Package ‘RandVar’

August 1, 2018

Version 1.1.0
Date 2018-08-01
Title Implementation of Random Variables
Description Implements random variables by means of S4 classes and methods.
Depends R (>= 2.14.0), methods, distr(>= 2.5.2), distrEx(>= 2.5)
Imports startupmsg
ByteCompile yes
LazyLoad yes
License LGPL-3
Encoding latin1
URL http://robast.r-forge.r-project.org/
LastChangedDate {$LastChangedDate: 2018-08-01 05:06:49 +0200 (Mi, 01. Aug 2018) $}
LastChangedRevision {$LastChangedRevision: 1083 $}
VCS/SVNRevision 1081
NeedsCompilation no
Author Matthias Kohl [cre, cph, aut],
Peter Ruckdeschel [aut, cph]
Maintainer Matthias Kohl <Matthias.Kohl@stamats.de>
Repository CRAN
Date/Publication 2018-08-01 20:30:03 UTC

R topics documented:

RandVar-package ................................................................. 2
EuclRandMatrix ............................................................... 3
EuclRandMatrix-class ....................................................... 4
EuclRandVariable ............................................................. 7
EuclRandVariable-class ..................................................... 9
RandVar-package

EuclRandVarList .......................................................... 12
EuclRandVarList-class ..................................................... 13
OptionalrSpace-class ...................................................... 15
RandVariable ................................................................. 16
RandVariable-class .......................................................... 17
RealRandVariable ............................................................. 19
RealRandVariable-class .................................................... 20

Index 21

RandVar-package  Implementation of Random Variables

Description

Implementation of random variables by means of S4 classes and methods.

Details

Package: RandVar
Version: 1.1.0
Date: 2018-08-01
Depends: R (>= 2.14.0), methods, distr(>= 2.5.2), distrEx(>= 2.5)
Imports: startupmsg
ByteCompile: yes
License: LGPL-3
URL: http://robast.r-forge.r-project.org/
VCS/SVNRevision: 1081

Package versions

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

Author(s)

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

Maintainer: Matthias Kohl <matthias.kohl@stamats.de>
EuclRandMatrix

References


See Also
distr-package, distrEx-package

Examples

library(RandVar)
#vignette("RandVar")

euclrandmatrix
Generating function for EuclRandMatrix-class

Description

Generates an object of class "EuclRandMatrix".

Usage

EuclRandMatrix(Map = list(function(x){1}), nrow = 1, ncol = 1, 
Domain = NULL, dimension = 1, Range)

Arguments

Map list of functions forming the map.
nrow number of rows.
ncol number of columns.
Domain object of class "OptionalrSpace": domain of Map
dimension positive integer: dimension of the range of Map
Range object of class "OptionalrSpace": range of Map

Value

Object of class "EuclRandMatrix"

Author(s)

Matthias Kohl <Matthias.Kohl@stamats.de>

See Also

EuclRandMatrix-class
Examples

```r
L1 <- list(function(x)({x}), function(x)({x^2}), function(x)({x^3}), function(x)({x^4}),
            function(x)({x^5}), function(x)({x^6}))
L2 <- list(function(x)({exp(x)}), function(x)({abs(x)}),
            function(x)({sin(x)}), function(x)({floor(x)}))

R1 <- EuclRandMatrix(Map = L1, nrow = 3, Domain = Reals(), dimension = 1)
R1[1:2, 2]
R1[1:2, 1:2]
Map(R1[1:2])
Map(t(R1[2,1]))

R2 <- EuclRandMatrix(Map = L2, ncol = 2, Domain = Reals(), dimension = 1)
(DL <- imageDistr(R2, Norm()))
plot(DL)

Map(gamma(R2)) # "Math" group

## "Arith" group
Map(R2 / R1)
Map(R2 * R2)
```

## The function is currently defined as
```r
function(Map = list(function(x)({}), nrow = 1, ncol = 1, 
           Domain = NULL, dimension = 1) {
    if (missing(nrow))
        nrow <- ceiling(length(Map)/ncol)
    else if (missing(ncol))
        ncol <- ceiling(length(Map)/nrow)
    if(missing(Range))
        return(new("EuclRandMatrix", Map = Map, Domain = Domain,
                     Range = Euclideanspace(dimension = dimension),
                     Dim = as.integer(c(nrow, ncol))))
    else
        return(new("EuclRandMatrix", Map = Map, Domain = Domain,
                     Range = Range, Dim = as.integer(c(nrow, ncol))))
}
```

---

**EuclRandMatrix-class**

**Euclidean random matrix**

### Description

Class of Euclidean random matrices.

### Objects from the Class

Objects can be created by calls of the form `new("EuclRandMatrix", ...)`. More frequently they are created via the generating function `EuclRandMatrix`. 
EuclRandMatrix-class

Slots

- **Dim** vector of positive integers: Dimensions of the random matrix.
- **Map** Object of class "list": list of functions.
- **Domain** Object of class "OptionalrSpace" domain of the random matrix.
- **Range** Object of class "OptionalrSpace" range of the random matrix.

Extends

Class "EuclRandVariable", directly.
Class "RandVariable", by class "EuclRandVariable".

