Package: melt (via r-universe)

October 29, 2026

Type Package

Title Multiple Empirical Likelihood Tests

Version 1.11.4

Description Performs multiple empirical likelihood tests. It offers an easy-to-use interface and flexibility in specifying hypotheses and calibration methods, extending the framework to simultaneous inferences. The core computational routines are implemented using the 'Eigen' 'C++' library and 'RcppEigen' interface, with 'OpenMP' for parallel computation. Details of the testing procedures are provided in Kim, MacEachern, and Peruggia (2023) <doi:10.1080/10485252.2023.2206919>. A companion paper by Kim, MacEachern, and Peruggia (2024) <doi:10.18637/jss.v108.i05> is available for further information. This work was supported by the U.S. National Science Foundation under Grants No. SES-1921523 and DMS-2015552.

License GPL (>= 2)

URL https://docs.ropensci.org/melt/, https://github.com/ropensci/melt

BugReports https://github.com/ropensci/melt/issues

Depends R (>= 4.1.0)

Imports checkmate, graphics, methods, Rcpp, stats, utils

Suggests car, carData, covr, dplyr, ggplot2, ggridges, knitr, MASS, microbenchmark, multcomp, rmarkdown, spelling, testthat (>= 3.0.0), withr

LinkingTo BH, dqrng, Rcpp, RcppEigen

VignetteBuilder knitr

Config/testthat/edition 3

Encoding UTF-8

Language en-US

LazyData true

NeedsCompilation yes

RoxygenNote 7.3.1

Author Eunseop Kim [aut, cph, cre], Steven MacEachern [ctb, ths], Mario Peruggia [ctb, ths], Pierre Chausse [rev], Alex Stringer [rev]

Maintainer Eunseop Kim <markean@pm.me>

Repository https://cran.r-universe.dev

Date/Publication 2024-05-18 02:38:44 UTC

RemoteUrl https://github.com/cran/melt

RemoteRef HEAD

RemoteSha 1f0a23934c4b13db05753e1d86324a927461fa16

Contents

CEL-class . . . 3
chisq . . . 5
clothianidin . . . 6
coef . . . 7
confint . . . 8
confreg . . . 9
ConfregEL-class . . . 10
ControlEL-class . . . 10
conv . . . 11
critVal . . . 12
EL-class . . . 13
el_control . . . 15
el_eval . . . 16
el_glm . . . 18
el_lm . . . 21
el_mean . . . 23
el_sd . . . 24
eld . . . 25
ELD-class . . . 26
elmt . . . 27
ELMT-class . . . 28
elt . . . 29
ELT-class . . . 31
getDF . . . 32
getOptim . . . 33
GLM-class . . . 34
LM-class . . . 36
logL . . . 37
logLR . . . 38
logProb . . . 39
nobs . . . 40

plot . . . 41
print . . . 42
pVal . . . 43
QGLM-class . . . 45
SD-class . . . 46
sigTests . . . 47
summary . . . 48
SummaryEL-class . . . 49
SummaryELMT-class . . . 50
SummaryELT-class . . . 50
SummaryGLM-class . . . 51
SummaryLM-class . . . 52
SummaryQGLM-class . . . 53
thiamethoxam . . . 54
weights . . . 55

Index **56**

`CEL-class` *CEL class*

Description

S4 class for constrained empirical likelihood. It inherits from EL class. Note that the `optim` slot has constrained optimization results with respect to the parameters, not the Lagrange multiplier.

Details

Let $l(\theta)$ denote minus twice the empirical log-likelihood ratio function. We consider a linear hypothesis of the form

$$L\theta = r,$$

where the left-hand-side L is a q by p matrix and the right-hand-side r is a q-dimensional vector. Under some regularity conditions, $l(\theta)$ converges in distribution to χ^2_q under the constraint of hypothesis, i.e.,

$$\min_{\theta: L\theta = r} l(\theta) \to_d \chi^2_q.$$

Minimization of $l(\theta)$ with respect to θ is computationally expensive since it implicitly involves the evaluation step as described in EL. Further, depending on the form of $g(X_i, \theta)$ and the constraint, the optimization problem can be nonconvex and have multiple local minima. For this reason, the package **melt** only considers linear hypotheses and performs local minimization of $l(\theta)$ using projected gradient descent method. With the orthogonal projection matrix P and a step size γ, the algorithm updates θ as

$$\theta^{(k+1)} \leftarrow \theta^{(k)} - \gamma P \nabla l(\theta^{(k)}),$$

where $\nabla l(\theta^{(k)})$ denotes the gradient of l at $\theta^{(k)}$. The first order optimality condition is $P\nabla l(\theta) = 0$, which is used as the stopping criterion.

Slots

`optim` A list of the following optimization results:

- `par` A numeric vector of the solution to the constrained optimization problem.
- `lambda` A numeric vector of the Lagrange multipliers of the dual problem corresponding to `par`.
- `iterations` A single integer for the number of iterations performed.
- `convergence` A single logical for the convergence status.
- `cstr` A single logical for whether constrained EL optimization is performed or not.

`logp` A numeric vector of the log probabilities of the constrained empirical likelihood.

`logl` A single numeric of the constrained empirical log-likelihood.

`loglr` A single numeric of the constrained empirical log-likelihood ratio.

`statistic` A single numeric of minus twice the constrained empirical log-likelihood ratio with an asymptotic chi-square distribution.

`df` A single integer for the degrees of freedom of the statistic.

`pval` A single numeric for the p-value of the statistic.

`nobs` A single integer for the number of observations.

`npar` A single integer for the number of parameters.

`weights` A numeric vector of the re-scaled weights used for the model fitting.

`coefficients` A numeric vector of the maximum empirical likelihood estimates of the parameters.

`method` A single character for the method dispatch in internal functions.

`data` A numeric matrix of the data for the model fitting.

`control` An object of class ControlEL constructed by `el_control()`.

References

Adimari G, Guolo A (2010). "A Note on the Asymptotic Behaviour of Empirical Likelihood Statistics." *Statistical Methods & Applications*, **19**(4), 463–476. doi:10.1007/s1026001001379.

Qin J, Lawless J (1995). "Estimating Equations, Empirical Likelihood and Constraints on Parameters." *Canadian Journal of Statistics*, **23**(2), 145–159. doi:10.2307/3315441.

Examples

```
showClass("CEL")
```

chisq *Chi-square statistic*

Description

Extracts the chi-square statistic from a model.

Usage

```
## S4 method for signature 'EL'
chisq(object, ...)

## S4 method for signature 'ELMT'
chisq(object, ...)

## S4 method for signature 'ELT'
chisq(object, ...)

## S4 method for signature 'SummaryEL'
chisq(object, ...)

## S4 method for signature 'SummaryELMT'
chisq(object, ...)

## S4 method for signature 'SummaryELT'
chisq(object, ...)

## S4 method for signature 'SummaryLM'
chisq(object, ...)
```

Arguments

object	An object that contains the chi-square statistic.
...	Further arguments passed to methods.

Value

The form of the value returned by chisq() depends on the class of its argument.

Methods (by class)

- chisq(EL): Extracts the chi-square statistic.
- chisq(ELMT): Extracts the vector of chi-square statistics.
- chisq(ELT): Extracts the chi-square statistic.
- chisq(SummaryEL): Extracts the chi-square statistic.
- chisq(SummaryELMT): Extracts the vector of chi-square statistics.

- `chisq(SummaryELT)`: Extracts the chi-square statistic.
- `chisq(SummaryLM)`: Extracts the chi-square statistic for the overall test of the model.

See Also

EL, ELMT, ELT, `pVal()`

Examples

```
data("precip")
fit <- el_mean(precip, par = 40)
chisq(fit)
```

`clothianidin` *Clothianidin concentration in maize plants*

Description

A dataset summarizing field experiments result of seed treatments on clothianidin concentration.

Usage

```
data("clothianidin")
```

Format

A data frame with 102 observations and 3 variables:

blk New blocks constructed from original data. The format is 'days post planting_original block_year'.

trt Seed treatment.

clo Log transformed clothianidin concentration (µg).

Details

The original data is provided by Alford and Krupke (2017). Only some of the shoot region observations are taken from the original data and processed for illustration.

Source

Alford A, Krupke CH (2017). "Translocation of the Neonicotinoid Seed Treatment Clothianidin in Maize." *PLOS ONE*, **12**(3), 1–19. doi:10.1371/journal.pone.0173836.

Examples

```
data("clothianidin")
clothianidin
```

coef *Model coefficients*

Description

Extracts the maximum empirical likelihood estimates from a model.

Usage

```
## S4 method for signature 'EL'
coef(object, ...)

## S4 method for signature 'ELMT'
coef(object, ...)

## S4 method for signature 'SummaryEL'
coef(object, ...)

## S4 method for signature 'SummaryLM'
coef(object, ...)
```

Arguments

object	An object that contains the maximum empirical likelihood estimates.
...	Further arguments passed to methods.

Value

The form of the value returned by `coef()` depends on the class of its argument.

Methods (by class)

- `coef(EL)`: Extracts the numeric vector of the maximum empirical likelihood estimates.
- `coef(ELMT)`: Extracts the list of numeric vectors of the maximum empirical likelihood estimates. Each element of the list corresponds to a distinct hypothesis.
- `coef(SummaryEL)`: Extracts the numeric vector of the maximum empirical likelihood estimates.
- `coef(SummaryLM)`: Extracts a matrix with the results of significance tests.

See Also

EL, ELMT

Examples

```
data("mtcars")
fit <- el_lm(mpg ~ wt, data = mtcars)
coef(fit)
```

confint *Confidence interval for model parameters*

Description

Computes confidence intervals for one or more parameters in a model.

Usage

```
## S4 method for signature 'EL'
confint(object, parm, level = 0.95, cv = NULL, control = NULL)

## S4 method for signature 'ELMT'
confint(object, cv = NULL, control = NULL)
```

Arguments

object	An object that inherits from EL or ELMT.
parm	A specification of which parameters are to be given confidence intervals, either a vector of numbers or a vector of names. If missing, all parameters are considered.
level	A single numeric for the confidence level required. Defaults to 0.95.
cv	A single numeric for the critical value for calibration of empirical likelihood ratio statistic. Defaults to NULL and set to qchisq(level, 1L). If non-NULL, level is ignored.
control	An object of class ControlEL constructed by el_control(). Defaults to NULL and inherits the control slot in object.

