Package ‘discreteRV’

September 16, 2015

Title Create and Manipulate Discrete Random Variables

Version 1.2.2

Description Create, manipulate, transform, and simulate from discrete random variables. The syntax is modeled after that which is used in mathematical statistics and probability courses, but with powerful support for more advanced probability calculations. This includes the creation of joint random variables, and the derivation and manipulation of their conditional and marginal distributions.

URL https://github.com/erichare/discreteRV

BugReports https://github.com/erichare/discreteRV/issues

Depends R (>= 3.0.2)

Imports plyr, MASS

Suggests testthat, roxygen2, knitr

License GPL-3

LazyData true

VignetteBuilder knitr

NeedsCompilation no

Author Andreas Buja [aut],
    Eric Hare [aut, cre],
    Heike Hofmann [aut]

Maintainer Eric Hare <erichare@iastate.edu>

Repository CRAN

Date/Publication 2015-09-16 10:05:21

R topics documented:

as.RV .......................................................... 2
E .............................................................. 3
iid .............................................................. 3
independent .................................................. 4
**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**

```r
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
**Expected value of a random variable**

**Description**

Expected value of a random variable

**Usage**

\[ E(X) \]

**Arguments**

- \( X \) random variable

**Examples**

\[
\begin{align*}
X.\text{Bern} & \leftarrow \text{RV}(c(1,0), c(0.5,0.5)) \\
E(X.\text{Bern}) & \\
X.\text{fair.die} & \leftarrow \text{RV}(1:6, \text{rep}(1/6,6)) \\
E(X.\text{fair.die}) &
\end{align*}
\]

**iid**

*Probability mass function of X^n*

**Description**

Probability mass function of \( X^n \)

**Usage**

\[
\text{iid}(X, n = 2, \text{sep} = ",", \text{fractions} = \text{attr}(X, "fractions"))
\]

**Arguments**

- \( X \) random variable
- \( n \) power
- \( \text{sep} \) separator between items from marginal distributions, by default set to ","
- \( \text{fractions} \) If TRUE, return the probabilities as fractions

**Author(s)**

Heike Hofmann <hofmann@iastate.edu>
**Examples**

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

---

**independent**

Tests whether the random variables X and Y are independent

---

**Description**

Tests whether the random variables X and Y are independent

**Usage**

```r
independent(X, Y)
```

**Arguments**

- **X**
  - A random variable
- **Y**
  - A random variable

**Author(s)**

Eric Hare <erichare@iastate.edu>

**Examples**

```r
AandB <- jointRV(outcomes = list(1:3, 0:2), probs = 1:9 / sum(1:9))
A <- marginal(AandB, 1)
B <- marginal(AandB, 2)
independent(A, B) # FALSE
CandD <- jointRV(outcomes = list(1:3, 0:2))
C <- marginal(CandD, 1)
D <- marginal(CandD, 2)
independent(C, D) # FALSE
```
joint

Joint probability mass function of random variables X and Y

Description

Joint probability mass function of random variables X and Y

Usage

\[ \text{joint}(X, Y, \text{sep} = "", \text{fractions} = (\text{attr}(X, "fractions") \& \text{attr}(Y, "fractions"))) \]

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

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

jointrv

Make a joint random variable consisting

Description

Make a joint random variable consisting

Usage

\[ \text{jointrv}(\text{outcomes, probs=} \text{NULL, ...}) \]

Arguments

- **outcomes**: The possible outcomes of the joint random variable, as a list
- **probs**: The probabilities of each event, in the order (x1, y1, x1, y2, ..., x2, y1, x2, y2, ..., xn, yn)
- **...**: Further arguments to be passed to the RV function
Value
An RV object

KURT
Kurtosis of a random variable

Description
Kurtosis of a random variable

Usage
KURT(X)

Arguments
X random variable

Examples
X.Bern <- RV(c(1,0), c(0.5,0.5))
KURT(X.Bern)

marginal
Marginal distribution of a joint random variable

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

Usage
marginal(X, num)

Arguments
X A random variable
num Number indicating which marginal distribution to extract

Author(s)
Eric Hare <erichare@iastate.edu>

Examples
AandB <- jointRV(outcomes = list(1:3, 0:2), probs = 1:9 / sum(1:9))
marginal(AandB, 1)
marginal(AandB, 2)
margins

Marginal distributions of a joint random variable

Description

Extracts the marginal probability mass functions from a joint distribution.

Usage

margins(xL sep ] BLBI

Arguments

x a random variable
sep parameter specifying the separator between dimensions, defaults to ",".

Author(s)

Heike Hofmann <hofmann@iastate.edu>

Examples

x <M rv(Q:VL QOVI
xS <M iid(xL SI
margins(xSI

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
plot.RV

Examples

X.Bern <- RV(c(1,0), c(0.5,0.5))
outcomes(X.Bern)

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

X.loaded.die <- 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(event)

Arguments

event A logical vector

Examples

X.fair.die <- RV(1:6, rep(1/6,6))
P(X.fair.die>3)

X.loaded.die <- RV(1:6, odds = c(1,1,1,1,2,4))
P(X.loaded.die>3)
P(X.loaded.die==6)

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.
- **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 <- 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 <- RV(c(100000,100000,0), c(0.00025,0.005,0.99475))
X.sim <- rsim(X, 200000)
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 <- 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**