Methods

- **coerce** signature(from = "EuclRandMatrix", to = "EuclRandVarList"): create a "EuclRandVarList" object from a Euclidean random matrix.
- **[** signature(x = "EuclRandMatrix"): generates a new Euclidean random variable/matrix by extracting elements of the slot Map of x.
- **Dim** signature(object = "EuclRandMatrix"): accessor function for slot Dim.
- **Dim<-** signature(object = "EuclRandMatrix", ): replacement function for slot Dim.
- **ncol** signature(x = "EuclRandMatrix"): number of columns of x.
- **nrow** signature(x = "EuclRandMatrix"): number of rows of x.
- **dimension** signature(object = "EuclRandMatrix"): dimension of the Euclidean random variable.
- **evalRandVar** signature(RandVar = "EuclRandMatrix", x = "numeric"): evaluate the slot Map of RandVar at x.
- **evalRandVar** signature(RandVar = "EuclRandMatrix", x = "matrix"): evaluate the slot Map of RandVar at x.
- **evalRandVar** signature(RandVar = "EuclRandMatrix", x = "numeric", distr = "Distribution"): evaluate the slot Map of RandVar at x assuming a probability space with distribution distr. In case x does not lie in the support of distr NA is returned.
- **evalRandVar** signature(RandVar = "EuclRandMatrix", x = "matrix", distr = "Distribution"): evaluate the slot Map of RandVar at rows of x assuming a probability space with distribution distr. For those rows of x which do not lie in the support of distr NA is returned.
- **t** signature(x = "EuclRandMatrix"): transposes x. In addition, the results of the functions in the slot Map of x are transposed.
- **show** signature(object = "EuclRandMatrix")
  ```
  %%% signature(x = "matrix", y = "EuclRandMatrix"): matrix multiplication of x and y. Generates an object of class "EuclRandMatrix".
  %%% signature(x = "numeric", y = "EuclRandMatrix"): matrix multiplication of x and y. Generates an object of class "EuclRandMatrix".
  %%% signature(x = "EuclRandVariable", y = "EuclRandMatrix"): matrix multiplication of x and y. Generates an object of class "EuclRandMatrix".
  ```
Generates an object of class "EuclRandMatrix".

%*% signature(x = "EuclRandMatrix", y = "numeric"): matrix multiplication of x and y. Generates an object of class "EuclRandMatrix".

%*% signature(x = "EuclRandMatrix", y = "EuclRandMatrix"): matrix multiplication of x and y. Generates an object of class "EuclRandMatrix".

%*% signature(x = "EuclRandMatrix", y = "EuclRandVariable"): matrix multiplication of x and y. Generates an object of class "EuclRandMatrix".

Arith signature(e1 = "numeric", e2 = "EuclRandMatrix"): Given a numeric vector e1, a Euclidean random matrix e2 and an arithmetic operator op, the Euclidean random matrix e1 op e2 is returned.

Arith signature(e1 = "EuclRandMatrix", e2 = "numeric"): Given a Euclidean random matrix e1, a numeric vector e2, and an arithmetic operator op, the Euclidean random matrix e1 op e2 is returned.

Arith signature(e1 = "EuclRandMatrix", e2 = "EuclRandMatrix"): Given two Euclidean random matrices e1 and e2, and an arithmetic operator op, the Euclidean random matrix e1 op e2 is returned.

Math signature(x = "EuclRandMatrix"): Given a "Math" group generic fct, the Euclidean random matrix fct(x) is returned.

E signature(object = "UnivariateDistribution", fun = "EuclRandMatrix", cond = "missing"): expectation of fun under univariate distributions.

E signature(object = "AbscontDistribution", fun = "EuclRandMatrix", cond = "missing"): expectation of fun under absolutely continuous univariate distributions.

E signature(object = "DiscreteDistribution", fun = "EuclRandMatrix", cond = "missing"): expectation of fun under discrete univariate distributions.

E signature(object = "MultivariateDistribution", fun = "EuclRandMatrix", cond = "missing"): expectation of fun under multivariate distributions.

E signature(object = "DiscreteMVDDistribution", fun = "EuclRandMatrix", cond = "missing"): expectation of fun under discrete multivariate distributions.

E signature(object = "UnivariateCondDistribution", fun = "EuclRandMatrix", cond = "numeric"): conditional expectation of fun under conditional univariate distributions.

E signature(object = "AbscontCondDistribution", fun = "EuclRandMatrix", cond = "numeric"): conditional expectation of fun under absolutely continuous conditional univariate distributions.

E signature(object = "DiscreteCondDistribution", fun = "EuclRandMatrix", cond = "numeric"): conditional expectation of fun under discrete conditional univariate distributions.

Author(s)

Matthias Kohl <Matthias.Kohl@stamats.de>

See Also

EuclRandMatrix, RandVariable-class, EuclRandVariable-class, EuclRandVarList-class, Distribution-class, Arith, Math, E
EuclRandVariable

Examples

L1 <- list(function(x){x}, function(x){x^2}, function(x){x^3}, function(x){x^4},
           function(x){x^5}, function(x){x^6})
L2 <- list(function(x){exp(x)}, function(x){abs(x)},
           function(x){sin(x)}, function(x){floor(x)})

R1 <- new("EuclRandMatrix", Map = L1, Dim = as.integer(c(3,2)),
         Domain = Reals(), Range = Reals())

dimension(R1)
R1[1:2,]
R1[1:2,1:2]
Map(R1[1,2])
Map(t(R1[2,1]))

R2 <- EuclRandMatrix(Map = L2, ncol = 2, Domain = Reals(), dimension = 1)
dimension(R2)
(DL <- imageDistr(R2, Norm()))
plot(DL)

Map(gamma(R2)) # "Math" group

## "Arith" group
Map(2/R1)
Map(R2 %*% R2)

EuclRandVariable

Generating function for EuclRandVariable-class

Description

Generates an object of class "EuclRandVariable".