Value

A matrix with columns giving lower and upper confidence limits for each parameter. In contrast to other methods that rely on studentization, the lower and upper limits obtained from empirical likelihood do not correspond to the (1 - level) / 2 and 1 - (1 - level) / 2 in %, respectively.

References

Owen A (1990). "Empirical Likelihood Ratio Confidence Regions." *The Annals of Statistics*, **18**(1), 90–120. doi:10.1214/aos/1176347494.

See Also

EL, ELMT, confreg(), elt(), el_control()

Examples

```
data("mtcars")
fit <- el_lm(mpg ~ ., data = mtcars)
confint(fit, parm = c(2, 3))
```

confreg *Confidence region for model parameters*

Description

Computes boundary points of a two-dimensional confidence region for model parameters.

Usage

```
## S4 method for signature 'EL'
confreg(object, parm, level = 0.95, cv = NULL, npoints = 50L, control = NULL)
```

Arguments

object	An object that inherits from EL.
parm	A specification of which parameters are to be given a confidence region, either a vector of numbers or a vector of names. It must be a vector of length two of the form c(x, y). If missing, the first two parameter in object are considered.
level	A single numeric for the confidence level required. Defaults to 0.95. It is ignored if cv is non-NULL.
cv	A single numeric for the critical value for calibration of empirical likelihood ratio statistic. Defaults to NULL and set to qchisq(level, 2L). It must be compatible with the th value in control.
npoints	A single integer for the number of boundary points to compute. Defaults to 50.
control	An object of class ControlEL constructed by el_control(). Defaults to NULL and inherits the control slot in object.

Value

An object of class ConfregEL.

References

Owen A (1990). "Empirical Likelihood Ratio Confidence Regions." *The Annals of Statistics*, **18**(1), 90–120. doi:10.1214/aos/1176347494.

See Also

EL, confint(), elt(), plot(), el_control()

Examples

```
data("mtcars")
fit <- el_lm(mpg ~ wt + qsec, data = mtcars)
cr <- confreg(fit, parm = c(2, 3), cv = qchisq(0.90, 2))
plot(cr)
```

ConfregEL-class *ConfregEL class*

Description

S4 class for confidence region. It inherits from "matrix".

Slots

estimates A numeric vector of length two for the parameter estimates.

level A single numeric for the confidence level required.

cv A single numeric for the critical value for calibration of empirical likelihood ratio statistic.

pnames A character vector of length two for the name of parameters.

Examples

```
showClass("ConfregEL")
```

ControlEL-class *ControlEL class*

Description

S4 class for computational details of empirical likelihood.

Slots

maxit A single integer for the maximum number of iterations for the optimization with respect to θ.

maxit_l A single integer for the maximum number of iterations for the optimization with respect to λ.

tol A single numeric for the convergence tolerance denoted by ϵ. The iteration stops when

$$\|P\nabla l(\theta^{(k)})\| < \epsilon.$$

tol_l A single numeric for the relative convergence tolerance denoted by δ. The iteration stops when

$$\|\lambda^{(k)} - \lambda^{(k-1)}\| < \delta\|\lambda^{(k-1)}\| + \delta^2.$$

step A single numeric for the step size γ for the projected gradient descent method.

th A single numeric for the threshold for the negative empirical log-likelihood ratio.

verbose A single logical for whether to print a message on the convergence status.

keep_data A single logical for whether to keep the data used for fitting model objects.

nthreads A single integer for the number of threads for parallel computation via OpenMP (if available).

seed A single integer for the seed for random number generation.

an A single numeric representing the scaling factor for adjusted empirical likelihood calibration.

b A single integer for the number of bootstrap replicates.

m A single integer for the number of Monte Carlo samples.

Examples

```
showClass("ControlEL")
```

---

conv *Convergence check*

Description

Extracts the convergence status from a model.

Usage

```
## S4 method for signature 'CEL'
conv(object, ...)

## S4 method for signature 'EL'
conv(object, ...)

## S4 method for signature 'ELT'
conv(object, ...)

## S4 method for signature 'SummaryEL'
conv(object, ...)

## S4 method for signature 'SummaryELT'
conv(object, ...)

## S4 method for signature 'SummaryLM'
conv(object, ...)
```

Arguments

object	An object that contains the convergence status.
...	Further arguments passed to methods.

Value

A single logical.

Methods (by class)

- conv(CEL): Extracts the convergence status of the model with respect to the parameter.
- conv(EL): Extracts the convergence status of the model with respect to the Lagrange multiplier.
- conv(ELT): Extracts the convergence status of the test with respect to the parameter (or the Lagrange multiplier if the argument lhs is NULL).
- conv(SummaryEL): Extracts the convergence status of the model with respect to the Lagrange multiplier.
- conv(SummaryELT): Extracts the convergence status of the test with respect to the parameter (or the Lagrange multiplier if the argument lhs is NULL).
- conv(SummaryLM): Extracts the convergence status of the model. See the documentation of EL and CEL.

See Also

CEL, EL, ELT, getOptim()

Examples

```
## Convergence check for the overall model test
data("mtcars")
fit <- el_lm(mpg ~ ., data = mtcars)
conv(fit)
```

---

critVal *Critical value*

Description

Extracts the critical value from a model.

Usage

```
## S4 method for signature 'ELMT'
critVal(object, ...)

## S4 method for signature 'ELT'
critVal(object, ...)

## S4 method for signature 'SummaryELMT'
critVal(object, ...)

## S4 method for signature 'SummaryELT'
critVal(object, ...)
```

Arguments

`object`	An object that contains the critical value.
`...`	Further arguments passed to methods.

Value

A single numeric.

See Also

ELMT, ELT

Examples

```
## F-calibrated critical value
data("precip")
fit <- el_mean(precip, 30)
elt <- elt(fit, rhs = 34, calibrate = "f")
critVal(elt)
```

`EL-class` *EL class*

Description

S4 class for empirical likelihood.

Details

Let X_i be independent and identically distributed p-dimensional random variable from an unknown distribution P for $i = 1, \ldots, n$. We assume that P has a positive definite covariance matrix. For a parameter of interest $\theta(F) \in \mathbb{R}^p$, consider a p-dimensional smooth estimating function $g(X_i, \theta)$ with a moment condition

$$\mathrm{E}[g(X_i, \theta)] = 0.$$

We assume that there exists an unique θ_0 that solves the above equation. Given a value of θ, the (profile) empirical likelihood ratio is defined by

$$R(\theta) = \max_{p_i} \left\{ \prod_{i=1}^{n} n p_i : \sum_{i=1}^{n} p_i g(X_i, \theta) = 0, p_i \geq 0, \sum_{i=1}^{n} p_i = 1 \right\}.$$

The Lagrange multiplier $\lambda \equiv \lambda(\theta)$ of the dual problem leads to

$$p_i = \frac{1}{n} \frac{1}{1 + \lambda^\top g(X_i, \theta)},$$

where λ solves

$$\frac{1}{n} \sum_{i=1}^{n} \frac{g(X_i, \theta)}{1 + \lambda^\top g(X_i, \theta)} = 0.$$

Then the empirical log-likelihood ratio is given by

$$\log R(\theta) = -\sum_{i=1}^{n} \log(1 + \lambda^\top g(X_i, \theta)).$$

This problem can be efficiently solved by the Newton-Raphson method when the zero vector is contained in the interior of the convex hull of $\{g(X_i, \theta)\}_{i=1}^n$.

It is known that $-2 \log R(\theta_0)$ converges in distribution to χ_p^2, where χ_p^2 has a chi-square distribution with p degrees of freedom. See the references below for more details.

Slots

`optim` A list of the following optimization results:

- `par` A numeric vector of the specified parameters.
- `lambda` A numeric vector of the Lagrange multipliers of the dual problem corresponding to `par`.
- `iterations` A single integer for the number of iterations performed.
- `convergence` A single logical for the convergence status.
- `cstr` A single logical for whether constrained EL optimization is performed or not.

`logp` A numeric vector of the log probabilities of the empirical likelihood.

`logl` A single numeric of the empirical log-likelihood.

`loglr` A single numeric of the empirical log-likelihood ratio.

`statistic` A single numeric of minus twice the empirical log-likelihood ratio with an asymptotic chi-square distribution.

`df` A single integer for the degrees of freedom of the statistic.

`pval` A single numeric for the p-value of the statistic.

`nobs` A single integer for the number of observations.

`npar` A single integer for the number of parameters.

`weights` A numeric vector of the re-scaled weights used for the model fitting.

`coefficients` A numeric vector of the maximum empirical likelihood estimates of the parameters.

`method` A single character for the method dispatch in internal functions.

`data` A numeric matrix of the data for the model fitting.

`control` An object of class ControlEL constructed by `el_control()`.

References

Owen A (2001). *Empirical Likelihood*. Chapman & Hall/CRC. doi:10.1201/9781420036152.

Qin J, Lawless J (1994). "Empirical Likelihood and General Estimating Equations." *The Annals of Statistics*, **22**(1), 300–325. doi:10.1214/aos/1176325370.

Examples

```
showClass("EL")
```

el_control *Control parameters for computation*

Description

Specifies computational details of (constrained) empirical likelihood.

