Package ‘discreteRV’

February 19, 2015

Maintainer  Eric Hare <erichare@iastate.edu>
Author       Andreas Buja
Version      1.1.4
Date         2015-02-02
License      GPL-3
Title        Functions to Create and Manipulate Discrete Random Variables
Description  discreteRV implements a set of functions that allow computations involving discrete random variables. It uses a syntax which is familiar to that which is used in mathematical statistics and probability courses.
Imports      plyr, MASS
Suggests     testthat, roxygen2
NeedsCompilation no
Repository    CRAN
Date/Publication  2015-02-03 06:38:33

R topics documented:

as.RV                     2
Conditional               2
E                        3
KURT                      4
make.RV                  4
margins                  5
mult                     6
multN                    6
outcomes                 7
P                        8
plot.RV                  8
plot.RVsim               9
print.RV                 10
probs                    10
Prop                     11
as.RV

Turn a probability vector with possible outcome values in the 'names()' attribute into a random variable:

Description

Turn a probability vector with possible outcome values in the 'names()' attribute into a random variable:

Usage

as.RV(px, fractions = TRUE)

Arguments

px A probability vector with possible outcome values in the 'names()' attribute
fractions If TRUE, return the probabilities as fractions

Conditional

Compute the conditional probability of two events

Description

Compute the conditional probability of two events

Usage

## S3 method for class 'RVresult'
vec1 | vec2
Arguments

vec1  an RVresult object
vec2  an RVresult object

Value

An RVcond object representing the conditional probability

Examples

xNfairNdie <- make.RV(1:6, rep(1/6, 6))
x.Nfair.coin <- make.RV(1:2, rep(1/2, 2))

P(X.fair.die == 4 | X.fair.die > 3)
P(X.fair.die == 5 | X.fair.die < 5)
P(X.fair.die == 4 | X.fair.coin == 1) # Independence

\[
E \quad \text{Expected value of a random variable}
\]

Description

Expected value of a random variable

Usage

E(X)

Arguments

X  random variable

Examples

X.Bern <- make.RV(c(1,0), c(.5,.5))
E(X.Bern)

X.fair.die <- make.RV(1:6, rep(1/6, 6))
E(X.fair.die)
KURT

Kurtosis of a random variable

Description
Kurtosis of a random variable

Usage
KURT(X)

Arguments
X random variable

Examples
X.Bern <- make.RV(c(1,0), c(.5,.5))
KURT(X.Bern)

make.RV

Make a random variable consisting of possible outcome values and their probabilities or odds

Description
Make a random variable consisting of possible outcome values and their probabilities or odds

Usage
make.RV(outcomes, probs = NULL, odds = NULL, fractions = (class(probs) != "function"), range = any(is.infinite(outcomes)), verifyprobs = TRUE, ...)

Arguments
outcomes Vector of possible outcomes
probs Vector of probabilities or function defining probabilities
odds Vector of odds
fractions If TRUE, return the probabilities as fractions when printing
range If TRUE, outcomes specify a range of values in the form c(lower, upper)
verifyprobs If TRUE, verify that the probs sum to one
... Additional parameters passed to the function defining outcome probabilities
Value
random variable as RV object.

Examples

# Make a 50:50 Bernoulli random variable:
X.Bern <- make.RV(c(1,0), c(0.5,0.5))

# Make a fair coin flip game with payoffs +$1 and -$1:
X.fair.coin <- make.RV(c(1,-1), c(0.5,0.5))

# Make a biased coin flip game with odds 1:2 and with fair payoffs +$2 and -$1:
X.biased.coin <- make.RV(c(2,-1), odds = c(1,2))

# Make a fair die
X.fair.die <- make.RV(1:6, 1/6)

# Make a loaded die, specifying odds 1:1:1:2:4 rather than probabilities:
X.loaded.die <- make.RV(1:6, odds = c(1,1,1,1,2,4))

# Make a Poisson random variable
pois.func <- function(x, lambda) { lambda^x * exp(-lambda) / factorial(x) }
X.poiis <- make.RV(c(0, Inf), pois.func, lambda = 5)

margins
Marginal distribution of a joint random variable

Description
Extracts the marginal probability mass functions from a joint distribution.

