Package 'univariateML'

July 22, 2025

Type Package

Title Maximum Likelihood Estimation for Univariate Densities

Version 1.5.0

Description User-friendly maximum likelihood estimation (Fisher (1921) <doi:10.1098/rsta.1922.0009>) of univariate densities.

License MIT + file LICENSE

Encoding UTF-8

LazyData true

Depends R (>= 2.10)

Imports assertthat, extraDistr, tibble, logitnorm, actuar, nakagami, fGarch, rlang, intervals, Rfast, pracma, sads

Suggests testthat, knitr, rmarkdown, markdown, copula, dplyr, covr

RoxygenNote 7.3.2

VignetteBuilder knitr

URL https://github.com/JonasMoss/univariateML,

https://jonasmoss.github.io/univariateML/

BugReports https://github.com/JonasMoss/univariateML/issues

NeedsCompilation no

Author Jonas Moss [aut, cre] (ORCID: <https://orcid.org/0000-0002-6876-6964>),
Thomas Nagler [ctb],
Chitu Okoli [ctb]

Maintainer Jonas Moss <jonas.gjertsen@gmail.com>

Repository CRAN

Date/Publication 2025-03-04 13:30:02 UTC

Contents

---

univariateML-package   *univariateML*

Description

An R-package for fast, easy, and reliable maximum likelihood estimation for a selection of parametric univariate densities.

Details

Data analysis often requires the estimation of univariate densities. Maximum likelihood estimation is sensible for almost every univariate density used in practice. Moreover, the maximum likelihood estimator is typically consistent and efficient.

The purpose of this package is to

- Support maximum likelihood estimation of a large selection of densities.
- Supports plenty of generics such as plot and AIC to aid your data analysis.

Read the vignettes to learn more about univariateML: browseVignettes(package = "univariateML")

Author(s)

Maintainer: Jonas Moss <jonas.gjertsen@gmail.com> (ORCID)

Other contributors:

- Thomas Nagler <mail@tnagler.com> [contributor]
- Chitu Okoli <chitu.okoli@skema.edu> [contributor]

See Also

Useful links:

- https://github.com/JonasMoss/univariateML
- https://jonasmoss.github.io/univariateML/
- Report bugs at https://github.com/JonasMoss/univariateML/issues

---

abalone   *Abalone data*

Description

Physical measurements of 4177 abalones, a species of sea snail.

Usage

    abalone

Format

A [tibble](#) with 4,177 observations and 9 variables:

sex Sex of the abalone, `F` is female, `M` male, and `I` infant.

length Longest shell measurement.

diameter Diameter perpendicular to length.

height Height with with meat in shell.

whole_weight Grams whole abalone.

shucked_weight Grams weight of meat.

viscera_weight Grams gut weight (after bleeding).

shell_weight Grams after being dried.

rings +1.5 gives the age in years.

Details

See the web page <https://archive.ics.uci.edu/ml/datasets/Abalone> for more information about the data set.

Source

Dua, D. and Graff, C. (2019). UCI Machine Learning Repository <https://archive.ics.uci.edu/ml/>. Irvine, CA: University of California, School of Information and Computer Science.

References

Ko, V., Hjort, N. L., & Hobaek Haff, I. (2019). Focused information criteria for copulas. Scandinavian Journal of Statistics.

Examples

    abalone

| bootstrapml | *Parametric Bootstrap on Distributions Fitted with Maximum Likelihood* |

Description

The parametric bootstrap is a resampling technique using random variates from a known parametric distribution. In this function the distribution of the random variates is completely determined by the unvariateML object object.

Usage

```
bootstrapml(
  object,
  reps = 1000,
  map = identity,
  reducer = stats::quantile,
  ...
)
```

Arguments

object	A univariateML object.
reps	Positive integer. The number of bootstrap samples.
map	A function of the parameters of the univariateML object. Defaults to the identity.
reducer	A reducer function. Defaults to stats::quantile with default argument probs = c(0.025, 0.975).
...	Passed to reducer.

Details

For each bootstrap iteration a maximum likelihood estimate is calculated using the ml*** function specified by object. The resulting numeric vector is then passed to map. The values returned by map is collected in an array and the reducer is called on each row of the array.

By default the map function is the identity and the default reducer is the quantile function taking the argument probs, which defaults to c(0.025, 0.975). This corresponds to a 95\ basic percentile confidence interval and is also reported by confint()

Note: The default confidence intervals are percentile intervals, not empirical intervals. These confidence intervals will in some cases have poor coverage as they are not studentized, see e.g. Carpenter, J., & Bithell, J. (2000).

Value

The transposed map-reduced bootstrap samples.

References

Efron, B., & Tibshirani, R. J. (1994). An introduction to the bootstrap. CRC press.

Carpenter, J., & Bithell, J. (2000). Bootstrap confidence intervals: when, which, what? A practical guide for medical statisticians. Statistics in medicine, 19(9), 1141-1164.

See Also

[confint()](#) for an application of `bootstrapml`.

Examples

```
set.seed(1)
object <- mlgamma(mtcars$qsec)

## Calculate c(0.025, 0.975) confidence interval for the gamma parameters.
bootstrapml(object)

#              2.5%      97.5%
# shape 68.624945 160.841557
# rate   3.896915   9.089194

## The mean of a gamma distribution is shape/rate. Now we calculate a
## parametric bootstrap confidence interval for the mean with confidence
## limits c(0.05, 0.95)

bootstrapml(object, map = function(x) x[1] / x[2], probs = c(0.05, 0.95))

#        5%       95%
# 17.33962 18.31253

## Print a histogram of the bootstrapped estimates from an exponential.
object <- mlexp(mtcars$qsec)
hist(bootstrapml(object, reducer = identity))
```

---

confint.univariateML   *Confidence Intervals for Maximum Likelihood Estimates*

Description

Computes a confidence interval for one or more parameters in a `unvariateML` object.

Usage

```
## S3 method for class 'univariateML'
confint(object, parm = NULL, level = 0.95, Nreps = 1000, ...)
```

Arguments

`object`	An object of class `univariateML`.
`parm`	Vector of strings; the parameters to calculate a confidence interval for. Each parameter must be a member of `names(object)`.
`level`	The confidence level.
`Nreps`	Number of bootstrap iterations. Passed to `bootstrapml()`.
`...`	Additional arguments passed to `bootstrapml()`.

Details

`confint.univariateML` is a wrapper for `bootstrapml()` that computes confidence intervals for the main parameters of `object`. The main parameters of `object` are the members of `names(object)`. For instance, the main parameters of an object obtained from `mlnorm` are mean and sd. The confidence intervals are parametric bootstrap percentile intervals with limits `(1-level)/2` and `1 - (1-level)`.

Value

A matrix or vector with columns giving lower and upper confidence limits for each parameter in `parm`.

See Also

`stats::confint()` for the generic function and `bootstrapml()` for the function used to calculate the confidence intervals.

Examples

```
object <- mlinvgauss(airquality$Wind)
confint(object) # 95% confidence interval for mean and shape
confint(object, "mean") # 95% confidence interval for the mean parameter
# confint(object, "variance") # Fails since 'variance isn't a main parameter.
```

---

corbet	*Frequencies of butterflies collected in Malaya*

Description

Species abundance data from 1943; a classical application of the logarithmic series distribution.

Usage

```
corbet
```

Format

A vector of size 501 containing integer observations between 1 and 24.

References

Fisher, R. A., Corbet, A. S., & Williams, C. B. (1943). The relation between the number of species and the number of individuals in a random sample of an animal population. The Journal of Animal Ecology, 12(1), 42. https://doi.org/10.2307/1411

Examples

```
corbet
```

---

egypt　　　　　　　　　　　　　　*Mortality data from ancient Egypt*

Description

Age at death of 141 Roman era Egyptian mummies.

Usage

```
egypt
```

Format

A [tibble](#) with 141 observations and 2 variables:

age Age at death.

sex Sex of deceased; 82 males and 49 females.

Details

This data was collected by Spiegelberg (1901) and analyzed by Karl Pearson (1902) in the first volume of Biometrika. It was analyzed by Claeskens & Hjort (2008) and the data is based on their transcription.

References

Spiegelberg, W. (1901). Aegyptische und Griechische Eigennamen aus Mumientiketten der R<U+00F6>mischen Kaiserzeit.

Pearson, K. (1902). On the change in expectation of life in man during a period of circa 2000 years. Biometrika, 1(2), 261-264.

Claeskens, G., & Hjort, N. L. (2008). Model selection and model averaging. Cambridge University Press.