Usage

```
el_control(
  maxit = 200L,
  maxit_l = 25L,
  tol = 1e-06,
  tol_l = 1e-06,
  step = NULL,
  th = NULL,
  verbose = FALSE,
  keep_data = TRUE,
  nthreads,
  seed = NULL,
  an = NULL,
  b = 10000L,
  m = 1000000L
)
```

Arguments

maxit	A single integer for the maximum number of iterations for constrained minimization of empirical likelihood. Defaults to 200.
maxit_l	A single integer for the maximum number of iterations for evaluation of empirical likelihood. Defaults to 25.
tol	A single numeric for the convergence tolerance for the constrained minimization. Defaults to 1e-06.
tol_l	A single numeric for the relative convergence tolerance for the evaluation. Defaults to 1e-06.
step	A single numeric for the step size for projected gradient descent method. Defaults to NULL and sets the step size to the reciprocal of the sample size.
th	A single numeric for the threshold for the negative empirical log-likelihood ratio. The iteration stops if the value exceeds the threshold. Defaults to NULL and sets the threshold to 200 * d, where d corresponds to the degrees of freedom of the limiting chi-squared distribution of the statistic.
verbose	A single logical. If TRUE, a message on the convergence status is printed when fitting objects that inherit from class EL. Defaults to FALSE.
keep_data	A single logical. If TRUE, the data used for fitting objects that inherit from class EL are stored for later use with other methods. Defaults to TRUE.

nthreads	A single integer for the number of threads for parallel computation via OpenMP (if available). Defaults to half the available threads. For better performance, it is generally recommended in most platforms to limit the number of threads to the number of physical cores. Note that it applies to the following functions that involve multiple evaluations or optimizations: `confint()`, `confreg()`, `el_lm()`, `el_glm()`, `eld()`, and `elt()`.
seed	A single integer for the seed for random number generation. It only applies to `elt()` when `calibrate` is set to `"boot"`. Defaults to `NULL`. In this case, a seed is set to a random integer generated from 1 to the maximum integer supported by R on the machine, which is determined by `set.seed()`. Only one seed is needed even when multiple threads are used with `nthreads`. Each thread is given a separate seed to produce a non-overlapping but reproducible sequence of random numbers. The Xoshiro256+ pseudo-random number generator is used internally to work with OpenMP.
an	A single numeric representing the scaling factor for adjusted empirical likelihood calibration. It only applies to `elt()` when `calibrate` is set to `"ael"`. Defaults to `NULL`.
b	A single integer for the number of bootstrap replicates. It only applies to `elt()` when `calibrate` is set to `"boot"`. Defaults to `10000`.
m	A single integer for the number of Monte Carlo samples. It only applies to `elmt()`. Defaults to `1e+06`.

Value

An object of class of ControlEL.

See Also

`el_eval()`, `elt()`

Examples

```
optcfg <- el_control(maxit = 300, step = 0.01, th = 200, nthreads = 1)
```

---

el_eval *Empirical likelihood for general estimating functions*

Description

Computes empirical likelihood with general estimating functions.

Usage

```
el_eval(g, weights = NULL, control = el_control())
```

Arguments

g	A numeric matrix, or an object that can be coerced to a numeric matrix. Each row corresponds to an observation of an estimating function. The number of rows must be greater than the number of columns.
weights	An optional numeric vector of weights to be used in the fitting process. The length of the vector must be the same as the number of rows in g. Defaults to NULL, corresponding to identical weights. If non-NULL, weighted empirical likelihood is computed.
control	An object of class ControlEL constructed by el_control().

Details

Let X_i be independent and identically distributed p-dimensional random variable from an unknown distribution P for $i = 1, \ldots, n$. We assume that P has a positive definite covariance matrix. For a parameter of interest $\theta(F) \in \mathbb{R}^p$, consider a p-dimensional smooth estimating function $g(X_i, \theta)$ with a moment condition

$$\mathrm{E}[g(X_i, \theta)] = 0.$$

We assume that there exists an unique θ_0 that solves the above equation. Given a value of θ, the (profile) empirical likelihood ratio is defined by

$$R(\theta) = \max_{p_i} \left\{ \prod_{i=1}^{n} np_i : \sum_{i=1}^{n} p_i g(X_i, \theta) = 0, p_i \geq 0, \sum_{i=1}^{n} p_i = 1 \right\}.$$

el_mean() computes the empirical log-likelihood ratio statistic $-2 \log R(\theta)$ with the n by p numeric matrix g, whose ith row is $g(X_i, \theta)$. Since the estimating function can be arbitrary, el_eval() does not return an object of class EL, and the associated generics and methods are not applicable.

Value

A list of the following optimization results:

- optim A list with the following optimization results:
  - lambda A numeric vector of the Lagrange multipliers of the dual problem.
  - iterations A single integer for the number of iterations performed.
  - convergence A single logical for the convergence status.
- logp A numeric vector of the log probabilities of the empirical likelihood.
- logl A single numeric of the empirical log-likelihood.
- loglr A single numeric of the empirical log-likelihood ratio.
- statistic A single numeric of minus twice the empirical log-likelihood ratio with an asymptotic chi-square distribution.
- df A single integer for the degrees of freedom of the statistic.
- pval A single numeric for the p-value of the statistic.
- nobs A single integer for the number of observations.
- npar A single integer for the number of parameters.
- weights A numeric vector of the re-scaled weights used for the model fitting.

References

Qin J, Lawless J (1994). "Empirical Likelihood and General Estimating Equations." *The Annals of Statistics*, **22**(1), 300–325. doi:10.1214/aos/1176325370.

See Also

EL, `el_control()`

Examples

```
set.seed(123526)
mu <- 0
sigma <- 1
x <- rnorm(100)
g <- matrix(c(x - mu, (x - mu)^2 - sigma^2), ncol = 2)
el_eval(g, weights = rep(c(1, 2), each = 50))
```

---

el_glm *Empirical likelihood for generalized linear models*

Description

Fits a generalized linear model with empirical likelihood.

Usage

```
el_glm(
  formula,
  family = gaussian,
  data,
  weights = NULL,
  na.action,
  start = NULL,
  etastart = NULL,
  mustart = NULL,
  offset,
  control = el_control(),
  ...
)
```

Arguments

`formula`	An object of class `formula` (or one that can be coerced to that class): a symbolic description of the model to be fitted.
`family`	A description of the error distribution and link function to be used in the model. Only the result of a call to a family function is supported. See 'Details'.

`data`	An optional data frame, list or environment (or object coercible by `as.data.frame()` to a data frame) containing the variables in the formula. If not found in data, the variables are taken from `environment(formula)`.
`weights`	An optional numeric vector of weights to be used in the fitting process. Defaults to `NULL`, corresponding to identical weights. If non-`NULL`, weighted empirical likelihood is computed.
`na.action`	A function which indicates what should happen when the data contain `NA`s. The default is set by the `na.action` setting of `options`, and is `na.fail` if that is unset.
`start`	Starting values for the parameters in the linear predictor. Defaults to `NULL` and is passed to `glm.fit()`.
`etastart`	Starting values for the linear predictor. Defaults to `NULL` and is passed to `glm.fit()`.
`mustart`	Starting values for the vector of means. Defaults to `NULL` and is passed to `glm.fit()`.
`offset`	An optional expression for specifying an *a priori* known component to be included in the linear predictor during fitting. This should be `NULL` or a numeric vector or matrix of extents matching those of the response. One or more `offset` terms can be included in the formula instead or as well, and if more than one are specified their sum is used.
`control`	An object of class ControlEL constructed by `el_control()`.
`...`	Additional arguments to be passed to `glm.control()`.

Details

Suppose that we observe n independent random variables $Z_i \equiv (X_i, Y_i)$ from a common distribution, where X_i is the p-dimensional covariate (including the intercept if any) and Y_i is the response. A generalized linear model specifies that $\mathrm{E}(Y_i|X_i) = \mu_i$, $G(\mu_i) = X_i^\top \theta$, and $\mathrm{Var}(Y_i|X_i) = \phi V(\mu_i)$, where $\theta = (\theta_0, \ldots, \theta_{p-1})$ is an unknown p-dimensional parameter, ϕ is an optional dispersion parameter, G is a known smooth link function, and V is a known variance function.

With H denoting the inverse link function, define the quasi-score

$$g_1(Z_i, \theta) = \left\{ H'(X_i^\top \theta) \left(Y_i - H(X_i^\top \theta)\right) / \left(\phi V\left(H(X_i^\top \theta)\right)\right) \right\} X_i.$$

Then we have the estimating equations $\sum_{i=1}^n g_1(Z_i, \theta) = 0$. When ϕ is known, the (profile) empirical likelihood ratio for a given θ is defined by

$$R_1(\theta) = \max_{p_i} \left\{ \prod_{i=1}^n n p_i : \sum_{i=1}^n p_i g_1(Z_i, \theta) = 0,\ p_i \geq 0,\ \sum_{i=1}^n p_i = 1 \right\}.$$

With unknown ϕ, we introduce another estimating function based on the squared residuals. Let $\eta = (\theta, \phi)$ and

$$g_2(Z_i, \eta) = \left(Y_i - H(X_i^\top \theta)\right)^2 / \left(\phi^2 V\left(H(X_i^\top \theta)\right)\right) - 1/\phi.$$

Now the empirical likelihood ratio is defined by

$$R_2(\eta) = \max_{p_i} \left\{ \prod_{i=1}^n n p_i : \sum_{i=1}^n p_i g_1(Z_i, \eta) = 0,\ \sum_{i=1}^n p_i g_2(Z_i, \eta) = 0,\ p_i \geq 0,\ \sum_{i=1}^n p_i = 1 \right\}.$$

el_glm() first computes the parameter estimates by calling glm.fit() (with ... if any) with the model.frame and model.matrix obtained from the formula. Note that the maximum empirical likelihood estimator is the same as the the quasi-maximum likelihood estimator in our model. Next, it tests hypotheses based on asymptotic chi-square distributions of the empirical likelihood ratio statistics. Included in the tests are overall test with

$$H_0 : \theta_1 = \theta_2 = \cdots = \theta_{p-1} = 0,$$

and significance tests for each parameter with

$$H_{0j} : \theta_j = 0,\ j = 0, \ldots, p-1.$$

The available families and link functions are as follows:

- gaussian: "identity", "log", and "inverse".
- binomial: "logit", "probit", and "log".
- poisson: "log", "identity", and "sqrt".
- quasipoisson: "log", "identity", and "sqrt".

Value

An object of class of GLM.

References

Chen SX, Cui H (2003). "An Extended Empirical Likelihood for Generalized Linear Models." *Statistica Sinica*, **13**(1), 69–81.

Kolaczyk ED (1994). "Empirical Likelihood for Generalized Linear Models." *Statistica Sinica*, **4**(1), 199–218.

See Also

EL, GLM, el_lm(), elt(), el_control()

Examples

```
data("warpbreaks")
fit <- el_glm(wool ~ .,
  family = binomial, data = warpbreaks, weights = NULL, na.action = na.omit,
  start = NULL, etastart = NULL, mustart = NULL, offset = NULL
)
summary(fit)
```

el_lm *Empirical likelihood for linear models*

Description

Fits a linear model with empirical likelihood.

