of the sample

`version`**:** Object of class `"character"`: the package version under which this object was generated

Methods

Data `signature(object = "Dataclass")`: returns the actual data

Data<- `signature(object = "Dataclass")`: changes the data

evaluate `signature(object = "Dataclass", estimator = "function")`: creates an object of type "Evaluation", see there for further information

filename `signature(object = "Dataclass")`: returns the the filename

filename<- `signature(object = "Dataclass")`: changes the the filename

name `signature(object = "Dataclass")`: returns the the name

name<- `signature(object = "Dataclass")`: changes the the name

initialize `signature(.Object = "Dataclass")`: initialize method

obsDim `signature(object = "Dataclass")`: returns the dimension of the observations

runs `signature(object = "Dataclass")`: returns the number of runs

samplesize `signature(object = "Dataclass")`: returns the size of the sample

getVersion `signature(object = "Dataclass")`: returns the version slot of this object

savedata `signature(object = "Dataclass")`: saves the object in the directory of R and also a copy without data

plot `signature(x = "Dataclass")`: produces a plot of the data matrix; ; for details confer `plot-methods`

print `signature(x = "Dataclass")`: returns filename, the observation dimension, the number of runs and the size of the sample, and, if from a version > 1.8, also the package version under which the object was generated

summary `signature(object = "Dataclass")`: returns the same information as print, moreover a statistical summary for each run

Note

The saved Dataclass can be loaded with the usual load-command, the saved comment with the function cload.

Author(s)

Thomas Stabla <statho3@web.de>,
Florian Camphausen <fcampi@gmx.de>,
Peter Ruckdeschel <peter.ruckdeschel@uni-oldenburg.de>,
Matthias Kohl <Matthias.Kohl@stamats.de>

See Also

Simulation-class Contsimulation-class Evaluation-class plot-methods
print-methods summary-methods load cload savedata-methods getVersion-methods

Examples

```
D66 <- Dataclass(filename="N74", Data = matrix(1:36,6))
D66
#
D <- Dataclass(Data = array(c(1,2,3,4,5,6),c(samplesize=2,obsdim=3,Runs=1)),
               filename = "xyz.sav")
# A new object of type "Dataclass" is created.
#
isOldVersion(D) ##NO!
#
savedata(D)
# creates a file called "xyz.sav" where the information is saved and a
# copy "xyz.sav.comment" without data
Data(D) <- array(c(11,12,13,14,15,16),c(samplesize=2,obsdim=3,Runs=1)) # changes the data of D
cload("xyz.sav") # loads the object without data - it is called "D.comment"
D.comment
load("xyz.sav") # loads the original object "D"
Data(D) # the original data: matrix(c(1,2,3,4,5,6),2)
#if you have distrTEst available:
#evaluate(object = D, estimator = mean) # returns the mean of each variable
## clean up
unlink("xyz.sav")
unlink("xyz.sav.comment")
```

---

distribution-methods *Methods for Function distribution in Package 'distrSim'*

Description

distribution-methods

Methods

distribution `signature(object = "Simulation")`: returns the slot `distribution` of the simulation

distribution<- `signature(object = "Simulation")`: changes the slot `distribution` of the simulation

`distribution.c-methods`

Methods for Function distribution.c in Package 'distrSim'

Description

distribution-methods

Methods

distribution.c `signature(object = "Contsimulation")`: returns the distribution of the contaminated data

distribution.c<- `signature(object = "Contsimulation")`: changes the distribution of the contaminated data

---

`distribution.id-methods`

Methods for Function distribution.id in Package 'distrSim'

Description

distribution-methods

Methods

distribution.id `signature(object = "Contsimulation")`: returns the distribution of the ideal data

distribution.id<- `signature(object = "Contsimulation")`: changes the distribution of the ideal data

distrSimMASK *Masking of/by other functions in package "distrSim"*

Description

Provides information on the (intended) masking of and (non-intended) masking by other other functions in package **distrSim**