See Also

The source of the data is https://feb.kuleuven.be/public/u0043181/modelselection/datasets/egyptlives_data.txt

Examples

```
egypt
```

---

get_start	*Returns appropriate starting value*

Description

Returns appropriate starting value

Usage

```
get_start(default, name, ...)
```

Arguments

default	Function to calculate default parameter value.
name	Name of default starting value.
...	Parameter list containing an element name.

Value

Default value if name is not present

---

inverse_digamma	*Inverse digamma function*

Description

Calculates the inverse digamma function using Newton–Raphson. Works for y > -500. Uses Newton–Raphson with relative tolerance of eps^0.25.

Usage

```
inverse_digamma(y)
```

Arguments

y	Values to invert.

Details

The number of iterations are few, 1 for most input values, especially those that are large. The starting value is the lower bound found by Batir (2017).

References

Batir, N. (2017). Inequalities for the inverses of the polygamma functions. arXiv. http://arxiv.org/abs/1705.06547

```
MaximumLikelihoodDistribution
```
Maximum likelihood estimated distribution

Description

Density, distribution function, quantile function and random generation for a univariate distribution estimated by maximum likelihood.

Usage

```
dml(x, obj, log = FALSE)

pml(q = q, obj, lower.tail = TRUE, log.p = FALSE)

qml(p = p, obj, lower.tail = TRUE, log.p = FALSE)

rml(n = n, obj)
```

Arguments

x, q	vector of quantiles.
obj	an `univariateML` object.
log, log.p	logical; if `TRUE`, the probabilities p are gives as `log(p)`.
lower.tail	logical; if `TRUE` (default), the probabilities are $P[X \leq x]$ otherwise, $P[X > x]$
p	vector of probabilities.
n	number of observations. If `length(n) > 1`, the length is taken to be the number required.

Details

`dml` is the density, `pml` is the distribution function, `qml` is the quantile function, and `rml` is the random variable generator.

These functions work like their counterparts in `stats`, e.g. [Normal](). The `univariateML` object contains both maximum likelihood estimates and the identity of the model these estimates were calculated under. These functions are wrappers around underlying density, distribution, quantile and random generation functions where unknown parameters are filled with the maximum likelihood estimates. See the example.

Value

`dml` gives the density, `pml` gives the distribution function, `qml` gives the quantile function, and `rml` generates random deviates.

Examples

```
## Simple example
obj <- mlnorm(airquality$Wind)
dml(0.5, obj) == dnorm(0.5, mean = obj[1], sd = obj[2])

obj <- mlbetapr(airquality$Wind)

# Plot the logarithm of the beta prime distribution.
plot(function(x) dml(x, obj, log = TRUE),
  from = 0, to = 20,
  main = "Logarithm of Density", ylab = NA, lwd = 2
)
```

---

mlbeta   *Beta distribution maximum likelihood estimation*

Description

Uses stat::nlm to estimate the parameters of the Beta distribution.

Usage

```
mlbeta(x, na.rm = FALSE, ...)
```

Arguments

x	a (non-empty) numeric vector of data values.
na.rm	logical. Should missing values be removed?
...	Ignored.

Details

For the density function of the Beta distribution see Beta.

For type, the option none is fastest.

Value

mlbeta returns an object of class univariateML. This is a named numeric vector with maximum likelihood estimates for shape1 and shape2 and the following attributes:

model	The name of the model.
density	The density associated with the estimates.
logLik	The loglikelihood at the maximum.
support	The support of the density.
n	The number of observations.
call	The call as captured my match.call

References

Johnson, N. L., Kotz, S. and Balakrishnan, N. (1995) Continuous Univariate Distributions, Volume 2, Chapter 25. Wiley, New York.

See Also

[Beta](#) for the Beta density, [nlm](#) for the optimizer this function uses.

Examples

```
AIC(mlbeta(USArrests$Rape / 100))
```

---

mlbetapr   *Beta prime distribution maximum likelihood estimation*

Description

This function does not estimate the scale parameter for the BetaPrime distribution. Transforms the data and uses stat::nlm to estimate the parameters of the Beta distribution.

Usage

```
mlbetapr(x, na.rm = FALSE, ...)
```

Arguments

x	a (non-empty) numeric vector of data values.
na.rm	logical. Should missing values be removed?
...	passed to [mlbeta](#).

Details

For the density function of the Beta prime distribution see [BetaPrime](#).

For type, the option none is fastest.

Value

mlbetapr returns an object of [class](#) univariateML. This is a named numeric vector with maximum likelihood estimates for shape1 and shape2 and the following attributes:

model	The name of the model.
density	The density associated with the estimates.
logLik	The loglikelihood at the maximum.
support	The support of the density.
n	The number of observations.
call	The call as captured my match.call

References

Johnson, N. L., Kotz, S. and Balakrishnan, N. (1995) Continuous Univariate Distributions, Volume 2, Chapter 25. Wiley, New York.

See Also

BetaPrime for the Beta prime density, nlm for the optimizer this function uses, mlbeta for the Beta distribution maximum likelihood estimator.

Examples

```
AIC(mlbetapr(USArrests$Rape))
```

---

mlbinom   *Binomial distribution maximum likelihood estimation*

Description

For the density function of the Binomial distribution see Binomial.

Usage

```
mlbinom(x, na.rm = FALSE, ...)
```

Arguments

x	a (non-empty) numeric vector of data values.
na.rm	logical. Should missing values be removed?
...	The arguments size can be specified to only return the ml of prob. reltol is the relative accuracy requested, defaults to .Machine$double.eps^0.25. iterlim is a positive integer specifying the maximum number of iterations to be performed before the program is terminated (defaults to 100).

Details

The estimator computes both the size and prob parameter by default. Be aware that the likelihood will often be unbounded. According to Olkin et al. (1981), the likelihood is unbounded when $\hat{\mu}/\hat{\sigma}^2 \leq 1$, where $\hat{\sigma}^2$ is the biased sample variance. When the likelihood is unbounded, the maximum likelihood estimator can be regarded as a Poisson with lambda parameter equal to the mean of the observation.

When $\hat{\mu}/\hat{\sigma}^2 \leq 1$ and size is not supplied by the user, an error is cast. If size is provided and size < max(x), an error is cast.

The maximum likelihood estimator of size is unstable, and improvements exist. See, e.g., Carroll and Lomard (1985) and DasGupta and Rubin (2005).

Value

mlbinom returns an object of [class](#) univariateML. This is a named numeric vector with maximum likelihood estimates for size and prob and the following attributes:

model	The name of the model.
density	The density associated with the estimates.
logLik	The loglikelihood at the maximum.
support	The support of the density.
n	The number of observations.
call	The call as captured my match.call

References

Olkin, I., Petkau, A. J., & Zidek, J. V. (1981). A comparison of n Estimators for the binomial distribution. Journal of the American Statistical Association, 76(375), 637-642. https://doi.org/10.1080/01621459.1981.10477697

Johnson, N. L., Kemp, A. W., & Kotz, S. (2005). Univariate Discrete Distributions (3rd ed.). Wiley-Blackwell.

Carroll, R. J., & Lombard, F. (1985). A Note on N Estimators for the Binomial Distribution. Journal of the American Statistical Association, 80(390), 423-426. https://doi.org/10.1080/01621459.1985.10478134

DasGupta, A., & Rubin, H. (2005). Estimation of binomial parameters when both n,p are unknown. Journal of Statistical Planning and Inference, 130(1-2), 391-404. https://doi.org/10.1016/j.jspi.2004.02.019

See Also

[Binomial](#) for the density.

Examples

```
# The likelihood will often be unbounded.
## Not run:
mlbinom(ChickWeight$weight)

## End(Not run)
# Provide a size
mlbinom(ChickWeight$weight, size = 400)

# Or use mlpoiss, the limiting likelihood of the binomial.
mlpois(ChickWeight$weight)
```

---

mlburr   *Burr distribution maximum likelihood estimation*

Description

The maximum likelihood estimator fails to exist when the data contains no values strictly smaller than 1. Then the likelihood converges to the likelihood of the Pareto distribution in this case.

Usage

```
mlburr(x, na.rm = FALSE, ...)
```

Arguments

x	a (non-empty) numeric vector of data values.
na.rm	logical. Should missing values be removed?
...	currently affects nothing.

Details

This function estimates the only the shape parameters of the Burr distribution. The shape is set to 1.

For the density function of the Burr distribution see Burr.

Value

mlburr returns an object of class univariateML. This is a named numeric vector with maximum likelihood estimates for shape1 and shape2 and the following attributes:

model	The name of the model.
density	The density associated with the estimates.
logLik	The loglikelihood at the maximum.
support	The support of the density.
n	The number of observations.
call	The call as captured my match.call

References

Johnson, N. L., Kotz, S. and Balakrishnan, N. (1995) Continuous Univariate Distributions, Volume 1, Chapter 20. Wiley, New York.