Usage

```
el_lm(
  formula,
  data,
  weights = NULL,
  na.action,
  offset,
  control = el_control(),
  ...
)
```

Arguments

formula	An object of class formula (or one that can be coerced to that class) for a symbolic description of the model to be fitted.
data	An optional data frame, list or environment (or object coercible by as.data.frame() to a data frame) containing the variables in formula. If not found in data, the variables are taken from environment(formula).
weights	An optional numeric vector of weights to be used in the fitting process. Defaults to NULL, corresponding to identical weights. If non-NULL, weighted empirical likelihood is computed.
na.action	A function which indicates what should happen when the data contain NAs. The default is set by the na.action setting of options, and is na.fail if that is unset.
offset	An optional expression for specifying an *a priori* known component to be included in the linear predictor during fitting. This should be NULL or a numeric vector or matrix of extents matching those of the response. One or more offset terms can be included in the formula instead or as well, and if more than one are specified their sum is used.
control	An object of class ControlEL constructed by el_control().
...	Additional arguments to be passed to the low level regression fitting functions. See 'Details'.

Details

Suppose that we observe n independent random variables $Z_i \equiv (X_i, Y_i)$ from a common distribution, where X_i is the p-dimensional covariate (including the intercept if any) and Y_i is the response.

We consider the following linear model:

$$Y_i = X_i^\top \theta + \epsilon_i,$$

where $\theta = (\theta_0, \ldots, \theta_{p-1})$ is an unknown p-dimensional parameter and the errors ϵ_i are independent random variables that satisfy $\mathrm{E}(\epsilon_i | X_i) = 0$. We assume that the errors have finite conditional variances. Then the least square estimator of θ solves the following estimating equations:

$$\sum_{i=1}^{n} (Y_i - X_i^\top \theta) X_i = 0.$$

Given a value of θ, let $g(Z_i, \theta) = (Y_i - X_i^\top \theta) X_i$ and the (profile) empirical likelihood ratio is defined by

$$R(\theta) = \max_{p_i} \left\{ \prod_{i=1}^{n} n p_i : \sum_{i=1}^{n} p_i g(Z_i, \theta) = \theta,\ p_i \geq 0,\ \sum_{i=1}^{n} p_i = 1 \right\}.$$

`el_lm()` first computes the parameter estimates by calling `lm.fit()` (with `...` if any) with the `model.frame` and `model.matrix` obtained from the `formula`. Note that the maximum empirical likelihood estimator is the same as the the quasi-maximum likelihood estimator in our model. Next, it tests hypotheses based on asymptotic chi-square distributions of the empirical likelihood ratio statistics. Included in the tests are overall test with

$$H_0 : \theta_1 = \theta_2 = \cdots = \theta_{p-1} = 0,$$

and significance tests for each parameter with

$$H_{0j} : \theta_j = 0,\ j = 0, \ldots, p-1.$$

Value

An object of class of LM.

References

Owen A (1991). "Empirical Likelihood for Linear Models." *The Annals of Statistics*, **19**(4), 1725–1747. doi:10.1214/aos/1176348368.

See Also

EL, LM, `el_glm()`, `elt()`, `el_control()`

Examples

```
## Linear model
data("thiamethoxam")
fit <- el_lm(fruit ~ trt, data = thiamethoxam)
summary(fit)

## Weighted data
wfit <- el_lm(fruit ~ trt, data = thiamethoxam, weights = visit)
summary(wfit)
```

```
## Missing data
fit2 <- el_lm(fruit ~ trt + scb, data = thiamethoxam,
  na.action = na.omit, offset = NULL
)
summary(fit2)
```

---

el_mean *Empirical likelihood for the mean*

Description

Computes empirical likelihood for the mean.

Usage

```
el_mean(x, par, weights = NULL, control = el_control())
```

Arguments

x	A numeric matrix, or an object that can be coerced to a numeric matrix. Each row corresponds to an observation. The number of rows must be greater than the number of columns.
par	A numeric vector of parameter values to be tested. The length of the vector must be the same as the number of columns in x.
weights	An optional numeric vector of weights to be used in the fitting process. The length of the vector must be the same as the number of rows in x. Defaults to NULL, corresponding to identical weights. If non-NULL, weighted empirical likelihood is computed.
control	An object of class ControlEL constructed by el_control().

Details

Let X_i be independent and identically distributed p-dimensional random variable from an unknown distribution P for $i = 1, \ldots, n$. We assume that $\mathrm{E}[X_i] = \theta_0 \in \mathbb{R}^p$ and that P has a positive definite covariance matrix. Given a value of θ, the (profile) empirical likelihood ratio is defined by

$$R(\theta) = \max_{p_i} \left\{ \prod_{i=1}^{n} np_i : \sum_{i=1}^{n} p_i X_i = \theta,\ p_i \geq 0,\ \sum_{i=1}^{n} p_i = 1 \right\}.$$

el_mean() computes the empirical log-likelihood ratio statistic $-2 \log R(\theta)$, along with other values in EL.

Value

An object of class EL.

References

Owen A (1990). "Empirical Likelihood Ratio Confidence Regions." *The Annals of Statistics*, **18**(1), 90–120. doi:10.1214/aos/1176347494.

See Also

EL, `elt()`, `el_eval()`, `el_control()`

Examples

```
## Scalar mean
data("precip")
fit <- el_mean(precip, 30)
fit
summary(fit)

## Vector mean
data("faithful")
fit2 <- el_mean(faithful, par = c(3.5, 70))
summary(fit2)

## Weighted data
w <- rep(c(1, 2), each = nrow(faithful) / 2)
fit3 <- el_mean(faithful, par = c(3.5, 70), weights = w)
summary(fit3)
```

---

el_sd *Empirical likelihood for the standard deviation*

Description

Computes empirical likelihood for the standard deviation.

Usage

```
el_sd(x, mean, sd, weights = NULL, control = el_control())
```

Arguments

`x`	A numeric vector, or an object that can be coerced to a numeric vector.
`mean`	A single numeric for the (known) mean value.
`sd`	A positive single numeric for the parameter value to be tested.
`weights`	An optional numeric vector of weights to be used in the fitting process. The length of the vector must be the same as the length of `x`. Defaults to `NULL`, corresponding to identical weights. If non-`NULL`, weighted empirical likelihood is computed.
`control`	An object of class ControlEL constructed by `el_control()`.

Details

Let X_i be independent and identically random variable from an unknown distribution P for $i = 1, \ldots, n$. We assume that $\mathrm{E}[X_i] = \mu_0$ is known and that P has a variance σ_0^2. Given a value of σ, the (profile) empirical likelihood ratio is defined by

$$R(\sigma) = \max_{p_i} \left\{ \prod_{i=1}^{n} np_i : \sum_{i=1}^{n} p_i (X_i - \mu_0)^2 = \sigma^2,\ p_i \geq 0,\ \sum_{i=1}^{n} p_i = 1 \right\}.$$

`el_sd()` computes the empirical log-likelihood ratio statistic $-2 \log R(\sigma)$, along with other values in SD.

Value

An object of class SD.

See Also

EL, SD, `el_mean()`, `elt()`, `el_control()`

Examples

```
data("women")
x <- women$height
w <- women$weight
fit <- el_sd(x, mean = 65, sd = 5, weights = w)
fit
summary(fit)
```

`eld` *Empirical likelihood displacement*

Description

Computes empirical likelihood displacement for model diagnostics and outlier detection.

Usage

```
## S4 method for signature 'EL'
eld(object, control = NULL)

## S4 method for signature 'GLM'
eld(object, control = NULL)
```

Arguments

`object`	An object that inherits from EL.
`control`	An object of class ControlEL constructed by `el_control()`. Defaults to `NULL` and inherits the `control` slot in `object`.

Details

Let $L(\theta)$ be the empirical log-likelihood function based on the full sample with n observations. The maximum empirical likelihood estimate is denoted by $\hat{\theta}$. Consider a reduced sample with the ith observation deleted and the corresponding estimate $\hat{\theta}_{(i)}$. The empirical likelihood displacement is defined by

$$\text{ELD}_i = 2\{L(\hat{\theta}) - L(\hat{\theta}_{(i)})\}.$$

If ELD_i is large, then the ith observation is an influential point and can be inspected as a possible outlier. `eld` computes ELD_i for $i = 1, \ldots, n$.

Value

An object of class ELD.

References

Lazar NA (2005). "Assessing the Effect of Individual Data Points on Inference From Empirical Likelihood." *Journal of Computational and Graphical Statistics*, **14**(3), 626–642. doi:10.1198/106186005X59568.

Zhu H, Ibrahim JG, Tang N, Zhang H (2008). "Diagnostic Measures for Empirical Likelihood of General Estimating Equations." *Biometrika*, **95**(2), 489–507. doi:10.1093/biomet/asm094.

See Also

EL, ELD, `el_control()`, `plot()`

Examples

```
data("precip")
fit <- el_mean(precip, par = 30)
eld <- eld(fit)
plot(eld)
```

`ELD-class` *ELD class*

Description

S4 class for empirical likelihood displacement. It inherits from `"numeric"`.

Examples

```
showClass("ELD")
```

`elmt` *Empirical likelihood multiple tests*

Description

Tests multiple linear hypotheses simultaneously.

Usage

```
## S4 method for signature 'EL'
elmt(object, rhs = NULL, lhs = NULL, alpha = 0.05, control = NULL)
```

Arguments

`object`	An object that inherits from EL.
`rhs`	A numeric vector (column matrix) or a list of numeric vectors for the right-hand sides of hypotheses. Defaults to `NULL`. See 'Details'.
`lhs`	A list or a numeric matrix for the left-hand sides of hypotheses. For a list `lhs`, each element must be specified as a single instance of the `lhs` in `elt()`. For a matrix `lhs`, each row gives a linear combination of the parameters in `object`. The number of columns must be equal to the number of parameters. Defaults to `NULL`. See 'Details'.
`alpha`	A single numeric for the overall significance level. Defaults to `0.05`.
`control`	An object of class ControlEL constructed by `el_control()`. Defaults to `NULL` and inherits the `control` slot in `object`.

Details

`elmt()` tests multiple hypotheses simultaneously. Each hypothesis corresponds to the constrained empirical likelihood ratio described in CEL. `rhs` and `lhs` cannot be both `NULL`. The right-hand side and left-hand side of each hypothesis must be specified as described in `elt()`.

For specifying linear contrasts more conveniently, `rhs` and `lhs` also take a numeric vector and a numeric matrix, respectively. Each element of `rhs` and each row of `lhs` correspond to a contrast (hypothesis).