Usage
margins(X, sep = "",")

Arguments
X a random variable
sep parameter specifying the separator between dimensions, defaults to ","

Author(s)
Heike Hofmann <hofmann@iastate.edu>

Examples
X <- make.RV(1:6, 1/6)
X3 <- multN(X, 3)
margins(X3)
**mult**

*Joint probability mass function of random variables X and Y*

**Description**

Joint probability mass function of random variables X and Y

**Usage**

```
mult(X, Y, sep = " ", fractions = (attr(X, "fractions") & attr(Y, "fractions")))
```

**Arguments**

- **X**: random variable
- **Y**: random variable
- **sep**: separator between items from marginal distributions, by default set to ",,"
- **fractions**: If TRUE, return the probabilities as fractions

**Author(s)**

Heike Hofmann <hofmann@iastate.edu>

**Examples**

```
d <- make.RV(c("A","B","C"), odds = c(3,5,11))
d2 <- mult(d,d)
probs(d2)
```

---

**multN**

*Probability mass function of X^n*

**Description**

Probability mass function of X^n

**Usage**

```
multN(X, n = 2, sep = ",", fractions = attr(X, "fractions"))
```

**Arguments**

- **X**: random variable
- **n**: power
- **sep**: separator between items from marginal distributions, by default set to ",,"
- **fractions**: If TRUE, return the probabilities as fractions
Author(s)
Heike Hofmann <hofmann@iastate.edu>

Examples

d <- make.RV(c("A","B","C"), odds = c(3,5,11))
d2 <- multN(d)
probs(d2)

outcomes

| outcomes | Outcomes of random variable X |

Description
Obtain the list of outcomes from a random variable

Usage
outcomes(X)

Arguments

| X | random variable |

Value
vector of outcomes of X

Examples

X.Bern <- make.RV(c(1,0), c(.5,.5))
outcomes(X.Bern)

X.fair.die <- make.RV(1:6, rep(1/6,6))
outcomes(X.fair.die)

X.loaded.die <- make.RV(1:6, odds = c(1,1,1,1,2,4))
outcomes(X.loaded.die)
P  

Calculate probabilities of events

Description

Calculate probabilities of events

Usage

\[ P(\text{event}) \]

Arguments

event  
A logical vector

Examples

\begin{verbatim}
X.fair.die <- make.RV(1:6, rep(1/6,6))
P(X.fair.die>3)

X.loaded.die <- make.RV(1:6, odds = c(1,1,1,1,2,4))
P(X.loaded.die>3)
P(X.loaded.die==6)
\end{verbatim}

plot.RV  

Plot a random variable of class "RV"

Description

Plot a random variable of class "RV"

Usage

\#
## S3 method for class 'RV'
plot(x, ..., tol = 1e-10, pch = 16, cex = 1.2, lwd = 2,
col = "black", xlab = "Possible Values", ylab = "Probabilities")

Arguments

x  
A random variable

...  
Additional arguments to be passed to the "plot" function

tol  
Only display outcomes with probabilities above tol

pch  
Either an integer specifying a symbol or a single character to be used as the default in plotting points.
**plot.RVsim**

- **cex**: A numerical value giving the amount by which plotting text and symbols should be magnified relative to the default.
- **lwd**: The line width, a positive number, defaulting to 2.
- **col**: A specification for the default plotting color
- **xlab**: Label for the X axis
- **ylab**: Label for the Y axis

**Examples**

```r
fair.die <- make.RV(1:6, rep(1/6, 6))
plot(fair.die)
```

---

**Description**

Plot a simulated random vector

**Usage**

```r
## S3 method for class 'RVsim'
plot(x, ...)
```

**Arguments**

- **x**: A simulated data vector produced with the 'rsim()' function
- **...**: Additional arguments to be passed to the 'plot()' function

**Examples**

```r
X <- make.RV(c(100000, 10000, 0), c(0.00025, 0.005, 0.99475))
X.sim <- rsim(200000, X)
plot(X.sim)
```
### print.RV

*Print a random variable of class "RV"*

**Description**

Print a random variable of class "RV"

**Usage**

```r
## S3 method for class 'RV'
print(x, odds = attr(x, "odds"), fractions = attr(x, "fractions"),
      all.outcomes = FALSE, digits = 3, ...)
```