Usage

EuclRandVariable(Map = list(function(x){1}), Domain = NULL,
                 dimension = 1, Range)

Arguments

Map        list of functions forming the map.
Domain     object of class "OptionalrSpace": domain of Map
dimension  positive integer: dimension of the range of Map
Range      object of class "OptionalrSpace": range of Map

Value

Object of class "EuclRandVariable"
EuclRandVariable

Author(s)
Matthias Kohl <Matthias.Kohl@stamats.de>

See Also
EuclRandVariable-class

Examples

L1 <- list(function(x)(x), function(x)(x^2), function(x)(x^3), function(x)(x^4))
L2 <- list(function(x)(exp(x)), function(x)(abs(x)),
    function(x)(sin(x)), function(x)(floor(x)))

R1 <- EuclRandVariable(Map = L1, Domain = Reals(), dimension = 1)
Map(R1)
Range(R1) <- Reals()
R1[2]
Map(R1[3])
Map(R1[c(1,2,4)])
Map(R1[2:4])
set.seed(123)
evalRandVar(R1, rnorm(1))
x <- as.matrix(rnorm(10))
res.R1 <- evalRandVar(R1, x)
res.R1[2,1] # results for Map(R1)[2](x)
res.R1[2,1] # results for Map(R1)[2][1](x[1,1])

R2 <- EuclRandVariable(L2, Domain = Reals(), dimension = 1)
DL1 <- imageDistr(R2, Norm())
plot(DL1)

Domain(R2) <- EuclideanSpace(dimension = 2)
Range(R2) <- EuclideanSpace(dimension = 2)
(X <- matrix(c(x, rnorm(10)), ncol = 2))
res.R2 <- evalRandVar(R2, X)
res.R2[3,1] # results for Map(R2)[3][1](X[1,1])

Map(log(abs(R2))) # "Math" group generic

    # "Arith" group generic
Map(3 + R1)
Map(c(1,3,5) * R1)
try(1:5 * R1) # error
Map(1:2 * R2)
Map(R2 - 5)
Map(R1 ^ R1)

# The function is currently defined as
function(Map = list(function(x){1}), Domain = NULL, dimension = 1, Range) {
  if(missing(Range))
EuclRandVariable-class

return(new("EuclRandVariable", Map = Map, Domain = Domain, Range = EuclideanSpace(dimension = dimension)))
else
return(new("EuclRandVariable", Map = Map, Domain = Domain, Range = Range))
}

EuclRandVariable-class

Euclidean random variable

Description

Class of Euclidean random variables.

Objects from the Class

Objects can be created by calls of the form new("EuclRandVariable", ...). More frequently they are created via the generating function EuclRandVariable.

Slots

Map Object of class "list": list of functions.
Domain Object of class "OptionalrSpace": domain of the random variable.
Range Object of class "EuclideanSpace": range of the random variable.

Extends

Class "RandVariable", directly.

Methods

coerce signature(from = "EuclRandVariable", to = "EuclRandMatrix"): create a "EuclRandMatrix" object from a Euclidean random variable.

coerce signature(from = "EuclRandVariable", to = "EuclRandVarList"): create a "EuclRandVarList" object from a Euclidean random variable.

Range<- signature(object = "EuclRandVariable"): replacement function for the slot Range.

[ signature(x = "EuclRandVariable"): generates a new Euclidean random variable by extracting elements of the slot Map of x.

evalRandVar signature(RandVar = "EuclRandVariable", x = "numeric", distr = "missing"): evaluate the slot Map of RandVar at x.

evalRandVar signature(RandVar = "EuclRandVariable", x = "matrix", distr = "missing"): evaluate the slot Map of RandVar at rows of x.

evalRandVar signature(RandVar = "EuclRandVariable", x = "numeric", distr = "Distribution"): evaluate the slot Map of RandVar at x assuming a probability space with distribution distr. In case x does not lie in the support of distr NA is returned.
**euclRandVariable-class**

**evalRandVar** signature(RandVar = "EuclRandVariable", x = "matrix", distr = "Distribution"): evaluate the slot Map of RandVar at rows of x assuming a probability space with distribution distr. For those rows of x which do not lie in the support of distr NA is returned.

**imageDistr** signature(RandVar = "EuclRandVariable", distr = "Distribution"): image distribution of distr under RandVar. Returns an object of class "DistrList".

**dimension** signature(object = "EuclRandVariable"): dimension of the Euclidean random variable.

**t** signature(x = "EuclRandVariable"): returns an object of class "EuclRandMatrix" where the r.h. results of the functions in the slot Map of x are transposed.

**%*%** signature(x = "matrix", y = "EuclRandVariable"): matrix multiplication of x and y. Generates an object of class "EuclRandMatrix".

**%*%** signature(x = "EuclRandVariable", y = "matrix"): matrix multiplication of x and y. Generates an object of class "EuclRandMatrix".

**%*%** signature(x = "numeric", y = "EuclRandVariable"): generates an object of class "EuclRandMatrix" (1 x 1 matrix) by multiplying (scalar/inner product) x and y.