The vector of empirical likelihood ratio statistics asymptotically follows a multivariate chi-square distribution under the complete null hypothesis. The multiple testing procedure asymptotically controls the family-wise error rate at the level `alpha`. Based on the distribution of the maximum of the test statistics, the adjusted p-values are estimated by Monte Carlo simulation.

Value

An object of class of ELMT.

References

Kim E, MacEachern SN, Peruggia M (2023). "Empirical likelihood for the analysis of experimental designs." *Journal of Nonparametric Statistics*, **35**(4), 709–732. doi:10.1080/10485252.2023.2206919.

Kim E, MacEachern SN, Peruggia M (2024). "melt: Multiple Empirical Likelihood Tests in R." *Journal of Statistical Software*, **108**(5), 1–33. doi:10.18637/jss.v108.i05.

See Also

EL, ELMT, `elt()`, `el_control()`

Examples

```
## Bivariate mean (list `rhs` & no `lhs`)
set.seed(143)
data("women")
fit <- el_mean(women, par = c(65, 135))
rhs <- list(c(64, 133), c(66, 140))
elmt(fit, rhs = rhs)

## Pairwise comparison (no `rhs` & list `lhs`)
data("clothianidin")
fit2 <- el_lm(clo ~ -1 + trt, clothianidin)
lhs2 <- list(
  "trtNaked - trtFungicide",
  "trtFungicide - trtLow",
  "trtLow - trtHigh"
)
elmt(fit2, lhs = lhs2)

## Arbitrary hypotheses (list `rhs` & list `lhs`)
data("mtcars")
fit3 <- el_lm(mpg ~ wt + qsec, data = mtcars)
lhs3 <- list(c(1, 4, 0), rbind(c(0, 1, 0), c(0, 0, 1)))
rhs3 <- list(0, c(-6, 1))
elmt(fit3, rhs = rhs3, lhs = lhs3)
```

ELMT-class *ELMT class*

Description

S4 class for empirical likelihood multiple tests.

Slots

`estimates` A list of numeric vectors of the estimates of the linear hypotheses.

`statistic` A numeric vector of minus twice the (constrained) empirical log-likelihood ratios with asymptotic chi-square distributions.

`df` An integer vector of the marginal degrees of freedom of the statistic.

`pval` A numeric vector for the multiplicity adjusted p-values.

`cv` A single numeric for the multiplicity adjusted critical value.

`rhs` A numeric vector for the right-hand sides of the hypotheses.

`lhs` A numeric matrix for the left-hand side of the hypotheses.

`alpha` A single numeric for the overall significance level.

`calibrate` A single character for the calibration method used.

`weights` A numeric vector of the re-scaled weights used for the model fitting.

`coefficients` A numeric vector of the maximum empirical likelihood estimates of the parameters.

`method` A single character for the method dispatch in internal functions.

`data` A numeric matrix of the data for the model fitting.

`control` An object of class ControlEL constructed by `el_control()`.

Examples

```
showClass("ELMT")
```

`elt` *Empirical likelihood test*

Description

Tests a linear hypothesis with various calibration options.

Usage

```
## S4 method for signature 'EL'
elt(
  object,
  rhs = NULL,
  lhs = NULL,
  alpha = 0.05,
  calibrate = "chisq",
  control = NULL
)
```

Arguments

`object`	An object that inherits from EL.
`rhs`	A numeric vector or a column matrix for the right-hand side of hypothesis, with as many entries as the rows in `lhs`. Defaults to `NULL`. See 'Details'.

lhs	A numeric matrix or a vector (treated as a row matrix) for the left-hand side of a hypothesis. Each row gives a linear combination of the parameters in object. The number of columns must be equal to the number of parameters. Or a character vector with a symbolic description of the hypothesis is allowed. Defaults to NULL. See 'Details'.
alpha	A single numeric for the significance level. Defaults to 0.05.
calibrate	A single character representing the calibration method. It is case-insensitive and must be one of "ael", "boot", "chisq", or "f". Defaults to "chisq". See 'Details'.
control	An object of class ControlEL constructed by el_control(). Defaults to NULL and inherits the control slot in object.

Details

elt() performs the constrained minimization of $l(\theta)$ described in CEL. rhs and lhs cannot be both NULL. For non-NULL lhs, it is required that lhs have full row rank $q \leq p$ and p be equal to the number of parameters in the object.

Depending on the specification of rhs and lhs, we have the following three cases:

1. If both rhs and lhs are non-NULL, the constrained minimization is performed with the right-hand side r and the left-hand side L as
$$\inf_{\theta: L\theta = r} l(\theta).$$
2. If rhs is NULL, r is set to the zero vector as $\inf_{\theta: L\theta=0} l(\theta)$.
3. If lhs is NULL, L is set to the identity matrix and the problem reduces to evaluating at r as $l(r)$.

calibrate specifies the calibration method used. Four methods are available: "ael" (adjusted empirical likelihood calibration), "boot" (bootstrap calibration), "chisq" (chi-square calibration), and "f" (F calibration). When lhs is not NULL, only "chisq" is available. "f" is applicable only to the mean parameter. The an slot in control applies specifically to "ael", while the nthreads, seed, and B slots apply to "boot".

Value

An object of class of ELT. If lhs is non-NULL, the optim slot corresponds to that of CEL. Otherwise, it corresponds to that of EL.

References

Adimari G, Guolo A (2010). "A Note on the Asymptotic Behaviour of Empirical Likelihood Statistics." *Statistical Methods & Applications*, **19**(4), 463–476. doi:10.1007/s1026001001379.

Chen J, Variyath AM, Abraham B (2008). "Adjusted Empirical Likelihood and Its Properties." *Journal of Computational and Graphical Statistics*, **17**(2), 426–443.

Kim E, MacEachern SN, Peruggia M (2024). "melt: Multiple Empirical Likelihood Tests in R." *Journal of Statistical Software*, **108**(5), 1–33. doi:10.18637/jss.v108.i05.

Qin J, Lawless J (1995). "Estimating Equations, Empirical Likelihood and Constraints on Parameters." *Canadian Journal of Statistics*, **23**(2), 145–159. doi:10.2307/3315441.

See Also

EL, ELT, `elmt()`, `el_control()`

Examples

```
## Adjusted empirical likelihood calibration
data("precip")
fit <- el_mean(precip, 32)
elt(fit, rhs = 100, calibrate = "ael")

## Bootstrap calibration
elt(fit, rhs = 32, calibrate = "boot")

## F calibration
elt(fit, rhs = 32, calibrate = "f")

## Test of no treatment effect
data("clothianidin")
contrast <- matrix(c(
  1, -1, 0, 0,
  0, 1, -1, 0,
  0, 0, 1, -1
), byrow = TRUE, nrow = 3)
fit2 <- el_lm(clo ~ -1 + trt, clothianidin)
elt(fit2, lhs = contrast)

## A symbolic description of the same hypothesis
elt(fit2, lhs = c(
  "trtNaked - trtFungicide",
  "trtFungicide - trtLow",
  "trtLow - trtHigh"
))
```

`ELT-class` *ELT class*

Description

S4 class for empirical likelihood test.

Slots

`optim` A list of the following optimization results:

- `par` A numeric vector of the solution to the (constrained) optimization problem.
- `lambda` A numeric vector of the Lagrange multipliers of the dual problem corresponding to `par`.
- `iterations` A single integer for the number of iterations performed.
- `convergence` A single logical for the convergence status.

- `cstr` A single logical for whether constrained EL optimization is performed or not.

`logp` A numeric vector of the log probabilities of the (constrained) empirical likelihood.

`logl` A single numeric of the (constrained) empirical log-likelihood.

`loglr` A single numeric of the (constrained) empirical log-likelihood ratio.

`statistic` A single numeric of minus twice the (constrained) empirical log-likelihood ratio with an asymptotic chi-square distribution.

`df` A single integer for the chi-square degrees of freedom of the statistic.

`pval` A single numeric for the (calibrated) p-value of the statistic.

`cv` A single numeric for the critical value.

`rhs` A numeric vector for the right-hand side of the hypothesis.

`lhs` A numeric matrix for the left-hand side of the hypothesis.

`alpha` A single numeric for the significance level.

`calibrate` A single character for the calibration method used.

`control` An object of class ControlEL constructed by `el_control()`.

Examples

```
showClass("ELT")
```

`getDF` *Degrees of freedom*

Description

Extracts the degrees of freedom from a model.

Usage

```
## S4 method for signature 'EL'
getDF(object)

## S4 method for signature 'ELMT'
getDF(object)

## S4 method for signature 'ELT'
getDF(object)

## S4 method for signature 'SummaryEL'
getDF(object)

## S4 method for signature 'SummaryELMT'
getDF(object)

## S4 method for signature 'SummaryLM'
getDF(object)
```

Arguments

`object`	An object that contains the degrees of freedom.

Value

An integer vector.

Methods (by class)

- `getDF(EL)`: Extracts the degrees of freedom.
- `getDF(ELMT)`: Extracts the vector of marginal degrees of freedom.
- `getDF(ELT)`: Extracts the (chi-square) degrees of freedom.
- `getDF(SummaryEL)`: Extracts the degrees of freedom.
- `getDF(SummaryELMT)`: Extracts the vector of marginal degrees of freedom.
- `getDF(SummaryLM)`: Extracts the degrees of freedom.

See Also

EL, ELMT, ELT

Examples

```
data("faithful")
fit <- el_mean(faithful, par = c(3.5, 70))
getDF(fit)
```

`getOptim` *Optimization results*

Description

Extracts the optimization results from a model.

Usage

```
## S4 method for signature 'EL'
getOptim(object, ...)

## S4 method for signature 'ELT'
getOptim(object, ...)

## S4 method for signature 'SummaryEL'
getOptim(object, ...)

## S4 method for signature 'SummaryELT'
getOptim(object, ...)
```

```
## S4 method for signature 'SummaryLM'
getOptim(object, ...)
```

Arguments

`object`	An object that contains the optimization results.
`...`	Further arguments passed to methods.

Value

A list with the following optimization results:

- `par` A numeric vector of the parameter value. See the documentation of EL and CEL.
- `lambda` A numeric vector of the Lagrange multipliers.
- `iterations` A single integer for the number of iterations performed.
- `convergence` A single logical for the convergence status.