**Arguments**

- `x`: A random variable
- `odds`: If TRUE, print as odds instead of probs
- `fractions`: If TRUE, print probs as fractions instead of decimals
- `all.outcomes`: If TRUE, print all outcomes rather than the first ten
- `digits`: Number of digits to print for probabilities
- `...`: Additional arguments to be passed to the "format" function

**Author(s)**

Eric Hare <erichare@iastate.edu>

**Examples**

```r
fair.die <- make.RV(1:6, rep(1/6,6))
print(fair.die)
```

### probs

*Probability mass function of random variable X*

**Description**

Obtain the list of probabilities from a random variable: p(x)

**Usage**

`probs(X)`

**Arguments**

- `X`: random variable
**Value**

named vector of probabilities for each element of the random variable

**Examples**

```r
X.Bern <- make.RV(c(1,0), c(.5,.5))
probs(X.Bern)

X.fair.die <- make.RV(1:6, rep(1/6,6))
probs(X.fair.die)

X.loaded.die <- make.RV(1:6, odds = c(1,1,1,1,2,4))
probs(X.loaded.die)
```

**Description**

Proportion of an event observed in a vector of simulated trials

**Usage**

```r
Prop(X.sim)
```

**Arguments**

- `X.sim` A simulated data vector produced with the `rsim()` function

**Examples**

```r
X <- make.RV(c(100000,10000,0), c(0.00025,0.005,0.99475))
X.sim <- rsim(200000, X)

Prop(X.sim>0)
Prop(X.sim==100000)
Prop(X.sim==2000)
```
props

Description
Proportions of observed outcomes in one or more vectors of simulated trials

Usage
props(...)

Arguments
...

Simulation data produced with the 'rsim()' function

Examples
X.Bern <- make.RV(c(1,0), c(.5,.5))
X.Bern.sim100 <- rsim(100, X.Bern)

X.loaded.die <- make.RV(1:6, odds = c(1,1,1,1,2,4))
X.loaded.die.sim100 <- rsim(100, X.loaded.die)
props(X.Bern.sim100)
props(X.loaded.die.sim100)

# Note: 'props()' is the analog of 'probs()', but
# 'props()' applies to SIMULATION DATA and tabulates them, whereas
# 'probs()' applies to RANDOM VARIABLES and lists their probabilities.
# By the LLN the results of 'props()' will be close to 'probs()' for
# for large simulations.

qqnorm.RV

Normal quantile plot for RVs to answer the question how close to normal it is

Description
Normal quantile plot for RVs to answer the question how close to normal it is

Usage
## S3 method for class 'RV'
qqnorm(y, ..., pch = 16, cex = 0.5, add = FALSE,
xlab = "Normal Quantiles", ylab = "Random Variable Quantiles",
tol = 1e-10)
Arguments

- **y**
  - A random variable
- **...**
  - Additional arguments to be passed to the "plot" or "points" function
- **pch**
  - Either an integer specifying a symbol or a single character to be used as the default in plotting points.
- **cex**
  - A numerical value giving the amount by which plotting text and symbols should be magnified relative to the default.
- **add**
  - A logical indicating whether to add to an existing plot
- **xlab**
  - Label for the X axis
- **ylab**
  - Label for the Y axis
- **tol**
  - Tolerance for the zero probability case

Examples

```r
fair.die <- make.RV(1:6, rep(1/6,6))
qqnorm(fair.die)
```

---

**rsim**

*Simulate n independent trials from a random variable X:*

**Description**

Simulate n independent trials from a random variable X:

**Usage**

`rsim(n, X)`

**Arguments**

- **n**
  - The number of independent trials to simulate
- **X**
  - A random variable

**Examples**

```r
X.Bern <- make.RV(c(1,0), c(.5,.5))
X.Bern.sim100 <- rsim(100, X.Bern)

X.loaded.die <- make.RV(1:6, odds = c(1,1,1,1,2,4))
X.loaded.die.sim100 <- rsim(100, X.loaded.die)
```

# The function 'rsim()' attaches the probabilities as names to the random draws.
# To get the values only, use 'as.vector()':

```r
as.vector(X.Bern.sim100)
as.vector(X.loaded.die.sim100)
```
**SD**  
*Standard deviation of a random variable*

