

Package ‘TruncExpFam’

October 12, 2022

Title Truncated Exponential Family

Version 1.0.0

Date 2022-02-21

Description Handles truncated members from the exponential family of probability distributions. Contains functions such as `rtruncnorm()` and `dtruncpois()`, which are truncated versions of `rnorm()` and `dpois()` from the `stats` package that also offer richer output containing, for example, the distribution parameters. It also provides functions to retrieve the original distribution parameters from a truncated sample by maximum-likelihood estimation.

License GPL-3

Encoding UTF-8

RoxygenNote 7.1.2

Imports methods, invgamma, rmutil

Suggests testthat

URL <https://github.com/ocbe-uio/TruncExpFam>

BugReports <https://github.com/ocbe-uio/TruncExpFam/issues>

NeedsCompilation no

Author René Holst [aut],
Waldir Leoncio [cre, aut]

Maintainer Waldir Leoncio <w.l.netto@medisin.uio.no>

Repository CRAN

Date/Publication 2022-02-24 09:40:05 UTC

R topics documented:

<code>.onAttach</code>	2
<code>dtrunc</code>	2
<code>genrtruncClass</code>	3
<code>init.parms</code>	4

mlEstimationTruncDist	5
natural2parameters	7
parameters2natural	7
print.trunc	8
rtrunc	8
TruncExpFam	11
validateFamilyParms	12

Index 13

`.onAttach` *Prints welcome message on package load*

Description

Prints package version number and welcome message on package load

Usage

```
.onAttach(libname, pkgname)
```

Arguments

<code>libname</code>	library location. See <code>?base::.onAttach</code> for details
<code>pkgname</code>	package name. See <code>?base::.onAttach</code> for details

`dtrunc` *Probability Density Function*

Description

Calculates the PDF for a given truncated distribution

Usage

```
dtrunc(y, eta, a, b)
```

```
dtruncbeta(y, eta, a = 0, b = 1)
```

```
dtruncbinom(y, eta, a = 0, b = attr(y, "parameters")$size, ...)
```

```
dtruncchisq(y, eta, a = 0, b = Inf)
```

```
dtrunccontbern(y, eta, a = 0, b = 1)
```

```
dtruncexp(y, eta, a = 0, b = Inf)
```

```

dtruncgamma(y, eta, a = 0, b = Inf)
dtruncinvgamma(y, eta, a = 0, b = Inf)
dtruncinvgauss(y, eta, a = 0, b = Inf)
dtrunclnorm(y, eta, a = 0, b = Inf)
dtruncbinom(y, eta, a = 0, b = Inf, ...)
dtruncnorm(y, eta, a = -Inf, b = Inf)
dtruncpois(y, eta, a = 0, b = Inf)

```

Arguments

y	output from rtrunc or any valid numeric value(s).
eta	Natural parameters
a	lower truncation limit
b	upper truncation limit
...	size

Value

The density of y for the given values of the eta parameter.

Examples

```

# Using the output of rtrunc
y <- rtrunc(50, mean = 5, sd = 2)
dtrunc(y, eta = c(0, -1))

# Directly-inputting values
dtruncnorm(y = c(5, 0, -10), eta = c(0, -0.05))

```

genrtruncClass	<i>Generates an rtrunc-dispatchable class</i>
----------------	---

Description

Matches a list of arguments to an rtrunc method

Usage

```
genrtruncClass(n, family, parms)
```

Arguments

n	sample size
family	distribution family
parms	list of parameters passed to rtrunc (through the ... element)

Value

A character string.