**%*%** signature(x = "EuclRandVariable", y = "numeric"): generates an object of class "EuclRandMatrix" (1 x 1 matrix) by multiplying (scalar/inner product) x and y.

**%*%** signature(x = "EuclRandVariable", y = "EuclRandVariable"): generates an object of class "EuclRandMatrix" (1 x 1 matrix) by multiplying (scalar/inner product) x and y.

**%*%** signature(x = "EuclRandVariable", y = "EuclRandMatrix"): matrix multiplication of x and y. Generates an object of class "EuclRandMatrix".

**%*%** signature(x = "numeric", y = "numeric"): generates an object of class "EuclRandMatrix" (1 x 1 matrix) by multiplying (scalar/inner product) x and y.

**%*%** signature(x = "EuclRandvariable", y = "EuclRandvariable"): generates an object of class "EuclRandMatrix" (1 x 1 matrix) by multiplying (scalar/inner product) x and y.

**%*%** signature(x = "EuclRandvariable", y = "EuclRandvariable"): generates an object of class "EuclRandMatrix" (1 x 1 matrix) by multiplying (scalar/inner product) x and y.

**%*%** signature(x = "EuclRandvariable", y = "EuclRandvariable"): generates an object of class "EuclRandMatrix" (1 x 1 matrix) by multiplying (scalar/inner product) x and y.

**%*%** signature(x = "EuclRandvariable", y = "EuclRandvariable"): generates an object of class "EuclRandMatrix" (1 x 1 matrix) by multiplying (scalar/inner product) x and y.

**%*%** signature(x = "EuclRandvariable", y = "EuclRandvariable"): generates an object of class "EuclRandMatrix" (1 x 1 matrix) by multiplying (scalar/inner product) x and y.

**Arith** signature(e1 = "numeric", e2 = "EuclRandVariable"): Given a numeric vector e1, a Euclidean random variable e2 and an arithmetic operator op, the Euclidean random variable e1 op e2 is returned.

**Arith** signature(e1 = "EuclRandVariable", e2 = "numeric"): Given a numeric vector e2, a Euclidean random variable e1 and an arithmetic operator op, the Euclidean random variable e1 op e2 is returned.

**Arith** signature(e1 = "EuclRandVariable", e2 = "EuclRandVariable"): Given two Euclidean random variables e1, e2 and an arithmetic operator op, the Euclidean random variable e1 op e2 is returned.

**Math** signature(x = "EuclRandVariable"): Given a "Math" group generic fct, the Euclidean random variable fct(x) is returned.

**E** signature(object = "UnivariateDistribution", fun = "EuclRandVariable", cond = "missing"): expectation of fun under univariate distributions.

**E** signature(object = "AbscontDistribution", fun = "EuclRandVariable", cond = "missing"): expectation of fun under absolutely continuous univariate distributions.

**E** signature(object = "DiscreteDistribution", fun = "EuclRandVariable", cond = "missing"): expectation of fun under discrete univariate distributions.

**E** signature(object = "MultivariateDistribution", fun = "EuclRandVariable", cond = "missing"): expectation of fun under multivariate distributions.
EuclRandVariable-class

E signature(object = "DiscreteMWDistribution", fun = "EuclRandVariable", cond = "missing"): expectation of fun under discrete multivariate distributions.

E signature(object = "UnivariateCondDistribution", fun = "EuclRandVariable", cond = "numeric"): conditional expectation of fun under absolutely continuous conditional univariate distributions.

E signature(object = "UnivariateCondDistribution", fun = "EuclRandVariable", cond = "numeric"): conditional expectation of fun under discrete conditional univariate distributions.

E signature(object = "UnivariateCondDistribution", fun = "EuclRandVariable", cond = "numeric"): conditional expectation of fun under conditional univariate distributions.

Author(s)

Matthias Kohl <Matthias.Kohl@stamats.de>

See Also

EuclRandVariable, RandVariable-class, EuclRandMatrix-class, EuclRandVarList-class, Distribution-class, Arith, Math, E

Examples

L1 <- list(function(x){x}, function(x){x^2}, function(x){x^3}, function(x){x^4})
L2 <- list(function(x){exp(x)}, function(x){abs(x)},
          function(x){sin(x)}, function(x){floor(x)})
R1 <- new("EuclRandVariable", Map = L1, Domain = Reals(), Range = Reals())
dimension(R1)
Map(R1)
Range(R1)
R1[2]
Map(R1[3])
Map(R1[c(1,2,4)])
Map(R1[2:4])
set.seed(123)
evalRandVar(R1, rnorm(1))
x <- as.matrix(rnorm(10))
res.R1 <- evalRandVar(R1, x)
res.R1[2, ,] # results for Map(R1)[[2]](x)
res.R1[2,1,] # results for Map(R1)[[2]](x[1,])

R2 <- EuclRandVariable(L2, Domain = Reals(), dimension = 1)
dimension(R2)
DL1 <- imageDistr(R2, Norm())
plot(DL1)

Domain(R2) <- EuclideanSpace(dimension = 2)
Range(R2) <- EuclideanSpace(dimension = 2)
dimension(R2)
(X <- matrix(c(x, rnorm(10)), ncol = 2))
res.R2 <- evalRandVar(R2, X)
res.R2[3,1] # results for Map(R2)[[3]](X[,1])
EuclRandVarList

Generating function for EuclRandVarList-class

Description

Generates an object of class "EuclRandVarList".

Usage

EuclRandVarList(...)