See Also

Burr for the Burr density.

Examples

```
mlburr(abalone$length)
```

mlcauchy   *Cauchy distribution maximum likelihood estimation*

Description

Calculates the estimates using nlm and an exponential transform of the location parameter. If n < 5, an exact solution is reported. In the edge case where no maximum likelihood estimator exists and error is thrown.

Usage

```
mlcauchy(x, na.rm = FALSE, ...)
```

Arguments

x	a (non-empty) numeric vector of data values.
na.rm	logical. Should missing values be removed?
...	currently affects nothing.

Details

For the density function of the Cauchy distribution see Cauchy.

Value

mlcauchy returns an object of class univariateML. This is a named numeric vector with maximum likelihood estimates for location and scale and the following attributes:

model	The name of the model.
density	The density associated with the estimates.
logLik	The loglikelihood at the maximum.
support	The support of the density.
n	The number of observations.
call	The call as captured my match.call

References

Johnson, N. L., Kotz, S. and Balakrishnan, N. (1995) Continuous Univariate Distributions, Volume 1, Chapter 16. Wiley, New York.

See Also

Cauchy for the Cauchy density, nlm for the optimizer this function uses.

Examples

```
mlcauchy(airquality$Temp)
```

mldunif

Discrete uniform distribution maximum likelihood estimation

Description

For the density function of the Discrete uniform distribution see [DiscreteUniform](#).

Usage

```
mldunif(x, na.rm = FALSE, ...)
```

Arguments

x	a (non-empty) numeric vector of data values.
na.rm	logical. Should missing values be removed?
...	not in use.

Value

mldunif returns an object of [class](#) univariateML. This is a named numeric vector with maximum likelihood estimates for min and max and the following attributes:

model	The name of the model.
density	The density associated with the estimates.
logLik	The loglikelihood at the maximum.
support	The support of the density.
n	The number of observations.
call	The call as captured my match.call

References

Johnson, N. L., Kemp, A. W., & Kotz, S. (2005). Univariate Discrete Distributions (3rd ed.). Wiley-Blackwell.

See Also

[DiscreteUniform](#) for the density.

Examples

```
mldunif(corbet)
```

mlexp	*Exponential distribution maximum likelihood estimation*

Description

The maximum likelihood estimate of rate is the inverse sample mean.

Usage

```
mlexp(x, na.rm = FALSE, ...)
```

Arguments

x	a (non-empty) numeric vector of data values.
na.rm	logical. Should missing values be removed? If FALSE, the function fails when x contains missing values.
...	currently affects nothing.

Details

For the density function of the exponential distribution see Exponential.

Value

mlexp returns an object of class univariateML. This is a named numeric vector with maximum likelihood estimates for rate and the following attributes:

model	The name of the model.
density	The density associated with the estimates.
logLik	The loglikelihood at the maximum.
support	The support of the density.
n	The number of observations.
call	The call as captured by match.call
continuous	Is the density continuous or discrete?

References

Johnson, N. L., Kotz, S. and Balakrishnan, N. (1995) Continuous Univariate Distributions, Volume 1, Chapter 19. Wiley, New York.

See Also

Exponential for the exponential density.

Examples

```
mlexp(precip)
```

mlfatigue   *Gamma distribution maximum likelihood estimation*

Description

Uses Newton-Raphson to estimate the parameters of the Birnbaum–Saunders distribution. The parameter mu is set to 0, hence only alpha and beta are estmated.

Usage

```
mlfatigue(x, na.rm = FALSE, ...)
```

Arguments

x	a (non-empty) numeric vector of data values.
na.rm	logical. Should missing values be removed?
...	reltol is the relative accuracy requested, defaults to .Machine$double.eps^0.25. iterlim is a positive integer specifying the maximum number of iterations to be performed before the program is terminated (defaults to 100).

Details

For the density function of the Birnbaum–Saunders distribution see BirnbaumSaunders.

Value

mlfatigue returns an object of class univariateML. This is a named numeric vector with maximum likelihood estimates for alpha and beta, with mu=0, and the following attributes:

model	The name of the model.
density	The density associated with the estimates.
logLik	The loglikelihood at the maximum.
support	The support of the density.
n	The number of observations.
call	The call as captured my match.call

References

From, S. G., & Li, L. (2006). Estimation of the parameters of the Birnbaum<U+2013>Saunders distribution. Communications in Statistics: Theory and Methods, 35(12), 2157<U+2013>2169. https://doi.org/10.1080/03610920600853563

Examples

```
mlfatigue(precip)
```

---

mlgamma   *Gamma distribution maximum likelihood estimation*

Description

Uses Newton-Raphson to estimate the parameters of the Gamma distribution.

Usage

```
mlgamma(x, na.rm = FALSE, ...)
```

Arguments

x	a (non-empty) numeric vector of data values.
na.rm	logical. Should missing values be removed?
...	`reltol` is the relative accuracy requested, defaults to `.Machine$double.eps^0.25`. `iterlim` is a positive integer specifying the maximum number of iterations to be performed before the program is terminated (defaults to `100`).

Details

For the density function of the Gamma distribution see [GammaDist](#).

Value

mlgamma returns an object of [class](#) univariateML. This is a named numeric vector with maximum likelihood estimates for shape and rate and the following attributes:

model	The name of the model.
density	The density associated with the estimates.
logLik	The loglikelihood at the maximum.
support	The support of the density.
n	The number of observations.
call	The call as captured my match.call

References

Choi, S. C, and R. Wette. "Maximum likelihood estimation of the parameters of the gamma distribution and their bias." Technometrics 11.4 (1969): 683-690.

Johnson, N. L., Kotz, S. and Balakrishnan, N. (1995) Continuous Univariate Distributions, Volume 1, Chapter 17. Wiley, New York.

See Also

[GammaDist](#) for the Gamma density.

Examples

```
mlgamma(precip)
```

---

mlged	*Generalized Error distribution maximum likelihood estimation*

Description

Joint maximum likelihood estimation as implemented by [fGarch::gedFit](#).

Usage

```
mlged(x, na.rm = FALSE, ...)
```

Arguments

x	a (non-empty) numeric vector of data values.
na.rm	logical. Should missing values be removed?
...	currently affects nothing.

Details

For the density function of the Student t-distribution see [ged](#).

Value

mlged returns an object of [class](#) univariateML. This is a named numeric vector with maximum likelihood estimates for the parameters mean, sd, nu and the following attributes:

model	The name of the model.
density	The density associated with the estimates.
logLik	The loglikelihood at the maximum.
support	The support of the density.
n	The number of observations.
call	The call as captured my match.call

References

Nelson D.B. (1991); Conditional Heteroscedasticity in Asset Returns: A New Approach, Econometrica, 59, 347<U+2013>370.

Fernandez C., Steel M.F.J. (2000); On Bayesian Modelling of Fat Tails and Skewness, Preprint.

See Also

[ged](#) for the Student t-density.

Examples

```
mlged(precip)
```

---

mlgeom   *Geometric distribution maximum likelihood estimation*

Description

For the density function of the Geometric distribution see Geometric.

Usage

```
mlgeom(x, na.rm = FALSE, ...)
```

Arguments

x	a (non-empty) numeric vector of data values.
na.rm	logical. Should missing values be removed?
...	not in use.

Value

mlgeom returns an object of class univariateML. This is a named numeric vector with maximum likelihood estimates for prob and the following attributes:

model	The name of the model.
density	The density associated with the estimates.
logLik	The loglikelihood at the maximum.
support	The support of the density.
n	The number of observations.
call	The call as captured my match.call

References

Johnson, N. L., Kemp, A. W., & Kotz, S. (2005). Univariate Discrete Distributions (3rd ed.). Wiley-Blackwell.

See Also

Geometric for the density.

Examples

```
mlgeom(corbet)
```

---

mlgompertz   *Gompertz distribution maximum likelihood estimation*

Description

Uses Newton-Raphson to estimate the parameters of the Gompertz distribution.

Usage

```
mlgompertz(x, na.rm = FALSE, ...)
```

Arguments

x	a (non-empty) numeric vector of data values.
na.rm	logical. Should missing values be removed?
...	`reltol` is the relative accuracy requested, defaults to `.Machine$double.eps^0.25`. `iterlim` is a positive integer specifying the maximum number of iterations to be performed before the program is terminated (defaults to `100`).

Details

For some data sets the maximum likelihood estimator of b fails to exist since the root of the profile maximum likelihood equation is non-positive. The value `1e-06` is returned in this case, along with a warning.

For the density function of the Gompertz distribution see [Gompertz](#).