Usage

```
distrSimMASK(library = NULL)
```

Arguments

library	a character vector with path names of R libraries, or NULL. The default value of NULL corresponds to all libraries currently known. If the default is used, the loaded packages are searched before the libraries

Value

no value is returned

Author(s)

Peter Ruckdeschel <peter.ruckdeschel@uni-oldenburg.de>

Examples

```
## IGNORE_RDIFF_BEGIN
distrSimMASK()
## IGNORE_RDIFF_END
```

distrSimoptions *functions to change the global variables of the package 'distrSim'*

Description

With distrSimoptions and getdistrSimOption you may inspect and change the global variables used by package **distrSim**.

Usage

```
distrSimoptions(...)
getdistrSimOption(x)
```

Arguments

...	any options can be defined, using name = value or by passing a list of such tagged values.
x	a character string holding an option name.

Details

Invoking `distrSimoptions()` with no arguments returns a list with the current values of the options. To access the value of a single option, one should use `getdistrSimOption("MaxNumberofSummarizedRuns")`, e.g., rather than `distroptions("MaxNumberofSummarizedRuns")` which is a *list* of length one.

Value

`distrSimoptions()` returns a list of the global options of **distrSim**.
`distrSimoptions("MaxNumberofSummarizedRuns")` returns the global option `MaxNumberofSummarizedRuns` as a list of length 1.
`distrSimoptions("MaxNumberofSummarizedRuns" = 3)` sets the value of the global option `MaxNumberofSummarizedRuns` to 3. `getdistrSimOption("MaxNumberofSummarizedRuns")` the current value set for option `MaxNumberofSummarizedRuns`

Currently available options

MaxNumberofPlottedObs maximal number of observation plotted; defaults to 4000

MaxNumberofPlottedObsDims maximal number of observation dimensions plotted in parallel; defaults to 6

MaxNumberofPlottedRuns maximal number of runs plotted in parallel; defaults to 6

MaxNumberofSummarizedObsDims maximal number of observation dimensions summarized in parallel; defaults to 6

MaxNumberofSummarizedRuns maximal number of runs summarized in parallel; defaults to 6

Author(s)

Peter Ruckdeschel <peter.ruckdeschel@uni-oldenburg.de>

See Also

`options`, `getOption`, `distroptions`, `getdistrOption`

Examples

```
distrSimoptions()
distrSimoptions("MaxNumberofPlottedObsDims")
distrSimoptions("MaxNumberofPlottedObsDims" = 5)
# or
getdistrSimOption("MaxNumberofPlottedObsDims")
```

filename-methods *Methods for Function filename in Package 'distrSim'*

Description

filename-methods

Methods

filename `signature(object = "Dataclass")`: returns the filename

filename<- `signature(object = "Dataclass")`: changes the filename

ind-methods *Methods for Function ind in Package 'distrSim'*

Description

ind-methods

Methods

ind `signature(object = "Contsimulation")`: returns an indicator showing which data is contaminated

name-methods *Methods for Function name in Package 'distrSim'*

Description

name-methods

Methods

name `signature(object = "Dataclass")`: returns the name

name<- `signature(object = "Dataclass")`: changes the name

obsDim-methods *Methods for Function obsDim in Package 'distrSim'*

Description

obsDim-methods

Methods

obsDim `signature(object = "Dataclass")` or `signature(object = "SeqDataFrames")`: returns the number of obsDim

plot-methods *Methods for Function plot in Package 'distrSim'*

Description

plot-methods

Value

An S3 object of class `c("plotInfo","DiagnInfo")`, i.e., a list containing the information needed to produce the respective plot, which at a later stage could be used by different graphic engines (like, e.g. `ggplot`) to produce the plot in a different framework. A more detailed description will follow in a subsequent version.