See Also

EL, ELT, `sigTests()`

Examples

```
data("precip")
fit <- el_mean(precip, par = 40)
getOptim(fit)
```

`GLM-class` *GLM class*

Description

S4 class for generalized linear models. It inherits from LM class.

Details

The overall test involves a constrained optimization problem. All the parameters except for the intercept are constrained to zero. The `optim` slot contains the results. When there is no intercept, all parameters are set to zero, and the results need to be understood in terms of EL class since no constrained optimization is involved. Once the solution is found, the log probabilities (`logp`) and the (constrained) empirical likelihood values (`logl`, `loglr`, `statistic`) readily follow, along with the degrees of freedom (`df`) and the p-value (`pval`). The significance tests for each parameter also involve constrained optimization problems where only one parameter is constrained to zero. The `sigTests` slot contains the results.

Slots

`family` A `family` object used.

`dispersion` A single numeric for the estimated dispersion parameter.

`sigTests` A list of the following results of significance tests:

- `statistic` A numeric vector of minus twice the (constrained) empirical log-likelihood ratios with asymptotic chi-square distributions.
- `iterations` An integer vector for the number of iterations performed for each parameter.
- `convergence` A logical vector for the convergence status of each parameter.
- `cstr` A single logical for whether constrained EL optimization is performed or not.

`call` A matched call.

`terms` A `terms` object used.

`misc` A list of various outputs obtained from the model fitting process. They are used in other generics and methods.

`optim` A list of the following optimization results:

- `par` A numeric vector of the solution to the (constrained) optimization problem.
- `lambda` A numeric vector of the Lagrange multipliers of the dual problem corresponding to `par`.
- `iterations` A single integer for the number of iterations performed.
- `convergence` A single logical for the convergence status.

`logp` A numeric vector of the log probabilities of the (constrained) empirical likelihood.

`logl` A single numeric of the (constrained) empirical log-likelihood.

`loglr` A single numeric of the (constrained) empirical log-likelihood ratio.

`statistic` A single numeric of minus twice the (constrained) empirical log-likelihood ratio with an asymptotic chi-square distribution.

`df` A single integer for the degrees of freedom of the statistic.

`pval` A single numeric for the p-value of the statistic.

`nobs` A single integer for the number of observations.

`npar` A single integer for the number of parameters.

`weights` A numeric vector of the re-scaled weights used for the model fitting.

`coefficients` A numeric vector of the maximum empirical likelihood estimates of the parameters.

`method` A single character for the method dispatch in internal functions.

`data` A numeric matrix of the data for the model fitting.

`control` An object of class ControlEL constructed by `el_control()`.

Examples

```
showClass("GLM")
```

LM-class *LM class*

Description

S4 class for linear models with empirical likelihood. It inherits from CEL class.

Details

The overall test involves a constrained optimization problem. All the parameters except for the intercept are constrained to zero. The `optim` slot contains the results. When there is no intercept, all parameters are set to zero, and the results need to be understood in terms of EL class since no constrained optimization is involved. Once the solution is found, the log probabilities (`logp`) and the (constrained) empirical likelihood values (`logl`, `loglr`, `statistic`) readily follow, along with the degrees of freedom (`df`) and the p-value (`pval`). The significance tests for each parameter also involve constrained optimization problems where only one parameter is constrained to zero. The `sigTests` slot contains the results.

Methods (by generic)

- `formula(LM)`: Extracts the symbolic model formula used in `el_lm()` or `el_glm()`.

Slots

`sigTests` A list of the following results of significance tests:

- `statistic` A numeric vector of minus twice the (constrained) empirical log-likelihood ratios with asymptotic chi-square distributions.
- `iterations` An integer vector for the number of iterations performed for each parameter.
- `convergence` A logical vector for the convergence status of each parameter.

`call` A matched call.

`terms` A `terms` object used.

`misc` A list of various outputs obtained from the model fitting process. They are used in other generics and methods.

`optim` A list of the following optimization results:

- `par` A numeric vector of the solution to the (constrained) optimization problem.
- `lambda` A numeric vector of the Lagrange multipliers of the dual problem corresponding to `par`.
- `iterations` A single integer for the number of iterations performed.
- `convergence` A single logical for the convergence status.

`logp` A numeric vector of the log probabilities of the (constrained) empirical likelihood.

`logl` A single numeric of the (constrained) empirical log-likelihood.

`loglr` A single numeric of the (constrained) empirical log-likelihood ratio.

`statistic` A single numeric of minus twice the (constrained) empirical log-likelihood ratio with an asymptotic chi-square distribution.

`df` A single integer for the degrees of freedom of the statistic.

`pval` A single numeric for the p-value of the statistic.

`nobs` A single integer for the number of observations.

`npar` A single integer for the number of parameters.

`weights` A numeric vector of the re-scaled weights used for the model fitting.

`coefficients` A numeric vector of the maximum empirical likelihood estimates of the parameters.

`method` A single character for the method dispatch in internal functions.

`data` A numeric matrix of the data for the model fitting.

`control` An object of class ControlEL constructed by `el_control()`.

Examples

```
showClass("LM")
```

`logL` *Empirical log-likelihood*

Description

Extracts the empirical log-likelihood from a model.

Usage

```
## S4 method for signature 'EL'
logL(object, ...)

## S4 method for signature 'ELT'
logL(object, ...)

## S4 method for signature 'SummaryEL'
logL(object, ...)

## S4 method for signature 'SummaryELT'
logL(object, ...)

## S4 method for signature 'SummaryLM'
logL(object, ...)
```

Arguments

`object`	An object that contains the empirical log-likelihood.
`...`	Further arguments passed to methods.

Value

A single numeric.

References

Baggerly KA (1998). "Empirical Likelihood as a Goodness-of-Fit Measure." *Biometrika*, **85**(3), 535–547. doi:10.1093/biomet/85.3.535.

See Also

EL, ELT

Examples

```
data("precip")
fit <- el_mean(precip, par = 40)
logL(fit)
```

---

logLR *Empirical log-likelihood ratio*

Description

Extracts the empirical log-likelihood ratio from a model.

Usage

```
## S4 method for signature 'EL'
logLR(object, ...)

## S4 method for signature 'ELT'
logLR(object, ...)

## S4 method for signature 'SummaryEL'
logLR(object, ...)

## S4 method for signature 'SummaryELT'
logLR(object, ...)

## S4 method for signature 'SummaryLM'
logLR(object, ...)
```

Arguments

`object`	An object that contains the empirical log-likelihood ratio.
`...`	Further arguments passed to methods.

Value

A single numeric.

References

Baggerly KA (1998). "Empirical Likelihood as a Goodness-of-Fit Measure." *Biometrika*, **85**(3), 535–547. doi:10.1093/biomet/85.3.535.

See Also

EL, ELT

Examples

```
data("precip")
fit <- el_mean(precip, par = 40)
logLR(fit)
```

`logProb` *Log probabilities*

Description

Extracts log probabilities of empirical likelihood from a model.

Usage

```
## S4 method for signature 'EL'
logProb(object, ...)

## S4 method for signature 'ELT'
logProb(object, ...)
```

Arguments

`object`	An object that inherits from EL or ELT.
`...`	Further arguments passed to methods.

Value

A numeric vector.

See Also

EL, ELT

Examples

```
data("precip")
fit <- el_mean(precip, par = 40)
logProb(fit)
```

nobs *Number of observations in a model*

Description

Extracts the number of observations from a model.

Usage

```
## S4 method for signature 'EL'
nobs(object, ...)

## S4 method for signature 'SummaryEL'
nobs(object, ...)

## S4 method for signature 'SummaryLM'
nobs(object, ...)
```

Arguments

object	An object that contains the number of observations.
...	Further arguments passed to methods.

Value

A single integer.

See Also

EL

Examples

```
data("precip")
fit <- el_mean(precip, par = 40)
nobs(fit)
```

plot *Plot methods*

Description

Provides plot methods for objects.

Usage

```
## S4 method for signature 'ConfregEL'
plot(x, y, ...)

## S4 method for signature 'EL'
plot(x, y, ...)

## S4 method for signature 'ELD'
plot(x, y, ...)
```

Arguments

x	An object to be plotted.
y	Not used.
...	Further graphical parameters (see par).

Value

No return value, called for side effects.

Methods (by class)

- plot(ConfregEL): Plots a two-dimensional confidence region for model parameters.
- plot(EL): Plots empirical likelihood displacement values versus observation index. eld() is called implicitly.
- plot(ELD): Plots empirical likelihood displacement values versus observation index.

See Also

ConfregEL, EL, ELD, confreg(), eld()

Examples

```
## Model
data("mtcars")
fit <- el_lm(hp ~ wt, data = mtcars)

## Confidence region
out1 <- confreg(fit, npoints = 500)
```

```
plot(out1)

## Empirical likelihood displacement
out2 <- eld(fit)
plot(out2)

## A shortcut to `ELD`
plot(fit)
```

---

print *Print methods*

Description

Provides print methods for objects.

Usage

```
## S4 method for signature 'EL'
print(x, digits = max(3L, getOption("digits") - 3L), ...)

## S4 method for signature 'ELMT'
print(x, digits = max(3L, getOption("digits") - 3L), ...)

## S4 method for signature 'ELT'
print(x, digits = max(3L, getOption("digits") - 3L), ...)

## S4 method for signature 'LM'
print(x, digits = max(3L, getOption("digits") - 3L), ...)

## S4 method for signature 'SummaryEL'
print(x, digits = max(3L, getOption("digits") - 3L), ...)

## S4 method for signature 'SummaryELMT'
print(
  x,
  digits = max(3L, getOption("digits") - 3L),
  signif.stars = getOption("show.signif.stars"),
  ...
)

## S4 method for signature 'SummaryELT'
print(x, digits = max(3L, getOption("digits") - 3L), ...)

## S4 method for signature 'SummaryGLM'
print(
  x,
  digits = max(3L, getOption("digits") - 3L),
```

```
  signif.stars = getOption("show.signif.stars"),
  ...
)

## S4 method for signature 'SummaryLM'
print(
  x,
  digits = max(3L, getOption("digits") - 3L),
  signif.stars = getOption("show.signif.stars"),
  ...
)
```

Arguments

x	An object to be printed.
...	Further arguments passed to methods.
digits	A single integer for the number of significant digits to be passed to format().
signif.stars	A single logical. If TRUE, 'significance stars' are printed for each parameter.