Value

`mlgompertz` returns an object of [class](#) `univariateML`. This is a named numeric vector with maximum likelihood estimates for a and b and the following attributes:

model	The name of the model.
density	The density associated with the estimates.
logLik	The loglikelihood at the maximum.
support	The support of the density.
n	The number of observations.
call	The call as captured my match.call

References

Lenart, A. (2012). The Gompertz distribution and Maximum Likelihood Estimation of its parameters - a revision. MPIDR WORKING PAPER WP 2012-008. https://www.demogr.mpg.de/papers/working/wp-2012-008.pdf

See Also

[Gompertz](#) for the Gompertz density.

Examples

```
mlgompertz(precip)
```

---

mlgumbel   *Gumbel distribution maximum likelihood estimation*

Description

Uses Newton-Raphson to estimate the parameters of the Gumbel distribution.

Usage

```
mlgumbel(x, na.rm = FALSE, ...)
```

Arguments

x	a (non-empty) numeric vector of data values.
na.rm	logical. Should missing values be removed?
...	reltol is the relative accuracy requested, defaults to .Machine$double.eps^0.25. iterlim is a positive integer specifying the maximum number of iterations to be performed before the program is terminated (defaults to 100).

Details

For the density function of the Gumbel distribution see [Gumbel](#).

Value

mlgumbel returns an object of [class](#) univariateML. This is a named numeric vector with maximum likelihood estimates for mu and s and the following attributes:

model	The name of the model.
density	The density associated with the estimates.
logLik	The loglikelihood at the maximum.
support	The support of the density.
n	The number of observations.
call	The call as captured my match.call

shape and sigma.

References

Johnson, N. L., Kotz, S. and Balakrishnan, N. (1995) Continuous Univariate Distributions, Volume 2, Chapter 22. Wiley, New York.

See Also

Gumbel for the Gumbel density.

Examples

```
mlgumbel(precip)
```

| mlinvburr | *Inverse Burr distribution maximum likelihood estimation* |

Description

The maximum likelihood estimator fails to exist when the data contains no values strictly greater than 1. Then the likelihood converges to the likelihood of the Pareto distribution in this case.

Usage

```
mlinvburr(x, na.rm = FALSE, ...)
```

Arguments

x	a (non-empty) numeric vector of data values.
na.rm	logical. Should missing values be removed?
...	currently affects nothing.

Details

`mlinvburr(x)` calls `mlburr(1/x)` internally.

For the density function of the Inverse Burr distribution see Inverse Burr.

Value

mlburr returns an object of class univariateML. This is a named numeric vector with maximum likelihood estimates for shape1 and shape2 and the following attributes:

model	The name of the model.
density	The density associated with the estimates.
logLik	The loglikelihood at the maximum.
support	The support of the density.
n	The number of observations.
call	The call as captured my match.call

References

Johnson, N. L., Kotz, S. and Balakrishnan, N. (1995) Continuous Univariate Distributions, Volume 1, Chapter 20. Wiley, New York.

See Also

Inverse Burr for the Inverse Burr density.

Examples

```
mlburr(abalone$length)
```

---

mlinvgamma	*Inverse Gamma distribution maximum likelihood estimation*

Description

Transforms the data and uses Newton-Raphson to estimate the parameters of the Gamma distribution.

Usage

```
mlinvgamma(x, na.rm = FALSE, ...)
```

Arguments

x	a (non-empty) numeric vector of data values.
na.rm	logical. Should missing values be removed?
...	passed to mlgamma.

Details

For the density function of the inverse Gamma distribution see InvGamma.

Value

A named numeric vector with maximum likelihood estimates for alpha and beta.

References

Choi, S. C, and R. Wette. "Maximum likelihood estimation of the parameters of the gamma distribution and their bias." Technometrics 11.4 (1969): 683-690.

Johnson, N. L., Kotz, S. and Balakrishnan, N. (1995) Continuous Univariate Distributions, Volume 1, Chapter 17. Wiley, New York.

Witkovsky, V. (2001). "Computing the Distribution of a Linear Combination of Inverted Gamma Variables". Kybernetika. 37 (1): 79<U+2013>90

See Also

[InvGamma](#) for the Inverse Gamma density.

Examples

```
mlinvgamma(precip)
```

---

mlinvgauss	*Inverse Gaussian (Wald) maximum likelihood estimation*

Description

The maximum likelihood estimate of mean is the empirical mean and the maximum likelihood estimate of 1/shape is the difference between the mean of reciprocals and the reciprocal of the mean.

Usage

```
mlinvgauss(x, na.rm = FALSE, ...)
```

Arguments

x	a (non-empty) numeric vector of data values.
na.rm	logical. Should missing values be removed?
...	currently affects nothing.

Details

For the density function of the Inverse Gamma distribution see [InverseGaussian](#).

Value

mlinvgauss returns an object of [class](#) univariateML. This is a named numeric vector with maximum likelihood estimates for mean and shape and the following attributes:

model	The name of the model.
density	The density associated with the estimates.
logLik	The loglikelihood at the maximum.
support	The support of the density.
n	The number of observations.
call	The call as captured my match.call

References

Johnson, N. L., Kotz, S. and Balakrishnan, N. (1995) Continuous Univariate Distributions, Volume 1, Chapter 15. Wiley, New York.

See Also

[InverseGaussian](#) for the Inverse Gaussian density.

Examples

```
mlinvgauss(precip)
```

---

mlinvweibull   *Inverse Weibull distribution maximum likelihood estimation*

Description

The maximum likelihood estimate of shape and rate are calculated by calling [mlweibull](#) on the transformed data.

Usage

```
mlinvweibull(x, na.rm = FALSE, ...)
```

Arguments

x	a (non-empty) numeric vector of data values.
na.rm	logical. Should missing values be removed?
...	passed to [mlweibull](#).

Details

For the density function of the log normal distribution see [InverseWeibull](#).

Value

mlinvweibull returns an object of [class](#) univariateML. This is a named numeric vector with maximum likelihood estimates for shape and rate and the following attributes:

model	The name of the model.
density	The density associated with the estimates.
logLik	The loglikelihood at the maximum.
support	The support of the density.
n	The number of observations.
call	The call as captured my match.call

References

Kleiber, C. and Kotz, S. (2003), Statistical Size Distributions in Economics and Actuarial Sciences, Wiley.

Klugman, S. A., Panjer, H. H. and Willmot, G. E. (2012), Loss Models, From Data to Decisions, Fourth Edition, Wiley.

Dutang, C., Goulet, V., & Pigeon, M. (2008). actuar: An R package for actuarial science. Journal of Statistical Software, 25(7), 1-37.

See Also

[InverseWeibull](#) for the Inverse Weibull density.

Examples

```
mlinvweibull(precip)
```

---

mlkumar   *Kumaraswamy distribution maximum likelihood estimation*

Description

Uses Newton-Raphson to estimate the parameters of the Kumaraswamy distribution.

Usage

```
mlkumar(x, na.rm = FALSE, ...)
```

Arguments

x	a (non-empty) numeric vector of data values.
na.rm	logical. Should missing values be removed?
...	a0 is an optional starting value for the a parameter. reltol is the relative accuracy requested, defaults to .Machine$double.eps^0.25. iterlim is a positive integer specifying the maximum number of iterations to be performed before the program is terminated (defaults to 100).

Details

For the density function of the Kumaraswamy distribution see [Kumaraswamy](#).

Value

mlkumar returns an object of [class](#) univariateML. This is a named numeric vector with maximum
likelihood estimates for a and b and the following attributes:

model	The name of the model.
density	The density associated with the estimates.
logLik	The loglikelihood at the maximum.
support	The support of the density.
n	The number of observations.
call	The call as captured my match.call

References

Jones, M. C. "Kumaraswamy's distribution: A beta-type distribution with some tractability advan-
tages." Statistical Methodology 6.1 (2009): 70-81.

Kumaraswamy, Ponnambalam. "A generalized probability density function for double-bounded
random processes." Journal of Hydrology 46.1-2 (1980): 79-88.

See Also

[Kumaraswamy](#) for the Kumaraswamy density.

Examples

```
AIC(mlkumar(USArrests$Rape / 100))
```

---

mllaplace	*Laplace distribution maximum likelihood estimation*

Description

The maximum likelihood estimate of mu is the sample median while the maximum likelihood esti-
mate of sigma is mean absolute deviation from the median.

Usage

```
mllaplace(x, na.rm = FALSE, ...)
```

Arguments

x	a (non-empty) numeric vector of data values.
na.rm	logical. Should missing values be removed?
...	currently affects nothing.

Details

For the density function of the Laplace distribution see [Laplace](#).