Author(s)

Waldir Leoncio

init.parms	<i>Initialize parameters</i>
------------	------------------------------

Description

Returns the empirical parameter estimate for a distribution

Usage

```
init.parms(y)
```

Arguments

y	output of rtrunc
---	------------------

Value

A vector of parameter estimates for the input sample

Examples

```
# Normal distribution
sampNorm <- rtrunc(50, mean = 5, sd = 2)
init.parms(sampNorm)

# Poisson distribution
sampPois <- rtrunc(10, lambda = 100, family = "Poisson")
init.parms(sampPois)
```

 mlEstimationTruncDist *ML Estimation of Distribution Parameters*

Description

ML-estimation of the parameters of the distribution of the specified family, truncated at `y.min` and `y.max`

Usage

```
mlEstimationTruncDist(
  y,
  y.min = attr(y, "truncation_limits")$a,
  y.max = attr(y, "truncation_limits")$b,
  tol = 1e-05,
  max.it = 25,
  delta = 0.33,
  print.iter = 0,
  ny = 100,
  ...
)
```

Arguments

<code>y</code>	Sequence spanning the domain of the truncated distribution
<code>y.min</code>	Lower bound for <code>y</code>
<code>y.max</code>	Upper bound for <code>y</code>
<code>tol</code>	Error tolerance for parameter estimation
<code>max.it</code>	Maximum number of iterations
<code>delta</code>	Indirectly, the difference between consecutive iterations to compare with the error tolerance
<code>print.iter</code>	Determines the frequency of printing (i.e., prints every <code>print.iter</code> iterations)
<code>ny</code>	size of intermediate <code>y</code> range sequence. Higher values yield better estimations but slower iterations
<code>...</code>	other parameters passed to subfunctions

Value

A vector of class `trunc_*` containing the maximum-likelihood estimation of the underlying distribution * parameters.

Note

`print.iter` can be `TRUE`, `FALSE` or an integer indicating an interval for printing every `X` iterations.

Author(s)

René Holst

References

Inspired by Salvador: Pueyo: "Algorithm for the maximum likelihood estimation of the parameters of the truncated normal and lognormal distributions"

Examples

```
sample_size <- 1000
# Normal
sample.norm <- rtrunc(n = sample_size, mean = 2, sd = 1.5, a = -1)
mlEstimationTruncDist(
  sample.norm,
  y.min = -1, max.it = 500, delta = 0.33,
  print.iter = TRUE
)

# Log-Normal
sample.lognorm <- rtrunc(
  n = sample_size, family = "lognormal", meanlog = 2.5, sdlog = 0.5, a = 7
)
ml_lognormal <- mlEstimationTruncDist(
  sample.lognorm,
  y.min = 7, max.it = 500, tol = 1e-10, delta = 0.3,
  print.iter = FALSE
)
ml_lognormal

# Poisson
sample.pois <- rtrunc(n = sample_size, lambda = 10, a = 4, family = "Poisson")
mlEstimationTruncDist(
  sample.pois,
  y.min = 4, max.it = 500, delta = 0.33,
  print.iter = 5
)

# Gamma
sample.gamma <- rtrunc(n = sample_size, shape = 6, rate = 2, a = 2, family = "Gamma")
mlEstimationTruncDist(
  sample.gamma,
  y.min = 2, max.it = 1500, delta = 0.3,
  print.iter = 10
)

# Negative binomial
sample.nbinom <- rtruncnbinom(sample_size, size = 50, prob = .3, a = 100, b = 120)
mlEstimationTruncDist(sample.nbinom, r=10)
```

natural2parameters *Convert natural parameters to distribution parameters*

Description

Convert natural parameters to distribution parameters

Usage

```
natural2parameters(eta)
```

Arguments

eta vector of natural parameters

Value

A vector of the original distribution parameters

Examples

```
samp <- rtrunc(n = 100, lambda = 2, family = "Poisson")
natural2parameters(init.parms(samp))
```

parameters2natural *Convert distribution parameters to natural parameters*

Description

Convert distribution parameters to natural parameters

Usage

```
parameters2natural(parms)
```

Arguments

parms A vector of parameters in a distribution distribution

Value

A vector containing the natural parameters

Examples

```
# Poisson distribution
samp <- rtrunc(n = 100, lambda = 2, family = "Poisson")
parameters2natural(init.parms(samp))
```

print.trunc	<i>Print sample from truncated distribution</i>
-------------	---

Description

Special printing methods for trunc_* classes.

