Package ‘TruncExpFam’

April 10, 2025

Title Truncated Exponential Family

Version 1.2.1

Date 2025-04-10

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 7.3.2

Imports methods, invgamma, rmutil

Suggests knitr, rmarkdown, testthat

URL `https://ocbe-uio.github.io/TruncExpFam/`

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

VignetteBuilder knitr

Language en-US

NeedsCompilation no

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

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

Repository CRAN

Date/Publication 2025-04-10 15:00:02 UTC

Contents

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

Description

Prints package version number and welcome message on package load

Usage

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

Arguments

`libname`	library location. See `?base::.onAttach` for details
`pkgname`	package name. See `?base::.onAttach` for details

`averageT` *Averages out the sufficient statistics T(y)*

Description

Takes a vector of values and returns the column average of their sufficient statistic (determined by their class)

Usage

```
averageT(y)
```

Arguments

`y`	vector of values

Value

A vector with the average of the sufficient statistics

`dtruncbeta` *Probability Density Function*

Description

Calculates the PDF for a given truncated distribution

Usage

```
dtruncbeta(y, shape1, shape2, eta, a = 0, b = 1, ...)

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

dtruncchisq(y, df, eta, a = 0, b = Inf, ...)

dtrunccontbern(y, lambda, eta, a = 0, b = 1, ...)

dtrunccontbern(y, lambda, eta, a = 0, b = 1, ...)

dtrunc(y, ...)

dtruncexp(y, rate = 1, eta, a = 0, b = Inf, ...)

dtruncgamma(y, shape, rate = 1, scale = 1/rate, eta, a = 0, b = Inf, ...)

dtruncinvgamma(y, shape, rate = 1, scale = 1/rate, eta, a = 0, b = Inf, ...)
```

```
dtruncinvgauss(y, m, s, eta, a = 0, b = Inf, ...)

dtrunclnorm(y, meanlog = 0, sdlog = 1, eta, a = 0, b = Inf, ...)

## S3 method for class 'trunc_nbinom'
dtrunc(y, size, prob, eta, a = 0, b = Inf, ...)

dtruncnbinom(y, size, prob, eta, a = 0, b = Inf, ...)

dtruncnbinom(y, size, prob, eta, a = 0, b = Inf, ...)

dtruncnorm(y, mean = 0, sd = 1, eta, a = -Inf, b = Inf, ...)

dtruncpois(y, lambda, eta, a = 0, b = Inf, ...)
```

Arguments

`y`	output from rtrunc or any valid numeric value(s).
`shape1`	positive shape parameter alpha
`shape2`	positive shape parameter beta
`eta`	vector of natural parameters
`a`	point of left truncation. For discrete distributions, `a` will be included in the support of the truncated distribution.
`b`	point of right truncation
`...`	size
`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 untruncated distribution
`sdlog`	standard deviation of untruncated distribution
`mean`	mean of parent distribution
`sd`	standard deviation is parent distribution

Value

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

Note

Either the common or the natural parameters must be provided.

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))
```

`empiricalParameters` *Calculate empirical parameters*

Description

Returns the empirical parameter estimate for a distribution

Usage

```
empiricalParameters(y, ...)
```

Arguments

`y`	output of rtrunc
`...`	other arguments passed to methods

Value

A vector of parameter estimates for the input sample

Examples

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

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

empiricalParameters.numeric
Extract parameters

Description

Extract parameters

Usage

```
## S3 method for class 'numeric'
empiricalParameters(y, family = "gaussian", natural = FALSE, ...)
```

Arguments

`y`	Numeric vector containing observations from a random variable
`family`	Distribution family to assume for `y`
`natural`	Should output be in terms of the natural parameter eta?
`...`	arguments passed to `empiricalParameters()`

Examples