**Description**
Standard deviation of a random variable

**Usage**
SD(X)

**Arguments**
X  random variable

**Examples**
X.Bern <- make.RV(c(1, 0), c(.5, .5))  
E(X.Bern)

---

**SKEW**  
*Skewness of a random variable*

**Description**
Skewness of a random variable

**Usage**
SKEW(X)

**Arguments**
X  random variable

**Examples**
X.Bern <- make.RV(c(1, 0), c(.5, .5))  
SKEW(X.Bern)
**skewSim**

*Skew of the empirical distribution of simulated data*

**Description**

Skew of the empirical distribution of simulated data

**Usage**

```r
skewSim(x.sim)
```

**Arguments**

- `x.sim`: A simulated data vector produced with the `rsim()` function

**Examples**

```r
x <- make.RV(c(100000,10000,0), c(0.00025,0.005,0.99475))
x.sim <- rsim(200000, x)
skewSim(x.sim)
```

**SofI**

*Sum of independent random variables*

**Description**

Sum of independent random variables

**Usage**

```r
SofI(..., fractions = attr(list(...)[[1]], "fractions"))
```

**Arguments**

- `...`: Arbitrary number of random variables
- `fractions`: If TRUE, return the probabilities as fractions

**Examples**

```r
X.Bern <- make.RV(c(1,0), c(.5,.5))
X.fair.die <- make.RV(1:6, rep(1/6,6))

S.mix <- SofI(X.Bern, X.fair.die)  # Independent but not IID
```
SofIID

**Sum of independent identically distributed random variables**

**Description**

Sum of independent identically distributed random variables

**Usage**

\[ \text{SofIID}(X, n = 2, \text{progress} = \text{TRUE}, \text{fractions} = \text{attr}(X, "fractions")) \]

**Arguments**

- **x**: A random variable
- **n**: The number of Xs to sum
- **progress**: If TRUE, display a progress bar
- **fractions**: If TRUE, return the probabilities as fractions

**Examples**

```r
X.Bern <- make.RV(c(1,0), c(.5,.5))
S5 <- SofIID(X.Bern, 5)
S128 <- SofIID(X.Bern, 128)
```

V

**Variance of a random variable**

**Description**

Variance of a random variable

**Usage**

\[ V(X) \]

**Arguments**

- **x**: random variable

**Examples**

```r
X.Bern <- make.RV(c(1,0), c(.5,.5))
E(X.Bern)
```
Compute the logical AND of two events

Description
Compute the logical AND of two events

Usage
\[ x \ %\text{AND}\% \ y \]

Arguments
\begin{itemize}
  \item \textbf{x} \quad \text{RVcond object}
  \item \textbf{y} \quad \text{RVcond object}
\end{itemize}

Value
An RVresult object which is two events ANDed together

Examples
\begin{verbatim}
X.fair.die <- make.RV(1:6, rep(1/6,6))
P((X.fair.die == 4 \ %\text{AND}\% \ (X.fair.die == 3))
\end{verbatim}

Compute the logical OR of two events

Description
Compute the logical OR of two events

Usage
\[ x \ %\text{OR}\% \ y \]

Arguments
\begin{itemize}
  \item \textbf{x} \quad \text{RVcond object}
  \item \textbf{y} \quad \text{RVcond object}
\end{itemize}

Value
An RVresult object which is two events ORed together
Examples

X.fair.die <- make.RV(1:6, rep(1/6, 6))
P((X.fair.die == 4) %in% (X.fair.die == 3))

%in%  

Generic method for in operator function

Description

Generic method for in operator function

Usage

e1 %in% e2

Arguments

e1  First vector
e2  Second vector

Value

A logical vector indicating which elements of e1 are in e2
Index

%AND%, 17
%OR%, 17
%in%, 18

as.RV, 2

Conditional, 2

E, 3

KURT, 4

make.RV, 4
make.RV-package (make.RV), 4
margins, 5
mult, 6
multN, 6

outcomes, 7

P, 8
plot.RV, 8
plot.RVsim, 9
print.RV, 10
probs, 10
Prop, 11
props, 12

qqnorm.RV, 12

rsim, 13

SD, 14
SKEW, 14
skewSim, 15
SofI, 15
SofIID, 16

V, 16