Methods

plot `signature(x = "Dataclass", y="missing")`: produces a plot of the data matrix; optional arguments:

obs0 the indices of observations to be plotted;— of this vector `runs0` maximally `MaxNumberofPlottedObs` runs are plotted where `MaxNumberofPlottedObs` is a global option, see `distrSimoptions`

runs0 the indices of runs to be plotted; — of this vector `runs0` maximally `MaxNumberofPlottedRuns` runs are plotted where `MaxNumberofPlottedRuns` is a global option, see `distrSimoptions`

dims0 the indices of observation dimensions to be plotted; — of this vector `dims0` maximally `MaxNumberofPlottedObsDims` dimensions are plotted where `MaxNumberofPlottedObsDims` is a global option, see `distrSimoptions`

typical plot arguments may be passed; some have slightly different meaning (compare `plot.default`):

ylim may be transmitted matrixwise (in a 2 x (number of dimensions) matrix)) or globally, using the usual recycling rules

col,cex,pch may be transmitted vectorwise (for the different dimensions) or globally, using the usual recycling rules

xlab,ylab,type are ignored

plot `signature(x = "Simulation", y="missing")`: produces a plot of the data matrix; optional and typical plot arguments: as with `signature(x = "Dataclass", y="missing")`

plot `signature(x = "Contsimulation")`: produces a plot of the real data matrix; optional arguments: as with `signature(x = "Dataclass", y="missing")`; typical plot arguments may be passed; some have slightly different meaning (compare `plot.default`):

ylim as with `signature(x = "Simulation", y="missing")` and `signature(x = "Dataclass", y="missing")`

col,cex,pch are ignored

col.id,cex.id,pch.id as `col,cex,pch` for `signature(x = "Dataclass", y="missing")` but only for ideal part of the data

col.c,cex.c,pch.c as `col,cex,pch` for `signature(x = "Dataclass", y="missing")` but only for contaminated part of the data

xlab,ylab,type are ignored

---

print-methods *Methods for Function print/show in Package 'distrSim'*

Description

print and show - methods

Methods

print `signature(x = "SeqDataFrames")`: extra argument `short = FALSE`; if `TRUE` the output is cut according to `distrSimoptions`.

print `signature(x = "Dataclass")`: returns name, filename, the observation dimension, the number of runs and the size of the sample, and, if from a version > 1.8, also the package version under which the object was generated

print `signature(x = "Simulation")`: as for `signature(x = "Dataclass")` and, additionally, seed and the distribution

print `signature(x = "Contsimulation")`: as for `signature(x = "Dataclass")` and, additionally, seed, the contamination rate and the distributions

show `signature(x = "Dataclass")`, `signature(x = "Simulation")`, `signature(x = "Contsimulation")`, `signature(x = "SeqDataFrames")`: as corresponding `print` method

---

rate-methods *Methods for Function rate in Package 'distrSim'*

Description

rate-methods

Methods

rate `signature(object = "Contsimulation")`: returns the contamination rate

rate<- `signature(object = "Contsimulation")`: modifies the contamination rate

runs-methods *Methods for Function runs in Package 'distrSim'*

Description

runs-methods

Methods

runs `signature(object = "SeqDataFrames")`: returns the number of runs

runs `signature(object = "Dataclass")`: returns the number of runs

runs<- `signature(object = "Simulation")`: changes the number of runs

runs<- `signature(object = "Contsimulation")`: changes the number of runs

samplesize-methods *Methods for Function samplesize in Package 'distrSim'*

Description

samplesize-methods

Methods

samplesize `signature(object = "SeqDataFrames")`: returns the size[s] of the sample[s]

samplesize `signature(object = "Dataclass")`: returns the size of the sample

samplesize<- `signature(object = "Simulation")`: changes the size of the sample

samplesize<- `signature(object = "Contsimulation")`: changes the size of the sample

savedata-methods *Methods for Function savedata in Package 'distrSim'*

Description

savedata-methods

Methods

savedata `signature(object = "Dataclass")`: saves the object (with the data) in the directory of R

savedata `signature(object = "Simulation")`: saves the object without the data in the directory of R (After loading the data can be reproduced by using simulate.)

savedata `signature(object = "Contsimulation")`: saves the object without the data in the directory of R (After loading the data can be reproduced by using simulate.)