```
# Some random data
x <- c(
  4, 3, 6, 3, 3, 3, 3, 4, 3, 2, 3, 0, 4, 2, 0, 1, 4, 3, 0, 0, 2, 3, 0, 3, 7,
  2, 1, 1, 2, 3, 2, 3, 3, 3, 2, 2, 2, 0, 2, 0, 2, 1, 0, 2, 3, 1, 0, 4, 2, 2,
  0, 1, 1, 1, 2, 2, 3, 1, 3, 1, 1, 0, 3, 3, 2, 0, 2, 2, 3, 0, 2, 1, 0, 0, 1,
  0, 2, 4, 2, 3, 3, 0, 1, 0, 5, 2, 4, 2, 7, 4, 4, 1, 2, 4, 3, 2, 4, 3, 1, 3
)

# Extracting parameters under different distribution assumptions
empiricalParameters(x, family = "normal")
empiricalParameters(x, family = "normal", natural = TRUE)
empiricalParameters(x, family = "binomial", nsize = max(x))
empiricalParameters(x, family = "poisson", natural = FALSE)
empiricalParameters(x, family = "poisson", natural = TRUE)
```

genrtruncClass
Generates an rtrunc-dispatchable class

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

`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 = 100,
  delta = 0.33,
  print.iter = 0,
  ny = 100,
  family = NULL,
  ...
)
```

Arguments

`y`	Sequence spanning the domain of the truncated distribution
`y.min`	Lower bound for `y`
`y.max`	Upper bound for `y`
`tol`	Error tolerance for parameter estimation
`max.it`	Maximum number of iterations
`delta`	Indirectly, the difference between consecutive iterations to compare with the error tolerance

`print.iter`	Determines the frequency of printing (i.e., prints every `print.iter` iterations)
`ny`	size of intermediate `y` range sequence. Higher values yield better estimations but slower iterations
`family`	distribution family to use
`...`	other parameters passed to subfunctions

Details

If `print.iter` = `TRUE`, the function prints the iteration, the sum of squares of `delta.eta.j` (`delta.L2`), and the current parameter estimates. The `delta` argument of this function is a factor in the calculation of `delta.eta.j`, which in turn is a factor in the calculation of `delta.L2`.

Value

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

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
...	other arguments passed to methods

Value

A vector of the original distribution parameters

See Also

parameters2natural()

Examples

```
samp <- rtrunc(n = 100, lambda = 2, family = "Poisson")
lambda_hat <- empiricalParameters(samp)
eta_hat <- parameters2natural(lambda_hat)
natural2parameters(eta_hat)  # yields back lambda
```

---

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
`...`	other arguments passed to methods

Value

A vector containing the natural parameters

See Also

`natural2parameters()`

Examples

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

`print.trunc` *Print sample from truncated distribution*

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 `base::print.default()`

Value

`x` with or without its attributes

Author(s)

Waldir Leoncio

`probdist-class` *Probability distribution class*

Description

An R object describing the properties of a probability distribution.

Value

An RC class containing statistical properties of that distribution, namely its name, parameter names and values and natural parameter names and values.

Author(s)

Waldir Leoncio

Examples

```
probdist(shape = 2, scale = .25, family = "gamma")
probdist(mean = 2, sd = 10, family = "normal")
probdist(eta1 = 2, eta2 = -1, family = "normal")
```

ptrunc *Cumulative Distribution Function*

Description

Calculates the cumulative probability for a given truncated distribution

Usage

```
ptrunc(q, family, ..., lower.tail = TRUE, log.p = FALSE)

ptruncnorm(
  q,
  mean = 0,
  sd = 1,
  a = -Inf,
  b = Inf,
  ...,
  lower.tail = TRUE,
  log.p = FALSE
)

ptruncbeta(
  q,
  shape1,
  shape2,
  a = 0,
  b = 1,
  ...,
  lower.tail = TRUE,
  log.p = FALSE
)

ptruncbinom(
  q,
  size,
  prob,
  a = 0,
  b = size,
  ...,
  lower.tail = TRUE,
  log.p = FALSE
)

ptruncpois(q, lambda, a = 0, b = Inf, ..., lower.tail = TRUE, log.p = FALSE)

ptruncchisq(q, df, a = 0, b = Inf, ..., lower.tail = TRUE, log.p = FALSE)
```

```
ptrunccontbern(q, lambda, a = 0, b = 1, ...)

ptruncexp(q, rate = 1, a = 0, b = Inf, ..., lower.tail = TRUE, log.p = FALSE)

ptruncgamma(
  q,
  shape,
  rate = 1,
  scale = 1/rate,
  a = 0,
  b = Inf,
  ...,
  lower.tail = TRUE,
  log.p = FALSE
)

ptruncinvgamma(
  q,
  shape,
  rate = 1,
  scale = 1/rate,
  a = 0,
  b = Inf,
  ...,
  lower.tail = TRUE,
  log.p = FALSE
)

ptruncinvgauss(q, m, s, a = 0, b = Inf, ...)

ptrunclnorm(
  q,
  meanlog = 0,
  sdlog = 1,
  a = 0,
  b = Inf,
  ...,
  lower.tail = TRUE,
  log.p = FALSE
)

ptruncnbinom(
  q,
  size,
  prob,
  mu,
  a = 0,
  b = Inf,
  ...,
  lower.tail = TRUE,
  log.p = FALSE
)
```