Value

mllaplace returns an object of [class](#) univariateML. This is a named numeric vector with maximum likelihood estimates for mu and sigma and the following attributes:

model	The name of the model.
density	The density associated with the estimates.
logLik	The loglikelihood at the maximum.
support	The support of the density.
n	The number of observations.
call	The call as captured my match.call

References

Johnson, N. L., Kotz, S. and Balakrishnan, N. (1995) Continuous Univariate Distributions, Volume 2, Chapter 24. Wiley, New York.

See Also

[Laplace](#) for the Laplace density.

Examples

```
mllaplace(precip)
```

---

mllgamma   *Log-gamma distribution maximum likelihood estimation*

Description

The maximum likelihood estimate of shapelog and ratelog are calculated by calling [mlgamma()](#) on the transformed data.

Usage

```
mllgamma(x, na.rm = FALSE, ...)
```

Arguments

x	a (non-empty) numeric vector of data values.
na.rm	logical. Should missing values be removed?
...	passed to [mlgamma](#).

Details

For the density function of the log normal distribution see [Loggamma](#).

Value

mllgamma returns an object of [class](#) univariateML. This is a named numeric vector with maximum likelihood estimates for shapelog and ratelog and the following attributes:

model	The name of the model.
density	The density associated with the estimates.
logLik	The loglikelihood at the maximum.
support	The support of the density.
n	The number of observations.
call	The call as captured my match.call

References

Hogg, R. V. and Klugman, S. A. (1984), Loss Distributions, Wiley.

Dutang, C., Goulet, V., & Pigeon, M. (2008). actuar: An R package for actuarial science. Journal of Statistical Software, 25(7), 1-37.

See Also

[Loggamma](#) for the log normal density.

Examples

```
mllgamma(precip)
```

---

mllgser   *Logarithmic series distribution maximum likelihood estimation*

Description

For the density function of the Logarithmic series distribution see [Logarithmic series](#). For an example data set, see [corbet](#).

Usage

```
mllgser(x, na.rm = FALSE, ...)
```

Arguments

x	a (non-empty) numeric vector of data values.
na.rm	logical. Should missing values be removed?
...	Not in use.

Value

mllgser returns an object of [class](#) univariateML. This is a named numeric vector with maximum likelihood estimates for theta.

model	The name of the model.
density	The density associated with the estimates.
logLik	The loglikelihood at the maximum.
support	The support of the density.
n	The number of observations.
call	The call as captured my match.call

References

Fisher, R. A., Corbet, A. S., & Williams, C. B. (1943). The relation between the number of species and the number of individuals in a random sample of an animal population. The Journal of Animal Ecology, 12(1), 42. https://doi.org/10.2307/1411

Johnson, N. L., Kemp, A. W., & Kotz, S. (2005). Univariate Discrete Distributions (3rd ed.). Wiley-Blackwell.

See Also

[Logarithmic series](#) for the density.

Examples

```
theta_hat <- mllgser(corbet)

# The corbert data contains observations from 1 to 24.
observed <- table(corbet)

# The chi square test evaluated at the maximum likelihood is highly significant.
expected <- extraDistr::dlgser(1:24, theta_hat)
chisq.test(observed, p = expected / sum(expected))

# But chi square test evaluated at 0.997 (used in Corbet) is not.
expected <- extraDistr::dlgser(1:24, 0.997)
chisq.test(observed, p = expected / sum(expected))

# The chi square for `dzipf` is similar.
expected <- sads::dzipf(1:24, mlzipf(corbet)[1], mlzipf(corbet)[2]) * length(corbet)
chisq.test(observed, p = expected / sum(expected))
```

---

mlllogis   *Log-logistic distribution maximum likelihood estimation*

Description

The maximum likelihood estimate of shape and rate are calculated by transforming the data back
to the logistic model and applying mllogis.

Usage

```
mlllogis(x, na.rm = FALSE, ...)
```

Arguments

x	a (non-empty) numeric vector of data values.
na.rm	logical. Should missing values be removed?
...	passed to mllogis.

Details

For the density function of the log-logistic distribution see Loglogistic

Value

mllogis returns an object of class univariateML. This is a named numeric vector with maximum
likelihood estimates for shape and rate and the following attributes:

model	The name of the model.
density	The density associated with the estimates.
logLik	The loglikelihood at the maximum.
support	The support of the density.
n	The number of observations.
call	The call as captured my match.call

References

Kleiber, C. and Kotz, S. (2003), Statistical Size Distributions in Economics and Actuarial Sciences,
Wiley.

Klugman, S. A., Panjer, H. H. and Willmot, G. E. (2012), Loss Models, From Data to Decisions,
Fourth Edition, Wiley.

Dutang, C., Goulet, V., & Pigeon, M. (2008). actuar: An R package for actuarial science. Journal
of Statistical Software, 25(7), 1-37.

See Also

Loglogistic for the log-logistic density.

Examples

```
mllnorm(precip)
```

---

mllnorm   *Log-normal distribution maximum likelihood estimation*

Description

The maximum likelihood estimate of meanlog is the empirical mean of the log-transformed data
and the maximum likelihood estimate of sdlog is the square root of the biased sample variance
based on the log-transformed data.

Usage

```
mllnorm(x, na.rm = FALSE, ...)
```

Arguments

x	a (non-empty) numeric vector of data values.
na.rm	logical. Should missing values be removed?
...	currently affects nothing.

Details

For the density function of the log normal distribution see [Lognormal](#).

Value

mllnorm returns an object of [class](#) univariateML. This is a named numeric vector with maximum
likelihood estimates for meanlog and sdlog and the following attributes:

model	The name of the model.
density	The density associated with the estimates.
logLik	The loglikelihood at the maximum.
support	The support of the density.
n	The number of observations.
call	The call as captured my match.call

References

Johnson, N. L., Kotz, S. and Balakrishnan, N. (1995) Continuous Univariate Distributions, Volume
1, Chapter 14. Wiley, New York.

See Also

[Lognormal](#) for the log normal density.

Examples

```
mllnorm(precip)
```

---

mllogis   *Logistic distribution maximum likelihood estimation*

Description

Calculates the estimates using `nlm` with an exponential transform of the location parameter.

Usage

```
mllogis(x, na.rm = FALSE, ...)
```

Arguments

x	a (non-empty) numeric vector of data values.
na.rm	logical. Should missing values be removed?
...	currently affects nothing.

Details

For the density function of the logistic distribution see [Logistic](#).

Value

mllogis returns an object of [class](#) univariateML. This is a named numeric vector with maximum likelihood estimates for location and scale and the following attributes:

model	The name of the model.
density	The density associated with the estimates.
logLik	The loglikelihood at the maximum.
support	The support of the density.
n	The number of observations.
call	The call as captured my match.call

References

Johnson, N. L., Kotz, S. and Balakrishnan, N. (1995) Continuous Univariate Distributions, Volume 2, Chapter 23. Wiley, New York.

See Also

[Logistic](#) for the Logistic density, [nlm](#) for the optimizer this function uses.

Examples

```
mllogis(precip)
```

---

mllogitnorm   *Logit-Normal distribution maximum likelihood estimation*

Description

The maximum likelihood estimate of mu is the empirical mean of the logit transformed data and the maximum likelihood estimate of sigma is the square root of the logit transformed biased sample variance.

Usage

```
mllogitnorm(x, na.rm = FALSE, ...)
```

Arguments

x	a (non-empty) numeric vector of data values.
na.rm	logical. Should missing values be removed?
...	currently affects nothing.

Details

For the density function of the logit-normal distribution see dlogitnorm.

Value

mllogitnorm returns an object of class univariateML. This is a named numeric vector with maximum likelihood estimates for mu and sigma and the following attributes:

model	The name of the model.
density	The density associated with the estimates.
logLik	The loglikelihood at the maximum.
support	The support of the density.
n	The number of observations.
call	The call as captured my match.call

References

Atchison, J., & Shen, S. M. (1980). Logistic-normal distributions: Some properties and uses. Biometrika, 67(2), 261-272.

See Also

Normal for the normal density.

Examples

```
AIC(mllogitnorm(USArrests$Rape / 100))
```

---

mllomax   *Lomax distribution maximum likelihood estimation*

Description

Uses Newton-Raphson to estimate the parameters of the Lomax distribution.

Usage

```
mllomax(x, na.rm = FALSE, ...)
```

Arguments

x	a (non-empty) numeric vector of data values.
na.rm	logical. Should missing values be removed?
...	lambda0 an optional starting value for the lambda parameter. reltol is the relative accuracy requested, defaults to .Machine$double.eps^0.25. iterlim is a positive integer specifying the maximum number of iterations to be performed before the program is terminated (defaults to 100).

Details

For the density function of the Lomax distribution see Lomax.