Value

The argument x (invisibly).

See Also

EL, ELMT, ELT, LM

Examples

```
data("precip")
fit <- el_mean(precip, par = 40)
print(fit)
```

---

pVal *p-value*

Description

Extracts the p-value from a model.

Usage

```
## S4 method for signature 'EL'
pVal(object, ...)

## S4 method for signature 'ELMT'
pVal(object, ...)
```

```
## S4 method for signature 'ELT'
pVal(object, ...)

## S4 method for signature 'SummaryEL'
pVal(object, ...)

## S4 method for signature 'SummaryELT'
pVal(object, ...)

## S4 method for signature 'SummaryELMT'
pVal(object, ...)

## S4 method for signature 'SummaryLM'
pVal(object, ...)
```

Arguments

`object`	An object that contains the p-value.
`...`	Further arguments passed to methods.

Value

The form of the value returned by `pVal()` depends on the class of its argument.

Methods (by class)

- `pVal(EL)`: Extracts the p-value.
- `pVal(ELMT)`: Extracts the multiplicity adjusted p-values.
- `pVal(ELT)`: Extracts the p-value.
- `pVal(SummaryEL)`: Extracts the p-value.
- `pVal(SummaryELT)`: Extracts the p-value.
- `pVal(SummaryELMT)`: Extracts the multiplicity adjusted p-values.
- `pVal(SummaryLM)`: Extracts the p-value.

See Also

EL, ELMT, ELT, `chisq()`

Examples

```
data("precip")
fit <- el_mean(precip, par = 40)
pVal(fit)
```

QGLM-class *QGLM class*

Description

S4 class for generalized linear models with quasi-likelihood methods. It inherits from GLM class.

Details

The overall test involves a constrained optimization problem. All the parameters except for the intercept are constrained to zero. The `optim` slot contains the results. When there is no intercept, all parameters are set to zero, and the results need to be understood in terms of EL class since no constrained optimization is involved. Once the solution is found, the log probabilities (`logp`) and the (constrained) empirical likelihood values (`logl`, `loglr`, `statistic`) readily follow, along with the degrees of freedom (`df`) and the p-value (`pval`). The significance tests for each parameter also involve constrained optimization problems where only one parameter is constrained to zero. The `sigTests` slot contains the results.

Slots

`family` A `family` object used.

`dispersion` A single numeric for the estimated dispersion parameter.

`sigTests` A list of the following results of significance tests:

- `statistic` A numeric vector of minus twice the (constrained) empirical log-likelihood ratios with asymptotic chi-square distributions.
- `iterations` An integer vector for the number of iterations performed for each parameter.
- `convergence` A logical vector for the convergence status of each parameter.
- `cstr` A single logical for whether constrained EL optimization is performed or not.

`call` A matched call.

`terms` A `terms` object used.

`misc` A list of various outputs obtained from the model fitting process. They are used in other generics and methods.

`optim` A list of the following optimization results:

- `par` A numeric vector of the solution to the (constrained) optimization problem.
- `lambda` A numeric vector of the Lagrange multipliers of the dual problem corresponding to `par`.
- `iterations` A single integer for the number of iterations performed.
- `convergence` A single logical for the convergence status.

`logp` A numeric vector of the log probabilities of the (constrained) empirical likelihood.

`logl` A single numeric of the (constrained) empirical log-likelihood.

`loglr` A single numeric of the (constrained) empirical log-likelihood ratio.

`statistic` A single numeric of minus twice the (constrained) empirical log-likelihood ratio with an asymptotic chi-square distribution.

`df` A single integer for the degrees of freedom of the statistic.

`pval` A single numeric for the p-value of the statistic.

`nobs` A single integer for the number of observations.

`npar` A single integer for the number of parameters.

`weights` A numeric vector of the re-scaled weights used for the model fitting.

`coefficients` A numeric vector of the maximum empirical likelihood estimates of the parameters.

`method` A single character for the method dispatch in internal functions.

`data` A numeric matrix of the data for the model fitting.

`control` An object of class ControlEL constructed by `el_control()`.

Examples

```
showClass("QGLM")
```

`SD-class` *SD class*

Description

S4 class for standard deviation. It inherits from EL class.

Slots

`optim` A list of the following optimization results:

- `par` A numeric vector of the specified parameters.
- `lambda` A numeric vector of the Lagrange multipliers of the dual problem corresponding to `par`.
- `iterations` A single integer for the number of iterations performed.
- `convergence` A single logical for the convergence status.
- `cstr` A single logical for whether constrained EL optimization is performed or not.

`logp` A numeric vector of the log probabilities of the empirical likelihood.

`logl` A single numeric of the empirical log-likelihood.

`loglr` A single numeric of the empirical log-likelihood ratio.

`statistic` A single numeric of minus twice the empirical log-likelihood ratio with an asymptotic chi-square distribution.

`df` A single integer for the degrees of freedom of the statistic.

`pval` A single numeric for the p-value of the statistic.

`nobs` A single integer for the number of observations.

`npar` A single integer for the number of parameters.

`weights` A numeric vector of the re-scaled weights used for the model fitting.

`coefficients` A numeric vector of the maximum empirical likelihood estimates of the parameters.

`method` A single character for the method dispatch in internal functions.

`data` A numeric matrix of the data for the model fitting.

`control` An object of class ControlEL constructed by `el_control()`.

Examples

```
showClass("SD")
```

`sigTests` *Significance tests*

Description

Extracts the results of significance tests from a model.

Usage

```
## S4 method for signature 'LM'
sigTests(object, ...)
```

Arguments

`object`	An object that inherits from LM.
`...`	Further arguments passed to methods.

Value

The form of the value returned by `sigTests()` depends on the class of its argument.

Methods (by class)

- `sigTests(LM)`: Extracts a list with the optimization results of significance tests.

See Also

LM, `getOptim()`

Examples

```
data("mtcars")
fit <- el_lm(mpg ~ ., data = mtcars)
sigTests(fit)
```

summary *Summary methods*

Description

Provides summary methods for objects.

Usage

```
## S4 method for signature 'EL'
summary(object, ...)

## S4 method for signature 'ELMT'
summary(object, ...)

## S4 method for signature 'ELT'
summary(object, ...)

## S4 method for signature 'GLM'
summary(object, ...)

## S4 method for signature 'LM'
summary(object, ...)

## S4 method for signature 'QGLM'
summary(object, ...)
```

Arguments

object	An object for which a summary is desired.
...	Further arguments passed to methods.

Value

The form of the value returned by summary() depends on the class of its argument.

Methods (by class)

- summary(EL): Summarizes the test results of the specified parameters.
- summary(ELMT): Summarizes the multiple testing results.
- summary(ELT): Summarizes the hypothesis test results.
- summary(GLM): Summarizes the results of the overall model test and the significance tests for coefficients. The dispersion parameter is extracted for display.
- summary(LM): Summarizes the results of the overall model test and the significance tests for coefficients.
- summary(QGLM): Summarizes the results of the overall model test and the significance tests for coefficients. The estimated dispersion parameter is extracted for display.

See Also

EL, ELMT, ELT, GLM, LM, QGLM,

Examples

```
data("faithful")
fit <- el_mean(faithful, par = c(3.5, 70))
summary(fit)

data("mtcars")
fit2 <- el_lm(mpg ~ wt, data = mtcars)
summary(fit2)
```

SummaryEL-class *SummaryEL class*

Description

S4 class for a summary of EL objects.

Slots

`optim` A list of the following optimization results:

- `par` A numeric vector of the specified parameters.
- `lambda` A numeric vector of the Lagrange multipliers of the dual problem corresponding to `par`.
- `iterations` A single integer for the number of iterations performed.
- `convergence` A single logical for the convergence status.
- `cstr` A single logical for whether constrained EL optimization is performed or not.

`logl` A single numeric of the empirical log-likelihood.

`loglr` A single numeric of the empirical log-likelihood ratio.

`statistic` A single numeric of minus twice the empirical log-likelihood ratio with an asymptotic chi-square distribution.

`df` A single integer for the degrees of freedom of the statistic.

`pval` A single numeric for the p-value of the statistic.

`nobs` A single integer for the number of observations.

`npar` A single integer for the number of parameters.

`weighted` A single logical for whether the data are weighted or not.

`coefficients` A numeric vector of the maximum empirical likelihood estimates of the parameters.

`method` A single character for the method dispatch in internal functions.

`control` An object of class ControlEL constructed by `el_control()`.

Examples

```
showClass("SummaryEL")
```

`SummaryELMT-class` *SummaryELMT class*

Description

S4 class for a summary of ELMT objects.

Slots

`aliased` A named logical vector showing if the original coefficients are aliased.

Examples

```
showClass("SummaryELMT")
```

`SummaryELT-class` *SummaryELT class*

Description

S4 class for a summary of ELT objects.

Slots

`optim` A list of the following optimization results:

- `par` A numeric vector of the solution to the (constrained) optimization problem.
- `lambda` A numeric vector of the Lagrange multipliers of the dual problem corresponding to `par`.
- `iterations` A single integer for the number of iterations performed.
- `convergence` A single logical for the convergence status.
- `cstr` A single logical for whether constrained EL optimization is performed or not.

`logl` A single numeric of the (constrained) empirical log-likelihood.

`loglr` A single numeric of the (constrained) empirical log-likelihood ratio.

`statistic` A single numeric of minus twice the (constrained) empirical log-likelihood ratio with an asymptotic chi-square distribution.

`df` A single integer for the chi-square degrees of freedom of the statistic.

`pval` A single numeric for the (calibrated) p-value of the statistic.

`cv` A single numeric for the critical value.

`rhs` A numeric vector for the right-hand side of the hypothesis.

`lhs` A numeric matrix for the left-hand side of the hypothesis.

`alpha` A single numeric for the significance level.

`calibrate` A single character for the calibration method used.

`control` An object of class ControlEL constructed by `el_control()`.

Examples

```
showClass("SummaryELT")
```

SummaryGLM-class *SummaryGLM class*

Description

S4 class for a summary of GLM objects. It inherits from SummaryLM class.

Slots

`family` A `family` object used.

`dispersion` A single numeric for the estimated dispersion parameter.

`coefficients` A numeric matrix of the results of significance tests.

`intercept` A single logical for whether the given model has an intercept term or not.