Arguments

... Objects of class "EuclRandVariable" which shall form the list of Euclidean random variables.

Value

Object of class "EuclRandVarList"

Author(s)

Matthias Kohl <Matthias.Kohl@stamats.de>

See Also

EuclRandVarList-class

Examples

L1 <- list(function(x){x}, function(x){x^2}, function(x){x^3}, function(x){x^4},
  function(x){x^5}, function(x){x^6})
L2 <- list(function(x){exp(x)}, function(x){abs(x)},
  function(x){sin(x)}, function(x){floor(x)})
R1 <- new("EuclRandVariable", Map = L1, ncol = 2, Domain = Reals(), Range = Reals())
R2 <- EuclRandMatrix(Map = L1, ncol = 2, Domain = Reals(), dimension = 1)
R3 <- EuclRandMatrix(Map = L2, ncol = 2, Domain = Reals(), dimension = 1)
EuclRandVarList-class

Create a list of Euclidean random variables

Objects from the Class

Objects can be created by calls of the form `new("EuclRandVarList", ...). More frequently they are created via the generating function EuclRandVarList.

Slots

.Data Object of class "list". A list of Euclidean random variables.

Extends

Class "list", from data part.
Class "vector", by class "list".

Methods

**coerce** signature(from = "EuclRandVariable", to = "EuclRandVarList"): create a "EuclRandVarList" object from a Euclidean random variable.

**coerce** signature(from = "EuclRandMatrix", to = "EuclRandVarList"): create a "EuclRandVarList" object from a Euclidean random matrix.

**numberOfMaps** signature(object = "EuclRandVarList"): number of functions contained in the slots Map of the members of object.
**Dimension**

`signature(object = "EuclRandVarList")`: dimension of the Euclidean random variable.

**EvalRandVar**

`signature(RandVar = "EuclRandVarList", x = "numeric")`: evaluate the elements of RandVar at x.

`signature(RandVar = "EuclRandVarList", x = "matrix")`: evaluate the elements of RandVar at rows of x.

`signature(RandVar = "EuclRandVarList", x = "numeric", distr = "Distribution")`: evaluate the elements of RandVar at x assuming a probability space with distribution distr. In case x does not lie in the support of distr NA is returned.

`signature(RandVar = "EuclRandVarList", x = "matrix", distr = "Distribution")`: evaluate the elements of RandVar at rows of x assuming a probability space with distribution distr. For those rows of x which do not lie in the support of distr NA is returned.

**ImageDistr**

`signature(RandVar = "EuclRandVarList", distr = "Distribution")`: image distribution of distr under RandVar. Returns an object of class "DistrList".

**Show**

`signature(object = "EuclRandVarList")`: returns an object of class "EuclRandVarList" where the results of the functions in the slots Map of the members of x are transposed.

`signature(x = "EuclRandVarList")`: returns an object of class "EuclRandVarList" where the results of the functions in the slots Map of the members of x are transposed.

`signature(x = "EuclRandVarList", y = "EuclRandVarList")`: matrix multiplication for objects of class "EuclRandVarList". Generates an object of class "EuclRandMatrix".

`signature(x = "matrix", y = "EuclRandVarList")`: matrix multiplication of x and y. Generates an object of class "EuclRandMatrix".

`signature(x = "EuclRandVarList", y = "matrix")`: matrix multiplication of x and y. Generates an object of class "EuclRandMatrix".

**Arith**

`signature(e1 = "numeric", e2 = "EuclRandVarList")`: Given a numeric vector e1, a list of Euclidean random variables e2 and an arithmetic operator op, the list of Euclidean random variables e1 op e2 is returned.

`signature(e1 = "EuclRandVarList", e2 = "numeric")`: Given a numeric vector e2, a list of Euclidean random variables e1 and an arithmetic operator op, the list of Euclidean random variables e1 op e2 is returned.

`signature(e1 = "EuclRandVarList", e2 = "EuclRandVarList")`: Given two lists of Euclidean random variables e1, e2 and an arithmetic operator op, the list of Euclidean random variables e1 op e2 is returned.

**Math**

`signature(x = "EuclRandVarList")`: Given a "Math" group generic fct, the list of Euclidean random variables fct(x) is returned.

`signature(object = "UnivariateDistribution", fun = "EuclRandVarList", cond = "missing")`: expectation of fun under univariate distributions.

`signature(object = "AbscontDistribution", fun = "EuclRandVarList", cond = "missing")`: expectation of fun under absolutely continuous univariate distributions.

`signature(object = "DiscreteDistribution", fun = "EuclRandVarList", cond = "missing")`: expectation of fun under discrete univariate distributions.

`signature(object = "MultivariateDistribution", fun = "EuclRandVarList", cond = "missing")`: expectation of fun under multivariate distributions.
E signature(object = "DiscreteMVDistribution", fun = "EuclRandVarList", cond = "missing"):
expectation of fun under discrete multivariate distributions.

E signature(object = "UnivariateCondDistribution", fun = "EuclRandVarList", cond = "numeric"):
expectation of fun under conditional univariate distributions.

E signature(object = "AbscontCondDistribution", fun = "EuclRandVarList", cond = "numeric"):
expectation of fun under absolutely continuous conditional univariate distributions.

E signature(object = "DiscreteCondDistribution", fun = "EuclRandVarList", cond = "numeric"):
expectation of fun under discrete conditional univariate distributions.