The likelihood estimator of the Lomax distribution is unbounded when mean(x^2) < 2*mean(x)^2. When this happens, the likelihood converges to an exponential distribution with parameter equal to the mean of the data. This is the natural limiting case for the Lomax distribution, and it is reasonable to use mlexp in this case.

Value

mllomax returns an object of class univariateML. This is a named numeric vector with maximum likelihood estimates for lambda and kappa and the following attributes:

model	The name of the model.
density	The density associated with the estimates.
logLik	The loglikelihood at the maximum.
support	The support of the density.
n	The number of observations.
call	The call as captured my match.call

References

Kleiber, Christian; Kotz, Samuel (2003), Statistical Size Distributions in Economics and Actuarial Sciences, Wiley Series in Probability and Statistics, 470, John Wiley & Sons, p. 60

See Also

[Lomax](#) for the Lomax density.

Examples

```
set.seed(3)
mllomax(extraDistr::rlomax(100, 2, 4))

# The maximum likelihood estimator may fail if the data is exponential.
## Not run:
set.seed(5)
mllomax(rexp(10))

## End(Not run)
```

---

mlnaka   *Nakagami distribution maximum likelihood estimation*

Description

The maximum likelihood estimates of `shape` and `scale` are calculated by calling `mlgamma` on the transformed data.

Usage

```
mlnaka(x, na.rm = FALSE, ...)
```

Arguments

x	a (non-empty) numeric vector of data values.
na.rm	logical. Should missing values be removed?
...	passed to [mlgamma](#).

Details

For the density function of the Nakagami distribution see [Nakagami](#).

Value

`mlgamma` returns an object of [class](#) `univariateML`. This is a named numeric vector with maximum likelihood estimates for `shape` and `rate` and the following attributes:

model	The name of the model.
density	The density associated with the estimates.
logLik	The loglikelihood at the maximum.
support	The support of the density.
n	The number of observations.
call	The call as captured by `match.call`
continuous	Is the density continuous or discrete?

References

Choi, S. C, and R. Wette. "Maximum likelihood estimation of the parameters of the gamma distribution and their bias." Technometrics 11.4 (1969): 683-690.

Johnson, N. L., Kotz, S. and Balakrishnan, N. (1995) Continuous Univariate Distributions, Volume 1, Chapter 17. Wiley, New York.

See Also

[Nakagami](#) for the Nakagami distribution. [GammaDist](#) for the closely related Gamma density. See [mlgamma](#) for the machinery underlying this function.

Examples

```
mlgamma(precip)
```

---

mlnbinom	*Negative binomial distribution maximum likelihood estimation*

Description

For the density function of the Negative binomial distribution see [Negative binomial](#).

Usage

```
mlnbinom(x, na.rm = FALSE, ...)
```

Arguments

x	a (non-empty) numeric vector of data values.
na.rm	logical. Should missing values be removed?
...	The arguments size can be specified to only return the ml of prob. reltol is the relative accuracy requested, defaults to .Machine$double.eps^0.25. iterlim is a positive integer specifying the maximum number of iterations to be performed before the program is terminated (defaults to 100).

Value

mlnbinom returns an object of [class](#) univariateML. This is a named numeric vector with maximum likelihood estimates for size and prob and the following attributes:

model	The name of the model.
density	The density associated with the estimates.
logLik	The loglikelihood at the maximum.
support	The support of the density.
n	The number of observations.
call	The call as captured my match.call

References

Johnson, N. L., Kemp, A. W., & Kotz, S. (2005). Univariate Discrete Distributions (3rd ed.). Wiley-Blackwell.

See Also

Negative binomial for the density.

Examples

```
mlnbinom(corbet)
```

---

mlnorm   *Normal distribution maximum likelihood estimation*

Description

The maximum likelihood estimate of mean is the empirical mean and the maximum likelihood estimate of sd is the square root of the biased sample variance.

Usage

```
mlnorm(x, na.rm = FALSE, ...)
```

Arguments

x	a (non-empty) numeric vector of data values.
na.rm	logical. Should missing values be removed?
...	currently affects nothing.

Details

For the density function of the normal distribution see Normal.

Value

mlnorm returns an object of class univariateML. This is a named numeric vector with maximum likelihood estimates for mean and sd and the following attributes:

model	The name of the model.
density	The density associated with the estimates.
logLik	The loglikelihood at the maximum.
support	The support of the density.
n	The number of observations.
call	The call as captured my match.call

References

Johnson, N. L., Kotz, S. and Balakrishnan, N. (1995) Continuous Univariate Distributions, Volume 1, Chapter 13. Wiley, New York.

See Also

[Normal](#) for the normal density.

Examples

```
mlnorm(precip)
```

---

mlparalogis	*Paralogistic distribution maximum likelihood estimation*

Description

This function estimates the only the shape parameters of the Paralogistic distribution. The rate is set to 1.

Usage

```
mlparalogis(x, na.rm = FALSE, ...)
```

Arguments

x	a (non-empty) numeric vector of data values.
na.rm	logical. Should missing values be removed?
...	currently affects nothing.

Details

For the density function of the Paralogistic distribution see [Paralogistic](#).

Value

mlparalogis returns an object of [class](#) univariateML. This is a named numeric vector with maximum likelihood estimates for shape and the following attributes:

model	The name of the model.
density	The density associated with the estimates.
logLik	The loglikelihood at the maximum.
support	The support of the density.
n	The number of observations.
call	The call as captured my match.call

References

Kleiber, C. and Kotz, S. (2003), Statistical Size Distributions in Economics and Actuarial Sciences, Wiley. Klugman, S. A., Panjer, H. H. and Willmot, G. E. (2012), Loss Models, From Data to Decisions, Fourth Edition, Wiley.

See Also

[Paralogistic](#) for the paralogistic density.

Examples

```
mlparalogis(abalone$length)
```

---

mlpareto	*Pareto distribution maximum likelihood estimation*

Description

The maximum likelihood estimate of b is the minimum of x and the maximum likelihood estimate of a is 1/(mean(log(x)) - log(b)).

Usage

```
mlpareto(x, na.rm = FALSE, ...)
```

Arguments

x	a (non-empty) numeric vector of data values.
na.rm	logical. Should missing values be removed?
...	currently affects nothing.

Details

For the density function of the Pareto distribution see [Pareto](#).

Value

mlpareto returns an object of [class](#) univariateML. This is a named numeric vector with maximum likelihood estimates for a and b and the following attributes:

model	The name of the model.
density	The density associated with the estimates.
logLik	The loglikelihood at the maximum.
support	The support of the density.
n	The number of observations.
call	The call as captured my match.call

References

Johnson, N. L., Kotz, S. and Balakrishnan, N. (1995) Continuous Univariate Distributions, Volume 1, Chapter 20. Wiley, New York.

See Also

Pareto for the Pareto density.

Examples

```
mlpareto(precip)
```

---

mlpois   *Poisson distribution maximum likelihood estimation*

Description

The maximum likelihood estimate of `lambda` is the empirical mean.

Usage

```
mlpois(x, na.rm = FALSE, ...)
```

Arguments

x	a (non-empty) numeric vector of data values.
na.rm	logical. Should missing values be removed?
...	currently affects nothing.

Details

For the density function of the Poisson distribution see Poisson.

Value

`mlpois` returns an object of class `univariateML`. This is a named numeric vector with maximum likelihood estimates for `lambda` and the following attributes:

model	The name of the model.
density	The density associated with the estimates.
logLik	The loglikelihood at the maximum.
support	The support of the density.
n	The number of observations.
call	The call as captured my match.call

References

Johnson, N. L., Kemp, A. W., & Kotz, S. (2005). Univariate Discrete Distributions (3rd ed.). Wiley-Blackwell.

See Also

Poisson for the Poisson density.

Examples

```
mlpois(ChickWeight$weight)
```

---

mlpower	*Power distribution maximum likelihood estimation*

Description

The maximum likelihood estimate of `alpha` is the maximum of `x + epsilon` (see the details) and the maximum likelihood estimate of `beta` is `1/(log(alpha)-mean(log(x)))`.

Usage

```
mlpower(x, na.rm = FALSE, ...)
```

Arguments

x	a (non-empty) numeric vector of data values.
na.rm	logical. Should missing values be removed?
...	`epsilon` is a positive number added to `max(x)` as an to the maximum likelihood. Defaults to `.Machine$double.eps^0.5`.

Details

For the density function of the power distribution see PowerDist. The maximum likelihood estimator of `alpha` does not exist, strictly speaking. This is because x is supported `c(0, alpha)` with an open endpoint on alpha in the `extraDistr` implementation of `dpower`. If the endpoint was closed, `max(x)` would have been the maximum likelihood estimator. To overcome this problem, we add a possibly user specified `epsilon` to `max(x)`.