```r
probs(x)
```

**Arguments**

- `x` random variable
**Value**

named vector of probabilities for each element of the random variable

**Examples**

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

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

X.loaded.die <- 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 <- RV(c(100000,10000,0), c(0.00025,0.005,0.99475))
X.sim <- rsim(x, 200000)

Prop(X.sim>0)
Prop(X.sim==100000)
Prop(X.sim==20000)
```
props

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

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

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

fair.die <- 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(X, n)

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

Examples

X.Bern <- RV(c(1,0), c(.5,.5))
X.Bern.sim100 <- rsim(X.Bern, 100)
X.loaded.die <- RV(1:6, odds = c(1,1,1,1,2,4))
X.loaded.die.sim100 <- rsim(X.loaded.die, 100)

# The function 'rsim()' attaches the probabilities as names to the random draws.
# To get the values only, use 'as.vector()':
as.vector(X.Bern.sim100)
as.vector(X.loaded.die.sim100)
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**

```r
RV(outcomes, probs = NULL, odds = NULL, fractions = (class(probs) != "function"), range = any(is.infinite(outcomes)), verifyprobs = TRUE, id = rnorm(1), ...)
```

**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
- **id**: Set the id of the random variable
- **...**: Additional parameters passed to the function defining outcome probabilities

**Value**

random variable as RV object.

**Examples**

```r
# Make a 50:50 Bernoulli random variable:
X.Bern <- RV(c(1,0), c(.5,.5))

# An equivalent method
X.Bern <- RV("bernoulli")

# Make a fair coin flip game with payoffs +$1 and -$1:
X.coin <- RV(c(1,-1), c(.5,.5))

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

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

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

# An equivalent method
X.pois <- RV("poisson")

---

### SD

**Standard deviation of a random variable**

**Description**

Standard deviation of a random variable

**Usage**

SD(X)

**Arguments**

X random variable

**Examples**

X.Bern <- 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 <- 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**

```
skewSim(x_sim)
```

**Arguments**

- `x_sim`: A simulated data vector produced with the `rsim()` function

**Examples**

```
X <- RV(c(100000,10000,0), c(0.00000,0.00005,0.99975))
X.sim <- rsim(X, 200000)

skewSim(X.sim)
```

Sofi  

*Sum of independent random variables*

**Description**

Sum of independent random variables

**Usage**

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

**Arguments**

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

**Examples**

```
X.Bern <- RV(c(1,0), c(0.5,.5))
X.fair.die <- RV(1:6, rep(1/6,6))

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

**Sum of independent identically distributed random variables**

**Description**

Sum of independent identically distributed random variables

**Usage**

SoIID(X, n = 2, progress = TRUE, fractions = 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 <- RV(c(1,0), c(.5,.5))
S5  <- SoIID(X.Bern, 5)
S128 <- SoIID(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 <- RV(c(1,0), c(.5,.5))
E(X.Bern)
```
%AND%  
*Compute the logical AND of two events*

**Description**
Compute the logical AND of two events

**Usage**

```
x %AND% y
```

**Arguments**

- **x**  RVcond object
- **y**  RVcond object

**Value**
An RVresult object which is two events ANDed together

**Examples**

```
x.fair.die <- RV(1:6, rep(1/6, 6))
P((x.fair.die == 4) %AND% (x.fair.die == 3))
```

%OR%  
*Compute the logical OR of two events*

**Description**
Compute the logical OR of two events

**Usage**

```
x %OR% y
```

**Arguments**

- **x**  RVcond object
- **y**  RVcond object

**Value**
An RVresult object which is two events ORed together
Examples

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

---

%in%

Generic method for in operator function

Description

Generic method for in operator function

Usage

```r
e1 %in% e2
```

Arguments

- `e1`: First vector
- `e2`: Second vector

Value

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

%AND%, 18
%OR%, 18
%in%, 19

as.RV, 2
E, 3

iid, 3
independent, 4

joint, 5
jointRV, 5

KURT, 6

marginal, 6
margins, 7

outcomes, 7

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

qqnorm.RV, 12

rsim, 13
RV, 14
RV-package (RV), 14

SD, 15
SKEW, 15
skewSim, 16
SofI, 16
SofIID, 17

V, 17