Author(s)
Matthias Kohl <Matthias.Kohl@stamats.de>

See Also
EuclRandMatrix, RandVariable-class, EuclRandVariable-class, EuclRandMatrix-class, Distribution-class,
Arith, Math, E

Examples
L1 <- list(function(x){x}, function(x){x^2}, function(x){x^3}, function(x){x^4},
function(x){x^5}, function(x){x^6})
L2 <- list(function(x){exp(x)}, function(x){abs(x)},
function(x){sin(x)}, function(x){floor(x)})
R1 <- new("EuclRandVariable", Map = L2, Domain = Reals(), Range = Reals())
R2 <- EuclRandMatrix(Map = L1, ncol = 2, Domain = Reals(), dimension = 1)
R3 <- EuclRandMatrix(Map = L2, ncol = 2, Domain = Reals(), dimension = 1)
(RL1 <- new("EuclRandVarList", list(R1, R2, R3)))
dimension(RL1)
as(R1, "EuclRandVarList")
as(R2, "EuclRandVarList")

Map(exp(RL1)[[1]]) # "Math" group
## "Arith" group
Map((1 + RL1)[[1]])
Map((RL1 * 2)[[2]])
Map((RL1 / RL1)[[3]])

Optional rSpace-class  Optional rSpace

Description
Optional object of class "rSpace".
Objects from the Class

A virtual Class: No objects may be created from it.

Author(s)

Matthias Kohl <Matthias.Kohl@stamats.de>

See Also

rSpace-class

---

RandVariable Generating function for RandVariable-class

Description

Generates an object of class "RandVariable".

Usage

RandVariable(Map = list(function(x){}), Domain = NULL, Range = NULL)

Arguments

Map list of functions forming the map.
Domain domain of Map: object of class "OptionalrSpace" (default = NULL).
Range range of Map: object of class "OptionalrSpace" (default = NULL).

Value

Object of class "RandVariable"

Author(s)

Matthias Kohl <Matthias.Kohl@stamats.de>

See Also

RandVariable-class
Examples

```r
(R1 <- RandVariable())
Map(R1)
Domain(R1)
Range(R1)
Map(R1) <- list(function(x){ceiling(x)}, function(x){floor(x)})
Domain(R1) <- Reals()
Range(R1) <- Naturals()
R1
Map(R1)
length(R1)
```

R2 <- R1
Domain(R2) <- Naturals()
compatibleDomains(R1, R2)
Domain(R2) <- NULL
compatibleDomains(R1, R2)
Domain(R2) <- EuclideanSpace(dimension = 1)
compatibleDomains(R1, R2)
Domain(R2) <- EuclideanSpace(dimension = 2)
compatibleDomains(R1, R2)

```r
## The function is currently defined as
function(Map = list(function(x){ }), Domain = NULL, Range = NULL) {
  return(new("RandVariable", Map = Map, Domain = Domain, Range = Range))
}
```

---

**RandVariable-class**  
*Random variable*

**Description**

Class of random variables; i.e., measurable maps from Domain to Range. The elements contained in the list Map are functions in one(!) argument named “x”.

**Objects from the Class**

Objects can be created by calls of the form `new("RandVariable", ...)`. More frequently they are created via the generating function `RandVariable`.

**Slots**

- **Map**  
  Object of class "list": list of functions.

- **Domain**  
  Object of class "OptionalrSpace": domain of the random variable.

- **Range**  
  Object of class "OptionalrSpace": range of the random variable.
Methods

**Map** signature(object = "RandVariable"): accessor function for the slot Map.

**Domain** signature(object = "RandVariable"): accessor function for the slot Domain.

**Range** signature(object = "RandVariable"): accessor function for the slot Range.

**Map<-** signature(object = "RandVariable"): replacement function for the slot Map.

**Domain<-** signature(object = "RandVariable"): replacement function for the slot Domain.

**Range<-** signature(object = "RandVariable"): replacement function for the slot Range.

**compatibleDomains** signature(e1 = "RandVariable", e2 = "RandVariable"): test if the domains of two random variables are compatible.

**length** signature(object = "RandVariable"): length of the list of functions in slot Map.

**show** signature(object = "RandVariable")

Author(s)

Matthias Kohl <Matthias.Kohl@stamats.de>

See Also

RandVariable, EuclRandVariable-class, EuclRandMatrix-class, EuclRandVarList-class

Examples

```r
(R1 <- new("RandVariable"))
Map(R1)
Domain(R1)
Range(R1)
Map(R1) <- list(function(x){ceiling(x)}, function(x){floor(x)})
Domain(R1) <- Reals()
Range(R1) <- Naturals()
R1
Map(R1)
length(R1)

R2 <- R1
Domain(R2) <- Naturals()
compatibleDomains(R1, R2)
Domain(R2) <- NULL
compatibleDomains(R1, R2)
Domain(R2) <- EuclideanSpace(dimension = 1)
compatibleDomains(R1, R2)
Domain(R2) <- EuclideanSpace(dimension = 2)
compatibleDomains(R1, R2)
```
RealRandVariable  Generating function for RealRandVariable-class

Description

Generates an object of class "RealRandVariable".

Usage

RealRandVariable(Map = list(function(x) {1}), Domain = NULL, Range)

Arguments

Map  list of functions forming the map.
Domain  domain of Map: object of class "OptionalRSpace".
Range  range of Map: object of class "Reals".