Note

For an example, see Simulation-class and Contsimulation-class

See Also

Dataclass-class Simulation-class Contsimulation-class Evaluation-class

---

seed-methods *Methods for Function seed in Package 'distrSim'*

Description

seed-methods

Methods

seed signature(object = "Simulation"): returns the slot seed of an object of class "Simulation"

seed<- signature(object = "Simulation"): changes the slot seed of an object of class "Simulation"

seed signature(object = "Contsimulation"): returns the slot seed of an object of class "Contsimulation"

seed<- signature(object = "Contsimulation"): changes the slot seed of an object of class "Contsimulation"

Note

The value to which the seed is set has to be consistent with the **setRNG**-package; to draw a "new" simulation, use something like seed(X)<-setRNG();simulate(X); cf. manual to this package, p.~9

SeqDataFrames-class *Class "SeqDataFrames"*

Description

An object of type "SeqDataFrames" is a list of data frames, all of which with the same numbers and names of columns (ideally with the same data-types for the columns), but with possibly varying number of rows; with correponding index operators it behaves like a three-dimensional array with dimensions sample size x observation dimension x runs.

Details

There is a validity method checking for each member of the list being a data frame and for the accordance of the column structures of the data frames.

Objects from the Class

Objects can be created by calls of the form `SeqDataFrames(...)`, where the `...` are a list of dataframes with according column structure.

Slots

`data`: a list of data frames

Methods

[`signature(x = "SeqDataFrames")`: returns (slices of) the data

[<- `signature(x = "SeqDataFrames")`: modifies (slices of) the data

print `signature(x = "SeqDataFrames", obs0 = NULL, dims0 = NULL, runs0 = NULL, short = FALSE, ...)`: slices can be printed and, if argument `short== TRUE` only a bounded number of dimensions is shown.

show `signature(object = "SeqDataFrames")`: a call to `print(x)`

names `signature(x = "SeqDataFrames")`: returns the names of the runs

runnames `signature(x = "SeqDataFrames")`: returns the names of the runs

obsdimnames `signature(x = "SeqDataFrames")`: returns the names of the observation dimensions

obsDim `signature(x = "SeqDataFrames")`: returns the dimension of the observations

runs `signature(x = "SeqDataFrames")`: returns the number of runs

samplesize `signature(x = "SeqDataFrames")`: returns the size of the samples for each run

rbind `signature(x = "SeqDataFrames")`: concatenates different a list of `SeqDataFrames` object (with the same column structure) to a new object of class `SeqDataFrames` to do so we mask the `rbind` method from package **base**

Author(s)

Thomas Stabla <statho3@web.de>,
Florian Camphausen <fcampi@gmx.de>,
Peter Ruckdeschel <peter.ruckdeschel@uni-oldenburg.de>,
Matthias Kohl <Matthias.Kohl@stamats.de>

See Also

`[-methods` `print-methods` `summary-methods`

simulate-methods *Methods for Function simulate in Package 'distrSim'*

Description

simulate-methods

Methods

simulate `signature(object = "Simulation")`: generates the random numbers for the simulation

simulate `signature(object = "Contsimulation")`: generates the random numbers for the simulation

Simulation-class *Class "Simulation"*

Description

In an object of type `Simulation` data can be simulated in any distribution and size.

Objects from the Class

Objects can be created by calls of the form `Simulation(filename, samplesize, runs, seed, distribution)` (observation dimension is deduced from slot `distribution`). A `Simulation`-object includes a filename, a name for the simulation, the number of runs, the size of the sample, the seed and the distribution of the random numbers. The slot `Data` stays empty until the method simulate has been used.