Arguments

`q`	vector of quantiles
`family`	distribution family to use
`...`	*named* distribution parameters and/or truncation limits (`a`, `b`)
`lower.tail`	logical; if `TRUE`, probabilities are $P(X <= x)$ otherwise, $P(X > x)$
`log.p`	logical; if `TRUE`, probabilities p are given as `log(p)`
`mean`	mean of parent distribution
`sd`	standard deviation is parent distribution
`a`	point of left truncation. For discrete distributions, `a` will be included in the support of the truncated distribution.
`b`	point of right truncation
`shape1`	positive shape parameter alpha
`shape2`	positive shape parameter beta
`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
`lambda`	mean and var of "parent" distribution
`df`	degrees of freedom for "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 untruncated distribution
`sdlog`	standard deviation of untruncated distribution
`mu`	alternative parametrization via mean

Value

The cumulative probability of y.

Examples

```
ptrunc(0)
ptrunc(6, family = "gaussian", mean = 5, sd = 10, b = 7)
pnorm(6, mean = 5, sd = 10) # for comparison
```

qtrunc — *Quantile Function*

Description

Calculates quantile for a given truncated distribution and probability.

Usage

```
qtrunc(p, family, ..., lower.tail = TRUE, log.p = FALSE)

qtruncbeta(
  p,
  shape1,
  shape2,
  a = 0,
  b = 1,
  ...,
  lower.tail = TRUE,
  log.p = FALSE
)

qtruncbinom(
  p,
  size,
  prob,
  a = 0,
  b = size,
  ...,
  lower.tail = TRUE,
  log.p = FALSE
)

qtruncchisq(p, df, a = 0, b = Inf, ..., lower.tail = TRUE, log.p = FALSE)

qtrunccontbern(p, lambda, a = 0, b = 1, ..., lower.tail = TRUE, log.p = FALSE)

qtruncexp(p, rate = 1, a = 0, b = Inf, ..., lower.tail = TRUE, log.p = FALSE)

qtruncgamma(
  p,
  shape,
  rate = 1,
  scale = 1/rate,
  a = 0,
  b = Inf,
  ...,
  lower.tail = TRUE,
  log.p = FALSE
)

qtruncinvgamma(
  p,
  shape,
  rate = 1,
  scale = 1/rate,
  a = 0,
  b = Inf,
  ...,
  lower.tail = TRUE,
  log.p = FALSE
)

qtruncinvgauss(p, m, s, a = 0, b = Inf, ..., lower.tail = TRUE, log.p = FALSE)

qtrunclnorm(
  p,
  meanlog = 0,
  sdlog = 1,
  a = 0,
  b = Inf,
  ...,
  lower.tail = TRUE,
  log.p = FALSE
)

qtruncnbinom(
  p,
  size,
  prob,
  mu,
  a = 0,
  b = Inf,
  ...,
  lower.tail = TRUE,
  log.p = FALSE
)

qtruncnorm(
  p,
  mean = 0,
  sd = 1,
  a = -Inf,
  b = Inf,
  ...,
  lower.tail = TRUE,
  log.p = FALSE
)

qtruncpois(p, lambda, a = 0, b = Inf, ..., lower.tail = TRUE, log.p = FALSE)
```