Value

Object of class "RealRandVariable"

Author(s)

Matthias Kohl <Matthias.Kohl@stamats.de>

See Also

RealRandVariable-class

Examples

RealRandVariable(Map = list(function(x){x}), Domain = Reals())

## The function is currently defined as
function(Map = list(function(x){1}), Domain = NULL, Range) {
  if(missing(Range)) Range <- Reals()
  if(!is(Range, "Reals"))
    stop("'Range' has to be of class 'Reals'")

  return(new("RealRandVariable", Map = Map,
              Domain = Domain, Range = Reals()))
}
RealRandVariable-class

Real random variable

Description

Class of real random variables.

Objects from the Class

Objects can be created by calls of the form `new("RealRandVariable", ...)`. More frequently they are created via the generating function `EuclRandVariable`.

Slots

- Map Object of class "list": list of functions.
- Domain Object of class "OptionalrSpace": domain of the random variable.
- Range Object of class "Reals": range of the random variable.

Extends

Class "EuclRandVariable", directly.
Class "RandVariable", by class "EuclRandVariable".

Methods

- `Range<-` signature(object = "EuclRandVariable"): replacement function for the slot Range.

Author(s)

Matthias Kohl <Matthias.Kohl@stamats.de>

See Also

- EuclRandVariable-class

Examples

```r
new("RealRandVariable", Map=list(function(x){x}), Range = Reals())
```
Index

*Topic **arith**
  EuclRandMatrix-class, 4
  EuclRandVariable-class, 9

*Topic **classes**
  EuclRandMatrix, 3
  EuclRandMatrix-class, 4
  EuclRandVariable, 7
  EuclRandVariable-class, 9
  EuclRandVarList, 12
  EuclRandVarList-class, 13
  OptionalSpace-class, 15
  RandVariable, 16
  RandVariable-class, 17
  RealRandVariable, 19
  RealRandVariable-class, 20

*Topic **math**
  EuclRandMatrix-class, 4
  EuclRandVariable-class, 9

*Topic **package**
  RandVar-package, 2

[.EuclRandMatrix-method
  (EuclRandMatrix-class), 4

[.EuclRandVariable-method
  (EuclRandVariable-class), 9

%%,EuclRandMatrix,EuclRandMatrix-method
  (EuclRandMatrix-class), 4

%%,EuclRandMatrix,EuclRandVariable-method
  (EuclRandVariable-class), 9

%%,EuclRandMatrix,matrix-method
  (EuclRandMatrix-class), 4

%%,EuclRandMatrix,numeric-method
  (EuclRandMatrix-class), 4

%%,EuclRandVarList,matrix-method
  (EuclRandVarList-class), 13

%%,EuclRandVariable,EuclRandMatrix-method
  (EuclRandVariable-class), 9

%%,EuclRandVariable,EuclRandVariable-method
  (EuclRandVariable-class), 9

%%,EuclRandVariable,matrix-method
  (EuclRandVariable-class), 9

%%,EuclRandVariable,numeric-method
  (EuclRandVariable-class), 9

%%,matrix,EuclRandMatrix-method
  (EuclRandMatrix-class), 9

%%,matrix,EuclRandVarList-method
  (EuclRandVarList-class), 13

%%,matrix,EuclRandVariable-method
  (EuclRandVariable-class), 9

%%,numeric,EuclRandMatrix-method
  (EuclRandMatrix-class), 4

%%,numeric,EuclRandVariable-method
  (EuclRandVariable-class), 9

%m%,EuclRandVarList-class, 13

%m%,EuclRandVarList,EuclRandVarList-method
  (EuclRandVarList-class), 13

Arith, 6, 11, 15

Arith,EuclRandMatrix,EuclRandMatrix-method
  (EuclRandMatrix-class), 4

Arith,EuclRandMatrix,numeric-method
  (EuclRandMatrix-class), 4

Arith,EuclRandVariable,EuclRandVariable-method
  (EuclRandVariable-class), 9

Arith,EuclRandVariable,numeric-method
  (EuclRandVariable-class), 9

Arith,EuclRandVarList,EuclRandVarList-method
  (EuclRandVarList-class), 13

Arith,EuclRandVarList,numeric-method
  (EuclRandVarList-class), 13

Arith,numeric,EuclRandMatrix-method
  (EuclRandMatrix-class), 4

Arith,numeric,EuclRandVariable-method
  (EuclRandVariable-class), 9

Arith,numeric,EuclRandVarList-method
  (EuclRandVarList-class), 13

coe,EuclRandMatrix,EuclRandVarList-method
  (EuclRandMatrix-class), 4
coerce, EuclRandVariable, EuclRandMatrix-method, DiscreteDistribution, EuclRandVariable, missing-method
(EuclRandVariable-class), 9
coerce, EuclRandVariable, EuclRandVarList-method, DiscreteDistribution, EuclRandVarList, missing-method
(EuclRandVarList-class), 13
compatibleDomains (RandVariable-class), 17
compatibleDomains, RandVariable, RandVariable-method
(RandVariable-class), 17
Dim (EuclRandMatrix-class), 4
Dim, EuclRandMatrix-method
(EuclRandMatrix-class), 4
Dim<- (EuclRandMatrix-class), 4
Dim<-, EuclRandMatrix-method
(EuclRandMatrix-class), 4
dimension, EuclRandMatrix-method
(EuclRandMatrix-class), 4
dimension, EuclRandVariable-method
(EuclRandVariable-class), 9
dimension, EuclRandVarList-method
(EuclRandVarList-class), 13
Domain (RandVariable-class), 17
Domain, RandVariable-method
(RandVariable-class), 17
Domain<-, RandVariable-method
(RandVariable-class), 17