Usage

```
## S3 method for class 'trunc'
print(x, details = FALSE, ...)
```

Arguments

x	object to print
details	if FALSE (default), hides the attributes of x
...	other arguments passed to print.default

Value

x with or without its attributes

Author(s)

Waldir Leoncio

rtrunc	<i>The Truncated Exponential Family</i>
--------	---

Description

Random generation for the truncated exponential family distributions. Please refer to the "Details" and "Examples" section for more information on how to use this function.

Usage

```
rtrunc(n, family = "gaussian", ...)

rtruncbeta(n, shape1, shape2, a = 0, b = 1)

rtruncbinom(n, size, prob, a = 0, b = size)

rtruncchisq(n, df, a = 0, b = Inf)

rtrunccontbern(n, lambda, a = 0, b = 1)
```



```

rtruncexp(n, rate = 1, a = 0, b = Inf)
rtruncgamma(n, shape, rate = 1, scale = 1/rate, a = 0, b = Inf)
rtruncinvgamma(n, shape, rate = 1, scale = 1/rate, a = 0, b = Inf)
rtruncinvgauss(n, m, s, a = 0, b = Inf)
rtrunclnorm(n, meanlog, sdlog, a = 0, b = Inf)
rtruncbinom(n, size, prob, mu, a = 0, b = Inf)
rtruncnorm(n, mean, sd, a = -Inf, b = Inf)
rtruncpois(n, lambda, a = 0, b = Inf)

```

Arguments

n	sample size
family	distribution family to use
...	individual arguments to each distribution
shape1	positive shape parameter alpha
shape2	positive shape parameter beta
a	point of left truncation
b	point of right truncation
size	target for number of successful trials, or dispersion parameter (the shape parameter of the gamma mixing distribution). Must be strictly positive, need not be integer.
prob	probability of success on each trial
df	degrees of freedom for "parent" distribution
lambda	mean and var of "parent" distribution
rate	inverse gamma rate parameter
shape	inverse gamma shape parameter
scale	inverse gamma scale parameter
m	vector of means
s	vector of dispersion parameters
meanlog	mean of un-truncated distribution
sdlog	standard deviation of un-truncated distribution
mu	alternative parametrization via mean
mean	mean of parent distribution
sd	standard deviation is parent distribution

Details

One way to use this function is by calling the `rtrunc` generic with the `family` parameter of your choice. You can also specifically call one of the methods (e.g. `rtrunc.poisson(10, lambda=3)` instead of `rtrunc(10, family="poisson", lambda=3)`). The latter is more flexible (i.e., easily programmable) and more robust (i.e., it contains better error handling and validation procedures), while the former better conforms with the nomenclature from other distribution-related functions in the `stats` package.

Value

A sample of size `n` drawn from a truncated distribution

vector of one of the `rtrunc_*` classes containing the sample elements, as well as some attributes related to the chosen distribution.

Note

The current sample-generating algorithm may be slow if the distribution is largely represented by low-probability values. This will be fixed soon. Please follow <https://github.com/ocbe-uio/TruncExpFam/issues/72> for details.

Author(s)

René Holst, Waldir Leôncio

Examples

```
# Truncated binomial distribution
sample.binom <- rtrunc(100, family = "binomial", prob = 0.6, size = 20, a = 4, b = 10)
sample.binom
plot(table(sample.binom), ylab = "Frequency", main = "Freq. of sampled values")

# Truncated Log-Normal distribution
sample.lognorm <- rtrunc(
  n = 100, family = "lognormal", meanlog = 2.5, sdlog = 0.5, a = 7
)
summary(sample.lognorm)

hist(
  sample.lognorm,
  nclass = 35, xlim = c(0, 60), freq = FALSE,
  ylim = c(0, 0.15)
)

# Normal distribution
sample.norm <- rtrunc(n = 100, mean = 2, sd = 1.5, a = -1)
head(sample.norm)
hist(sample.norm, nclass = 25)

# Gamma distribution
sample.gamma <- rtrunc(n = 100, family = "gamma", shape = 6, rate = 2, a = 2)
hist(sample.gamma, nclass = 15)
```

```
# Poisson distribution
sample.pois <- rtrunc(n = 10, family = "poisson", lambda = 10, a = 4)
sample.pois
plot(table(sample.pois))
```

TruncExpFam

Truncated Exponential Family

Description

TruncExpFam is an R package to handle truncated members from the exponential family.