Arguments

`p`	vector of quantiles
`family`	distribution family to use
`...`	*named* distribution parameters and/or truncation limits (a, b)
`lower.tail`	logical; if TRUE, probabilities are $P(X <= x)$ otherwise, $P(X > x)$
`log.p`	logical; if TRUE, probabilities p are given as log(p)
`shape1`	positive shape parameter alpha
`shape2`	positive shape parameter beta
`a`	point of left truncation. For discrete distributions, a will be included in the support of the truncated distribution.
`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 untruncated distribution
`sdlog`	standard deviation of untruncated distribution
`mu`	alternative parametrization via mean
`mean`	mean of parent distribution
`sd`	standard deviation is parent distribution

Value

The quantile of p.

Examples

```
qtrunc(0.75)
qtrunc(.2, family = "gaussian", mean = 5, sd = 10, b = 7)
qnorm(.2, mean = 5, sd = 10) # for comparison
```

`rtruncbeta` *The Truncated Exponential Family*

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

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

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

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

rtrunccontbern(n, lambda, a = 0, b = 1, faster = FALSE)

rtruncexp(n, rate = 1, a = 0, b = Inf, faster = FALSE)

rtruncgamma(n, shape, rate = 1, scale = 1/rate, a = 0, b = Inf, faster = FALSE)

rtruncinvgamma(
  n,
  shape,
  rate = 1,
  scale = 1/rate,
  a = 0,
  b = Inf,
  faster = FALSE
)

rtruncinvgauss(n, m, s, a = 0, b = Inf, faster = FALSE)

rtrunclnorm(n, meanlog, sdlog, a = 0, b = Inf, faster = FALSE)

rtruncnbinom(n, size, prob, mu, a = 0, b = Inf, faster = FALSE)

rtruncnorm(n, mean, sd, a = -Inf, b = Inf, faster = FALSE)

rtruncpois(n, lambda, a = 0, b = Inf, faster = FALSE)

rtrunc(n, family = "gaussian", faster = FALSE, ...)

rtrunc_direct(n, family = "gaussian", parms, a, b, ...)
```

Arguments

`n`	sample size
`shape1`	positive shape parameter alpha
`shape2`	positive shape parameter beta
`a`	point of left truncation. For discrete distributions, `a` will be included in the support of the truncated distribution.
`b`	point of right truncation
`faster`	if `TRUE`, samples directly from the truncated distribution (more info in details)
`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 untruncated distribution
`sdlog`	standard deviation of untruncated distribution
`mu`	alternative parametrization via mean
`mean`	mean of parent distribution
`sd`	standard deviation is parent distribution
`family`	distribution family to use
`...`	individual arguments to each distribution
`parms`	list of parameters passed to rtrunc (through the `...` element)

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. `rtruncpois(10, lambda=3)` instead of `rtrunc(10, family="poisson", lambda=3)). The latter is more flexible (i.e., easily programmable) and` package.

Setting `faster=TRUE` uses a new algorithm that samples directly from the truncated distribution, as opposed to the old algorithm that samples from the untruncated distribution and then truncates the result. The advantage of the new algorithm is that it is way faster than the old one, particularly for highly-truncated distributions. On the other hand, the sample for untruncated distributions called through `rtrunc()` will no longer match their stats-package counterparts for the same seed.

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 counterparts of the density-, distribution-, quantile- and sampling-functions for a broad range of distributions from the exponential family, as implemented in the stats package.

The package also provides functions for estimating the parameters of the distributions from data, given the truncation limits.

For more info, please check `rtrunc()`, `dtrunc()` and `print.trunc()`. Counterparts for density and probability functions are on the roadmap for a future release.

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.

Author(s)

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

Authors:

- René Holst <`rene.holst@medisin.uio.no`>

See Also

Useful links:

- https://ocbe-uio.github.io/TruncExpFam/
- Report bugs at https://github.com/ocbe-uio/TruncExpFam/issues

validateFamilyParms *Validate family parameters*

Description

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

Usage

```
validateFamilyParms(family, parms)
```

Arguments

family	character with family distribution name
parms	character vector with distribution parameter names

Value

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

Author(s)

Waldir Leoncio

Index