Slots

`seed`**:** Object of class `"list"`: the seed the simulation has been generated with

`distribution`**:** Object of class `"UnivariateDistribution"`: the distribution of the random numbers

`filename`**:** Object of class `"character"`: the filename the simulation shall be saved

`name`**:** Object of class `"character"`: a name for the Simulation

`Data`**:** Object of class `"ArrayorNULLorVector"`: the simulated data

`samplesize`**:** Object of class `"numeric"`: the size of the sample

`obsDim`**:** Object of class `"numeric"`: the dimension of the observations of the data

`runs`**:** Object of class `"numeric"`: the number of runs of the data

`version`**:** Object of class `"character"`: the version of this package, under which this object was generated

Extends

Class "Dataclass", directly.

Methods

Data signature(object = "Simulation"): returns the simulated data.

Data<- signature(object = "Simulation"): ERROR: A modification of simulated data is not allowed.

filename signature(object = "Simulation"): returns the the filename

filename<- signature(object = "Simulation"): changes the the filename

name signature(object = "Simulation"): returns the the name

name<- signature(object = "Simulation"): changes the the name

distribution signature(object = "Simulation"): returns the distribution

distribution<- signature(object = "Simulation"): changes the distribution

seed signature(object = "Simulation"): returns the seed

seed<- signature(object = "Simulation"): changes the seed

obsDim signature(object = "Simulation"): returns the dimension of the observations

getVersion signature(object = "Simulation"): returns the version of this package, under which this object was generated

runs signature(object = "Simulation"): returns the number of runs

runs<- signature(object = "Simulation"): changes the number of runs

samplesize signature(object = "Simulation"): returns the size of the sample

samplesize<- signature(object = "Simulation"): changes the size of the sample

savedata signature(object = "Simulation"): saves the object without the data in the directory of R (After loading the data can be reproduced by using simulate.)

initialize signature(.Object = "Simulation"): initialize method

plot signature(x = "Simulation"): produces a plot of the data matrix; for details confer plot-methods

print signature(x = "Simulation"): returns filename, seed, the observation dimension, the number of runs, the size of the sample, the distribution generating the simulations, and, if from a version > 1.8, also the package version under which the object was generated

show signature(x = "Simulation"): the same as print.

simulate signature(x = "Simulation"): generates the random numbers for the simulation

summary signature(object = "Simulation"): returns filename, seed, number of runs, the size of the sample and a statistical summary for each run

Note

Changing distribution, seed, runs or samplesize deletes possibly simulated data, as it would not fit to the new parameters.

Author(s)

Thomas Stabla <statho3@web.de>,
Florian Camphausen <fcampi@gmx.de>,
Peter Ruckdeschel <peter.ruckdeschel@uni-oldenburg.de>,
Matthias Kohl <Matthias.Kohl@stamats.de>

See Also

Dataclass-class Contsimulation-class plot-methods print-methods summary-methods simulate-methods getVersion-methods

Examples

```
N=Norm() # N is a standard normal distribution.
S=Simulation(filename="xyz",runs=10,samplesize=3,seed=setRNG(),distribution=N)
Data(S) # no data yet
simulate(S)
Data(S) # now there are random numbers
Data(S) # the same data as before because the seed has not changed
seed(S)=setRNG()
simulate(S)
Data(S) # different data
savedata(S) # saves the object in the directory of R...
load("xyz") # loads it again...
Data(S) # ...without the data - use simulate to return it!
unlink("xyz")
```

Subsetting-methods *Subsetting/Indexing methods for SeqDataFrames objects in Package 'distrSim'*

Description

Operators acting on SeqDataFrames objects to extract or replace parts.