Value

`mlpower` returns an object of class `univariateML`. This is a named numeric vector with maximum likelihood estimates for `alpha` and `beta` and the following attributes:

model	The name of the model.
density	The density associated with the estimates.
logLik	The loglikelihood at the maximum.

support	The support of the density.
n	The number of observations.
call	The call as captured my match.call

References

Arslan, G. "A new characterization of the power distribution." Journal of Computational and Applied Mathematics 260 (2014): 99-102.

See Also

[PowerDist](#) for the power density. [extraDistr::Pareto](#) for the closely related Pareto distribution.

Examples

```
mlpower(precip)
```

---

mlrayleigh	*Rayleigh distribution maximum likelihood estimation*

Description

Calculates the `sigma` parameter as the square root of half the empirical second moment.

Usage

```
mlrayleigh(x, na.rm = FALSE, ...)
```

Arguments

x	a (non-empty) numeric vector of data values.
na.rm	logical. Should missing values be removed?
...	currently affects nothing.

Details

For the density function of the Rayleigh distribution see [Rayleigh](#).

Value

mlrayleigh returns an object of [class](#) univariateML. This is a named numeric vector with maximum likelihood estimates for `sigma` and the following attributes:

model	The name of the model.
density	The density associated with the estimates.
logLik	The loglikelihood at the maximum.
support	The support of the density.
n	The number of observations.
call	The call as captured my match.call

References

Johnson, N. L., Kotz, S. and Balakrishnan, N. (1995) Continuous Univariate Distributions, Volume 1, Chapter 18. Wiley, New York.

See Also

Rayleigh for the Rayleigh density.

Examples

```
mlrayleigh(precip)
```

---

mlsged   *Skew Generalized Error distribution maximum likelihood estimation*

Description

Joint maximum likelihood estimation as implemented by fGarch::sgedFit.

Usage

```
mlsged(x, na.rm = FALSE, ...)
```

Arguments

x	a (non-empty) numeric vector of data values.
na.rm	logical. Should missing values be removed?
...	currently affects nothing.

Details

For the density function of the Student t-distribution see sged.

Value

mlsged returns an object of class univariateML. This is a named numeric vector with maximum likelihood estimates for the parameters mean, sd, nu, xi, and the following attributes:

model	The name of the model.
density	The density associated with the estimates.
logLik	The loglikelihood at the maximum.
support	The support of the density.
n	The number of observations.
call	The call as captured my match.call

References

Nelson D.B. (1991); Conditional Heteroscedasticity in Asset Returns: A New Approach, Econometrica, 59, 347<U+2013>370.

Fernandez C., Steel M.F.J. (2000); On Bayesian Modelling of Fat Tails and Skewness, Preprint.

See Also

sged for the Student t-density.

Examples

```
mlsged(precip)
```

---

mlsnorm   *Skew Normal distribution maximum likelihood estimation*

Description

Joint maximum likelihood estimation as implemented by fGarch::snormFit.

Usage

```
mlsnorm(x, na.rm = FALSE, ...)
```

Arguments

x	a (non-empty) numeric vector of data values.
na.rm	logical. Should missing values be removed?
...	currently affects nothing.

Details

For the density function of the Student t distribution see dsnorm.

Value

mlsnorm returns an object of class univariateML. This is a named numeric vector with maximum likelihood estimates for the parameters mean, sd, xi and the following attributes:

model	The name of the model.
density	The density associated with the estimates.
logLik	The loglikelihood at the maximum.
support	The support of the density.
n	The number of observations.
call	The call as captured my match.call

References

Fernandez C., Steel M.F.J. (2000); On Bayesian Modelling of Fat Tails and Skewness, Preprint.

See Also

dsnorm for the Student-t density.

Examples

```
mlsnorm(precip)
```

---

mlsstd   *Skew Student t-distribution maximum likelihood estimation*

Description

Joint maximum likelihood estimation as implemented by fGarch::sstdFit.

Usage

```
mlsstd(x, na.rm = FALSE, ...)
```

Arguments

x	a (non-empty) numeric vector of data values.
na.rm	logical. Should missing values be removed?
...	currently affects nothing.

Details

For the density function of the skew Student t-distribution see sstd.

Value

mlsstd returns an object of class univariateML. This is a named numeric vector with maximum likelihood estimates for the parameters mean, sd, nu, xi and the following attributes:

model	The name of the model.
density	The density associated with the estimates.
logLik	The loglikelihood at the maximum.
support	The support of the density.
n	The number of observations.
call	The call as captured my match.call

References

Fernandez C., Steel M.F.J. (2000); On Bayesian Modelling of Fat Tails and Skewness, Preprint.

See Also

sstd for the Skew Student t-density.

Examples

```
mlsstd(precip)
```

---

mlstd   *Student-t distribution maximum likelihood estimation*

Description

Joint maximum likelihood estimation as implemented by fGarch::stdFit.

Usage

```
mlstd(x, na.rm = FALSE, ...)
```

Arguments

x	a (non-empty) numeric vector of data values.
na.rm	logical. Should missing values be removed?
...	currently affects nothing.

Details

For the density function of the Student t-distribution see std.

Value

mlstd returns an object of class univariateML. This is a named numeric vector with maximum likelihood estimates for the parameters mean, sd, nu and the following attributes:

model	The name of the model.
density	The density associated with the estimates.
logLik	The loglikelihood at the maximum.
support	The support of the density.
n	The number of observations.
call	The call as captured my match.call

References

Johnson, N. L., Kotz, S. and Balakrishnan, N. (1995) Continuous Univariate Distributions, Volume 1, Chapter 13. Wiley, New York.

See Also

[std](#) for the Student-t density.

Examples

```
mlstd(precip)
```

---

mlunif   *Uniform distribution maximum likelihood estimation*

Description

The estimates are `min(x)` and `max(x)`.

Usage

```
mlunif(x, na.rm = FALSE, ...)
```

Arguments

x	a (non-empty) numeric vector of data values.
na.rm	logical. Should missing values be removed?
...	currently affects nothing.

Details

For the density function of the logistic distribution see [Uniform](#).

Value

mlunif returns an object of [class](#) univariateML. This is a named numeric vector with maximum likelihood estimates for min and max and the following attributes:

model	The name of the model.
density	The density associated with the estimates.
logLik	The loglikelihood at the maximum.
support	The support of the density.
n	The number of observations.
call	The call as captured my match.call

References

Johnson, N. L., Kotz, S. and Balakrishnan, N. (1995) Continuous Univariate Distributions, Volume 2, Chapter 26. Wiley, New York.

See Also

[Uniform](#) for the uniform density.

Examples

```
mlunif(precip)
```

---

mlweibull   *Weibull distribution maximum likelihood estimation*

Description

For the density function of the Weibull distribution see [Weibull](#).

Usage

```
mlweibull(x, na.rm = FALSE, ...)
```

Arguments

x	a (non-empty) numeric vector of data values.
na.rm	logical. Should missing values be removed?
...	shape0 is an optional starting value for the shape parameter. reltol is the relative accuracy requested, defaults to .Machine$double.eps^0.25. iterlim is a positive integer specifying the maximum number of iterations to be performed before the program is terminated (defaults to 100).

Details

Uses [mlgumbel](#) to estimate the parameters of the Weibull distribution.

Value

mlweibull returns an object of [class](#) univariateML. This is a named numeric vector with maximum likelihood estimates for shape and scale and the following attributes:

model	The name of the model.
density	The density associated with the estimates.
logLik	The loglikelihood at the maximum.
support	The support of the density.
n	The number of observations.
call	The call as captured my match.call

References

Johnson, N. L., Kotz, S. and Balakrishnan, N. (1995) Continuous Univariate Distributions, Volume 1, Chapter 21. Wiley, New York.

See Also

[Weibull](#) for the Weibull density.

Examples

```
BIC(mlweibull(precip))
```

---

mlzip	*Zero-inflated Poisson distribution maximum likelihood estimation*

Description

For the density function of the zeero-inflated Poisson distribution see [Zero-inflated Poisson distribution](#).

Usage

```
mlzip(x, na.rm = FALSE, ...)
```

Arguments

x	a (non-empty) numeric vector of data values.
na.rm	logical. Should missing values be removed?
...	Not currently in use.

Value

mlzip returns an object of [class](#) univariateML. This is a named numeric vector with maximum likelihood estimates for lambda and pi and the following attributes:

model	The name of the model.
density	The density associated with the estimates.
logLik	The loglikelihood at the maximum.
support	The support of the density.
n	The number of observations.
call	The call as captured my match.call

References

Johnson, N. L., Kemp, A. W., & Kotz, S. (2005). Univariate Discrete Distributions (3rd ed.). Wiley-Blackwell.

See Also

[Zero-inflated Poisson distribution](#) for the density.