E, 6, 11, 15
E, AbscontCondDistribution, EuclRandMatrix, numeric-method
(EuclRandMatrix-class), 4
E, AbscontCondDistribution, EuclRandVariable, numeric-method
(EuclRandVariable-class), 9
E, AbscontCondDistribution, EuclRandVarList, numeric-method
(EuclRandVarList-class), 13
E, AbscontDistribution, EuclRandMatrix, missing-method
(EuclRandMatrix-class), 4
E, AbscontDistribution, EuclRandVariable, missing-method
(EuclRandVariable-class), 9
E, AbscontDistribution, EuclRandVarList, missing-method
(EuclRandVarList-class), 13
E, DiscreteCondDistribution, EuclRandMatrix, numeric-method
(EuclRandMatrix-class), 4
E, DiscreteCondDistribution, EuclRandVariable, numeric-method
(EuclRandVariable-class), 9
E, DiscreteCondDistribution, EuclRandVarList, numeric-method
(EuclRandVarList-class), 13
E, DiscreteDistribution, EuclRandMatrix, missing-method
(EuclRandMatrix-class), 4
E, DiscreteDistribution, EuclRandVariable, missing-method
(EuclRandVariable-class), 9
E, DiscreteDistribution, EuclRandVarList, missing-method
(EuclRandVarList-class), 13
E, DiscreteMVDistribution, EuclRandMatrix, missing-method
(EuclRandMatrix-class), 4
E, DiscreteMVDistribution, EuclRandVarList, missing-method
(EuclRandVarList-class), 13
E, MultivariateDistribution, EuclRandMatrix, missing-method
(EuclRandMatrix-class), 4
E, MultivariateDistribution, EuclRandVariable, missing-method
(EuclRandVariable-class), 9
E, MultivariateDistribution, EuclRandVarList, missing-method
(EuclRandVarList-class), 13
E, UnivariateCondDistribution, EuclRandMatrix, numeric-method
(EuclRandMatrix-class), 4
E, UnivariateCondDistribution, EuclRandVariable, numeric-method
(EuclRandVariable-class), 9
E, UnivariateCondDistribution, EuclRandVarList, numeric-method
(EuclRandVarList-class), 13
E, UnivariateDistribution, EuclRandMatrix, missing-method
(EuclRandMatrix-class), 4
E, UnivariateDistribution, EuclRandVariable, missing-method
(EuclRandVariable-class), 9
E, UnivariateDistribution, EuclRandVarList, missing-method
(EuclRandVarList-class), 13
E, UnivariateMVDistribution, EuclRandMatrix, missing-method
(EuclRandMatrix-class), 4
E, UnivariateMVDistribution, EuclRandVarList, missing-method
(EuclRandVarList-class), 13
E, Discretemvdistribution, EuclRandMatrix, missing-method
(EuclRandMatrix-class), 4
E, Discretemvdistribution, EuclRandVariable, missing-method
(EuclRandVariable-class), 9
E, Discretemvdistribution, EuclRandVarList, missing-method
(EuclRandVarList-class), 13

(EuclRandVariable-class), 9
Range, RandVariable-method
evalRandVar, EuclRandVariable, numeric, missing-method (RandVariable-class), 17
(EuclRandVariable-class), 9
Range<- (RandVariable-class), 17
evalRandVar, EuclRandVarList, matrix, Distribution-method (EuclRandVariable-class), 13
(EuclRandVarList-class), 13
EvalRandVar, EuclRandVarList, matrix, missing-method (EuclRandVariable-class), 9
(EuclRandVarList-class), 13
evalRandVar, EuclRandVarList, numeric, Distribution-method (EuclRandVariable-class), 17
(EuclRandVarList-class), 13
evalRandVar, EuclRandVarList, numeric, missing-method (EuclRandVariable-class), 19
(EuclRandVarList-class), 13
RealRandVariable-class, 20

imageDistr (EuclRandVariable-class), 9
show, EuclRandMatrix-method
imageDistr, EuclRandVariable, Distribution-method (EuclRandMatrix-class), 4
show, EuclRandVarList-method
imageDistr, EuclRandVarList, Distribution-method (EuclRandVarList-class), 13
show, RandVariable-method (RandVariable-class), 17

length, RandVariable-method
(RandVariable-class), 17

Map (RandVariable-class), 17
Map, RandVariable-method (RandVariable-class), 17
Map<- (RandVariable-class), 17
Map<-, RandVariable-method (RandVariable-class), 17

Math, 6, 11, 15
Math, EuclRandMatrix-method
(EuclRandMatrix-class), 4
Math, EuclRandVariable-method (EuclRandVariable-class), 9
Math, EuclRandVarList-method (EuclRandVarList-class), 13

ncol, EuclRandMatrix-method (EuclRandMatrix-class), 4
nrow, EuclRandMatrix-method (EuclRandMatrix-class), 4
numberOfMaps (EuclRandVarList-class), 13
numberOfMaps, EuclRandVarList-method (EuclRandVarList-class), 13

OptionalSpace-class, 15

RandVar (RandVar-package), 2
RandVar-package, 2
RandVariable, 16, 18
RandVariable-class, 17
Range (RandVariable-class), 17