Usage

```
## S4 method for signature 'SeqDataFrames'
x[i, j, k, drop = FALSE]

## S4 replacement method for signature 'SeqDataFrames'
x[i, j, k] <- value
```

Arguments

x	object of class SeqDataFrames from which to extract element(s) or in which to replace element(s).
i	observation index; may be missing

j	observation dimension index; may be missing
k	run index; may be missing
drop	as in the general indexing functions
value	modification to be assigned to

Value

again an object of class SeqDataFrames with the prescribed indices / values

Methods

"[" signature(x = "SeqDataFrames"): access method for indices for objects of class SeqDataFrames

"[<-" signature(x = "SeqDataFrames"): replacement method for indices for objects of class SeqDataFrames

See Also

"["

Examples

```
s0 <- matrix(1:6,3,2)
d0 <- data.frame(s0)
d1 <- data.frame(s0 + 3)
SF <- SeqDataFrames(d0, d1)
SF[1,2,1]
```

---

summary-methods *Methods for Function summary in Package 'distrSim'*

Description

summary-methods

Methods

summary signature(object = "Dataclass"): returns the same information as print, moreover a statistical summary for each run; optional arguments:

- **runs0** the indices of runs to be summarized; — of this vector runs0 maximally MaxNumberofSummarizedRuns runs are summarized where MaxNumberofSummarizedRuns is a global option, see distrSimoptions
- **dims0** the indices of observation dimensions to be summarized; — of this vector dims0 maximally MaxNumberofSummarizedObsDims dimensions are summarized where MaxNumberofSummarizedObsDims is a global option, see distrSimoptions

summary signature(object = "Simulation"): returns name, filename, seed, number of runs, the size of the sample and a statistical summary for each run; optional arguments: as with signature(object = "Dataclass")

summary `signature(object = "Contsimulation")`: returns name, filename, seed, number of runs, the size of the sample, the rate and a statistical summary for each run of the real data; optional arguments: as with `signature(object = "Dataclass")`

`vectororNULL-class` *Classes "vectororNULL", "MatrixorNULLorVector", "ArrayorNULLorVector", "DataframeorSeqDataFrame" and "ArrayorNULLorVectororDataframeorSeqDataFrames"*

Description

auxiliary classes; may contain either a vector or NULL, [or a matrix, or an array, respectively], cf. J. Chambers, "green book".

Objects from the Classes

these classes are all virtual: No objects may be created from it.

Methods

No methods defined with classs "vectororNULL", "MatrixorNULLorVector", and "ArrayorNULLorVectororDataframeorSeqDataFrames" in the signature. However, the generating function `Dataclass` dispatches according to "DataframeorSeqDataFrames" or "ArrayorNULLorVector".

Note

Dataclass-class can save data either of type "NULL" (means no data) or "vector" or "array" or "data.frame"

Author(s)

Thomas Stabla <statho3@web.de>,
Florian Camphausen <fcampi@gmx.de>,
Peter Ruckdeschel <peter.ruckdeschel@uni-oldenburg.de>,
Matthias Kohl <Matthias.Kohl@stamats.de>

See Also

`Dataclass-class`

`Version Management` *Methods for Version Management in Package 'distrSim'*

Description

Version-Management-methods

Usage

```
## S4 method for signature 'Dataclass'
getVersion(object)
## S4 method for signature 'Dataclass'
conv2NewVersion(object)
```

Arguments

`object`	object of class `"Dataclass"` (or subclasses)

Far-reaching Change in Design

From version 1.8 of this package on, we have changed the format how data / simulations are stored: In order to be able to cope with multivariate distributions, regression distributions and (later) time series distributions, we have switched to the common array format: samplesize x obsDim x runs; for saved objects from earlier versions, we provide the functions `isOldVersion` (from package **distr**) and `conv2NewVersion` to check whether the object was generated by an older version of this package and to convert such an object to the new format, respectively.

Methods

getVersion `signature(object = "Dataclass")`: returns slot `version` of an object of class `"Dataclass"`.

conv2NewVersion `signature(object = "Dataclass")`: changes an object with format runs x samplesize (old format) to samplesize x obsDim x runs (new format)

See Also

`isOldVersion`, `conv2NewVersion`

Index