Examples

```
mlzip(corbet)
```

---

mlzipf   *Zipf distribution maximum likelihood estimation*

Description

For the density function of the Zipf distribution see Zipf.

Usage

```
mlzipf(x, na.rm = FALSE, ...)
```

Arguments

x	a (non-empty) numeric vector of data values.
na.rm	logical. Should missing values be removed?
...	Not currently in use.

Details

This function follows the same format as every other function in the package, but most applications of Zipf's law use rank-abundance data. See, e.g., sads::fitzipf for estimation of this sort of data.

Value

mlzipf returns an object of class univariateML. This is a named numeric vector with maximum likelihood estimates for N and s and the following attributes:

model	The name of the model.
density	The density associated with the estimates.
logLik	The loglikelihood at the maximum.
support	The support of the density.
n	The number of observations.
call	The call as captured my match.call

References

Johnson, N. L., Kemp, A. W., & Kotz, S. (2005). Univariate Discrete Distributions (3rd ed.). Wiley-Blackwell.

See Also

Zipf for the density.

Examples

```
AIC(mlzipf(corbet)) # 2729.536
AIC(mllgser(corbet)) # 2835.883
```

---

model_select   *Fit multiple models and select the best fit*

Description

Selects the best model by log-likelihood, aic, or bic.

Usage

```
model_select(
  x,
  models = univariateML_models,
  criterion = c("AIC", "BIC", "logLik"),
  na.rm = FALSE,
  type = c("both", "discrete", "continuous"),
  return = c("best", "all"),
  ...
)
```

Arguments

x	a (non-empty) numeric vector of data values.
models	a character vector containing the distribution models to select from; see print(univariateML_models). Defaults to all implemented models.
criterion	the model selection criterion. Must be one of "AIC", "BIC", and "logLik", ignoring case. Defaults to "AIC".
na.rm	logical. Should missing values be removed?
type	Either "both", "discrete", or "continuous". The supplied models vector is restricted to the desired class.
return	character length 1. "univariateML" (default) if the function should return the single best model; "all" if a tibble data frame of all results should be returned, sorted by decreasing model performance.
...	unused.

Value

The return value depends on the return argument. For return = "best" (default), model_select returns an object of [class](#) univariateML

For return = "all", model_select returns a tibble data frame with the following columns:

model	The name of the model.

```
\verb{d_loglik, d_aic, d_bic}
                    See loglik, aic, bic.
```

p   Number of parameters fitted.

```
\verb{loglik, aic, bic}
```
                   The negative log-likelihood at the maximum, the aic, and the bic, respectively. The minimum of each of these is noted and then subtracted from each value to give their delta versions d_loglik, d_aic, d_bic

. So, the model with the lowest aic will have d_aic of 0; the d_aic of all the other models shows how much higher their aics are from the minimum. The same goes with d_loglik and d_bic.

ml   The internal code name for the model.

univariateML   The univariateML object for the model. This is return = "all", this object is returned for all tested models.

See Also

Johnson, N. L., Kotz, S. and Balakrishnan, N. (1995) Continuous Univariate Distributions, Volume 1, Chapter 17. Wiley, New York.

Examples

```
# Select among all possible continuous models.
model_select(precip, type = "continuous")

# View possible models to fit.
print(univariateML_models)

# Try out only gamma, Weibull, and exponential.
model_select(precip, c("gamma", "weibull", "exp"))

# Fit the discrete `corbet` data to all available discrete models
model_select(corbet, type = "discrete", return = "all")
```

---

plot.univariateML   *Plot, Lines and Points Methods for Maximum Likelihood Estimates*

Description

The plot, lines, and points methods for univariateML objects.

Usage

```
## S3 method for class 'univariateML'
plot(x, range = NULL, kind = c("d", "p", "q"), ...)

## S3 method for class 'univariateML'
```

```
lines(x, range = NULL, kind = c("d", "p", "q"), ...)

## S3 method for class 'univariateML'
points(x, range = NULL, kind = c("d", "p", "q"), ...)
```

Arguments

x	a univariateML object.
range	range of x values to plot, i.e. c(lower, upper).
kind	can be density, probability, or quantile.
...	parameters passed to plot, lines, or points.

Value

An invisible copy of x.

Examples

```
plot(mlweibull(datasets::precip), main = "Annual Precipitation in US Cities")
lines(mlgamma(datasets::precip), lty = 2)
rug(datasets::precip)
```

ProbabilityPlots *Probability Plots Using Maximum Likelihood Estimates*

Description

Make quantile-quantile plots and probability-probability plots using maximum likelihood estimation.

Usage

```
ppmlplot(y, obj, plot.it = TRUE, datax = FALSE, ...)

ppmlline(...)

ppmlpoints(y, obj, plot.it = TRUE, datax = TRUE, ...)

qqmlplot(y, obj, plot.it = TRUE, datax = FALSE, ...)

qqmlline(y, obj, datax = FALSE, probs = c(0.25, 0.75), qtype = 7, ...)

qqmlpoints(y, obj, plot.it = TRUE, datax = TRUE, ...)
```

Arguments

y	Numeric vector; The data to plot on the y axis when `datax` is `FALSE`.
obj	Either an `univariateML` object or a function that returns a `univariateML` object when called with y as its only argument.
plot.it	Logical; should the result be plotted?
datax	Logical; should y be plotted on the x-axis? Defaults to `FALSE` in `qqmlplot` and `ppmlplot` but `TRUE` in `qqmlpoints` and `ppmlpoints`.
...	Graphical parameters.
probs	Numeric vector of length two, representing probabilities. Corresponding quantile pairs define the line drawn.
qtype	The `type` of quantile computation used in `quantile`.

Details

`qqmlplot` produces a quantile-quantile plot (Q-Q plot) of the values in y with respect to the distribution defined by `obj`, which is either a `univariateML` object or a function returning a `univariateML` object when called with y. `qqmlline` adds a line to a "theoretical", quantile-quantile plot which passes through the `probs` quantiles, by default the first and third quartiles. `qqmlpoints`behaves like `stats::points` and adds a Q-Q plot to an existing plot.

`ppmlplot`, `ppmlline`, and `ppmlpoints` produce probability-probability plots (or P-P plots). They behave similarly to the quantile-quantile plot functions.

This function is modeled after [qqnorm](#).

Quantile-quantile plots and probability-probability plots are only supported for continuous distributions.

Graphical parameters may be given as arguments to all the functions below.

Value

For `qqmlplot`, `qqmlpoints`, `ppmlplot`, and `ppmlpoints`, a list with components x (plotted on the x axis) and y (plotted on the y axis). `qqmlline` and `ppmlline` returns nothing.

References

M. B. Wilk, R. Gnadadesikan, Probability plotting methods for the analysis for the analysis of data, Biometrika, Volume 55, Issue 1, March 1968, Pages 1–17, https://doi.org/10.1093/biomet/55.1.1

Examples

```
## Make a single probability plot with a line.

obj <- mlgamma(Nile)
qqmlplot(Nile, obj)
qqmlline(Nile, obj)

## Make multiple probability plots. datax = TRUE must be used to make this
## look good.
```

```
ppmlplot(airquality$Wind, mlgamma, main = "Many P-P plots")
ppmlpoints(airquality$Wind, mlexp, col = "red")
ppmlpoints(airquality$Wind, mlweibull, col = "purple")
ppmlpoints(airquality$Wind, mllnorm, col = "blue")
```

---

univariateML_construct

Construct univariateML *object.*

Description

Construct univariateML object.

Usage

```
univariateML_construct(estimates, name, params)
```

Arguments

estimates	The estimated parameters
name	Name of the ml*** function.
params	List of loglik, call, and n.

Value

Object of class univariateML

---

univariateML_metadata   *Metadata for* univariateML *models.*

Description

Metadata for univariateML models.

Usage

```
univariateML_metadata
```

Format

An object of class list of length 42.

univariateML_models   *Implemented models*

Description

Vector of all supported models in `univariateML`.

Usage

```
univariateML_models
```

Format

An object of class `character` of length 42.

Details

The currently supported models are mlbeta, mlbetapr, mlbinom, mlburr, mlcauchy, mldunif, mlexp, mlfatigue, mlgamma, mlged, mlgeom, mlgompertz, mlgumbel, mlinvburr, mlinvgamma, mlinvgauss, mlinvweibull, mlkumar, mllaplace, mllgamma, mllgser, mlllogis, mllnorm, mllogis, mllogitnorm, mllomax, mlnaka, mlnbinom, mlnorm, mlparalogis, mlpareto, mlpois, mlpower, mlrayleigh, mlsged, mlsnorm, mlsstd, mlstd, mlunif, mlweibull, mlzip, mlzipf

Index