Details

This package offers truncated versions of popular distribution functions from the stats package, as well as functions to retrieve the original distribution parameters from a truncated sample. For more info, please check [rtrunc\(\)](#), [dtrunc\(\)](#) and [print.trunc\(\)](#).

Supported distributions

- Beta
- Binomial
- Chi-Square
- Continuous Bernoulli
- Exponential
- Gamma
- Inverse Gamma
- Inverse Gaussian
- Log-normal
- Negative Binomial
- Normal
- Poisson

Note

Found a bug? Want to suggest a feature? Contribute to the scientific and open source communities by opening an issue on our home page. Check the "BugReports" field on `packageDescription("TruncExpFam")` for the URL.

validateFamilyParms *Validate family parameters*

Description

Checks if a combination of distribution family and parameters is valid.

Usage

```
validateFamilyParms(family, parms, verbose = FALSE)
```

Arguments

family	character with family distribution name
parms	character vector with distribution parameter names
verbose	print intermediate messages?

Value

list telling if family-parm combo is valid + the family name

Author(s)

Waldir Leoncio

Index

`.onAttach`, 2

`dtrunc`, 2
`dtrunc()`, 11
`dtruncbeta (dtrunc)`, 2
`dtruncbinom (dtrunc)`, 2
`dtruncchisq (dtrunc)`, 2
`dtrunccontbern (dtrunc)`, 2
`dtruncexp (dtrunc)`, 2
`dtruncgamma (dtrunc)`, 2
`dtruncinvgamma (dtrunc)`, 2
`dtruncinvgauss (dtrunc)`, 2
`dtrunclnorm (dtrunc)`, 2
`dtruncbinom (dtrunc)`, 2
`dtruncnorm (dtrunc)`, 2
`dtruncpois (dtrunc)`, 2

`genrtruncClass`, 3

`init.parms`, 4

`mlEstimationTruncDist`, 5

`natural2parameters`, 7

`parameters2natural`, 7
`print.trunc`, 8
`print.trunc()`, 11
`print.trunc_beta (print.trunc)`, 8
`print.trunc_binomial (print.trunc)`, 8
`print.trunc_chisq (print.trunc)`, 8
`print.trunc_contbern (print.trunc)`, 8
`print.trunc_exp (print.trunc)`, 8
`print.trunc_gamma (print.trunc)`, 8
`print.trunc_invgamma (print.trunc)`, 8
`print.trunc_invgauss (print.trunc)`, 8
`print.trunc_lognormal (print.trunc)`, 8
`print.trunc_nbinom (print.trunc)`, 8
`print.trunc_normal (print.trunc)`, 8
`print.trunc_poisson (print.trunc)`, 8

`rtrunc`, 8
`rtrunc()`, 11
`rtruncbeta (rtrunc)`, 8
`rtruncbinom (rtrunc)`, 8
`rtruncchisq (rtrunc)`, 8
`rtrunccontbern (rtrunc)`, 8
`rtruncexp (rtrunc)`, 8
`rtruncgamma (rtrunc)`, 8
`rtruncinvgamma (rtrunc)`, 8
`rtruncinvgauss (rtrunc)`, 8
`rtrunclnorm (rtrunc)`, 8
`rtruncbinom (rtrunc)`, 8
`rtruncnorm (rtrunc)`, 8
`rtruncpois (rtrunc)`, 8

`TruncExpFam`, 11

`validateFamilyParms`, 12