`na.action` Information returned by `model.frame` on the special handling of NAs.

`call` A matched call.

`terms` A `terms` object used.

`aliased` A named logical vector showing if the original coefficients are aliased.

`optim` A list of the following optimization results:

- `par` A numeric vector of the solution to the (constrained) optimization problem.
- `lambda` A numeric vector of the Lagrange multipliers of the dual problem corresponding to `par`.
- `iterations` A single integer for the number of iterations performed.
- `convergence` A single logical for the convergence status.
- `cstr` A single logical for whether constrained EL optimization is performed or not.

`logl` A single numeric of the empirical log-likelihood.

`loglr` A single numeric of the empirical log-likelihood ratio.

`statistic` A single numeric of minus twice the (constrained) empirical log-likelihood ratio for the overall test.

`df` A single integer for the degrees of freedom of the statistic.

`pval` A single numeric for the p-value of the statistic.

`nobs` A single integer for the number of observations.

`npar` A single integer for the number of parameters.

`weighted` A single logical for whether the data are weighted or not.

`method` A single character for the method dispatch in internal functions.

`control` An object of class ControlEL constructed by `el_control()`.

Examples

```
showClass("SummaryGLM")
```

SummaryLM-class *SummaryLM class*

Description

S4 class for a summary of LM objects.

Slots

`coefficients` A numeric matrix of the results of significance tests.

`intercept` A single logical for whether the given model has an intercept term or not.

`na.action` Information returned by `model.frame` on the special handling of NAs.

`call` A matched call.

`terms` A `terms` object used.

`aliased` A named logical vector showing if the original coefficients are aliased.

`optim` A list of the following optimization results:

- `par` A numeric vector of the solution to the (constrained) optimization problem.
- `lambda` A numeric vector of the Lagrange multipliers of the dual problem corresponding to `par`.
- `iterations` A single integer for the number of iterations performed.
- `convergence` A single logical for the convergence status.
- `cstr` A single logical for whether constrained EL optimization is performed or not.

`logl` A single numeric of the empirical log-likelihood.

`loglr` A single numeric of the empirical log-likelihood ratio.

`statistic` A single numeric of minus twice the (constrained) empirical log-likelihood ratio for the overall test.

`df` A single integer for the degrees of freedom of the statistic.

`pval` A single numeric for the p-value of the statistic.

`nobs` A single integer for the number of observations.

`npar` A single integer for the number of parameters.

`weighted` A single logical for whether the data are weighted or not.

`method` A single character for the method dispatch in internal functions.

`control` An object of class ControlEL constructed by `el_control()`.

Examples

```
showClass("SummaryLM")
```

SummaryQGLM-class *SummaryQGLM class*

Description

S4 class for a summary of QGLM objects. It inherits from SummaryGLM class.

Slots

`family` A `family` object used.

`dispersion` A single numeric for the estimated dispersion parameter.

`coefficients` A numeric matrix of the results of significance tests.

`intercept` A single logical for whether the given model has an intercept term or not.

`na.action` Information returned by `model.frame` on the special handling of NAs.

`call` A matched call.

`terms` A `terms` object used.

`aliased` A named logical vector showing if the original coefficients are aliased.

`optim` A list of the following optimization results:

- `par` A numeric vector of the solution to the (constrained) optimization problem.
- `lambda` A numeric vector of the Lagrange multipliers of the dual problem corresponding to `par`.
- `iterations` A single integer for the number of iterations performed.
- `convergence` A single logical for the convergence status.
- `cstr` A single logical for whether constrained EL optimization is performed or not.

`logl` A single numeric of the empirical log-likelihood.

`loglr` A single numeric of the empirical log-likelihood ratio.

`statistic` A single numeric of minus twice the (constrained) empirical log-likelihood ratio for the overall test.

`df` A single integer for the degrees of freedom of the statistic.

`pval` A single numeric for the p-value of the statistic.

`nobs` A single integer for the number of observations.

`npar` A single integer for the number of parameters.

`weighted` A single logical for whether the data are weighted or not.

`method` A single character for the method dispatch in internal functions.

`control` An object of class ControlEL constructed by `el_control()`.

Examples

```
showClass("SummaryQGLM")
```

`thiamethoxam` *Thiamethoxam applications in squash crops*

Description

A dataset on the effect of the thiamethoxam application method and plant variety on bees.

Usage

```
data("thiamethoxam")
```

Format

A data frame with 165 observations and 11 variables:

trt Treatment.

var Variety.

rep Replicate.

fruit Average fruit number per plant.

avg_mass Individual Fruit mass average (g).

mass Fruit mass per plant (g).

yield Yield (4 plants).

visit Bee visits per plot.

foliage Proportion of foliage consumed by striped cucumber beetle.

scb Striped cucumber beetle per plant.

defoliation Defoliation percentage.

Source

Obregon D, Pederson G, Taylor A, Poveda K (2022). "The Pest Control and Pollinator Protection Dilemma: The Case of Thiamethoxam Prophylactic Applications in Squash Crops." *PLOS ONE*, **17**(5), 1–18. doi:10.1371/journal.pone.0267984.

Examples

```
data("thiamethoxam")
thiamethoxam
```

`weights` *Model weights*

Description

Extracts weights from model objects. The weights are re-scaled to up to the total number of observations in the fitting procedure.

Usage

```
## S4 method for signature 'EL'
weights(object, ...)
```

Arguments

`object`	An object that inherits from EL.
`...`	Further arguments passed to methods.

Value

A numeric vector of the re-scaled weights.

References

Glenn N, Zhao Y (2007). "Weighted Empirical Likelihood Estimates and Their Robustness Properties." *Computational Statistics & Data Analysis*, **51**(10), 5130–5141. doi:10.1016/j.csda.2006.07.032.

See Also

EL

Examples

```
data("airquality")
x <- airquality$Wind
w <- airquality$Day
fit <- el_mean(x, par = 10, weights = w)
weights(fit)
```

Index

∗ **datasets**
clothianidin, 6
thiamethoxam, 54

as.data.frame(), *19*, *21*

CEL, *3*, *12*, *27*, *30*, *34*, *36*
CEL *(*CEL-class*)*, 3
CEL-class, 3
chisq, 5
chisq(), *5*, *44*
chisq,EL-method *(*chisq*)*, 5
chisq,ELMT-method *(*chisq*)*, 5
chisq,ELT-method *(*chisq*)*, 5
chisq,SummaryEL-method *(*chisq*)*, 5
chisq,SummaryELMT-method *(*chisq*)*, 5
chisq,SummaryELT-method *(*chisq*)*, 5
chisq,SummaryLM-method *(*chisq*)*, 5
clothianidin, 6
coef, 7
coef(), *7*
coef,EL-method *(*coef*)*, 7
coef,ELMT-method *(*coef*)*, 7
coef,SummaryEL-method *(*coef*)*, 7
coef,SummaryLM-method *(*coef*)*, 7
confint, 8
confint(), *9*, *16*
confint,EL-method *(*confint*)*, 8
confint,ELMT-method *(*confint*)*, 8
confreg, 9
confreg(), *8*, *16*, *41*
confreg,EL-method *(*confreg*)*, 9
ConfregEL, *9, 10*, *41*
ConfregEL *(*ConfregEL-class*)*, 10
ConfregEL-class, 10
ControlEL, *4*, *8–10*, *14*, *16, 17*, *19*, *21*, *23–25*, *27*, *29, 30*, *32*, *35*, *37*, *46*, *49–53*
ControlEL *(*ControlEL-class*)*, 10
ControlEL-class, 10
conv, 11
conv,CEL-method *(*conv*)*, 11
conv,EL-method *(*conv*)*, 11
conv,ELT-method *(*conv*)*, 11
conv,SummaryEL-method *(*conv*)*, 11
conv,SummaryELT-method *(*conv*)*, 11
conv,SummaryLM-method *(*conv*)*, 11
critVal, 12
critVal,ELMT-method *(*critVal*)*, 12
critVal,ELT-method *(*critVal*)*, 12
critVal,SummaryELMT-method *(*critVal*)*, 12
critVal,SummaryELT-method *(*critVal*)*, 12

EL, *3*, *6–9*, *12, 13*, *15*, *17, 18*, *20*, *22–31*, *33, 34*, *36*, *38–41*, *43–46*, *49*, *55*
EL *(*EL-class*)*, 13
EL-class, 13
el_control, 15
el_control(), *4*, *8, 9*, *14*, *17–32*, *35*, *37*, *46*, *49–53*
el_eval, 16
el_eval(), *16, 17*, *24*
el_glm, 18
el_glm(), *16*, *20*, *22*, *36*
el_lm, 21
el_lm(), *16*, *20*, *22*, *36*
el_mean, 23
el_mean(), *17*, *23*, *25*
el_sd, 24
el_sd(), *25*
ELD, *26*, *41*
ELD *(*ELD-class*)*, 26
eld, 25
eld(), *16*, *41*
eld,EL-method *(*eld*)*, 25
eld,GLM-method *(*eld*)*, 25
eld,missing-method *(*eld*)*, 25
ELD-class, 26
ELMT, *6–8*, *13*, *27, 28*, *33*, *43, 44*, *49, 50*
ELMT *(*ELMT-class*)*, 28
elmt, 27

elmt(), *16*, *27*, *31*
elmt,EL-method *(elmt)*, 27
ELMT-class, 28
ELT, *6*, *12*, *13*, *30*, *31*, *33*, *34*, *38*, *39*, *43*, *44*, *49*, *50*
ELT *(ELT-class)*, 31
elt, 29
elt(), *8*, *9*, *16*, *20*, *22*, *24*, *25*, *27*, *28*, *30*
elt,EL-method *(elt)*, 29
elt,missing-method *(elt)*, 29
elt,QGLM-method *(elt)*, 29
elt,SD-method *(elt)*, 29
ELT-class, 31

family, *35*, *45*, *51*, *53*
format(), *43*
formula, *18*, *21*
formula,LM-method *(LM-class)*, 36

getDF, 32
getDF,EL-method *(getDF)*, 32
getDF,ELMT-method *(getDF)*, 32
getDF,ELT-method *(getDF)*, 32
getDF,SummaryEL-method *(getDF)*, 32
getDF,SummaryELMT-method *(getDF)*, 32
getDF,SummaryLM-method *(getDF)*, 32
getOptim, 33
getOptim(), *12*, *47*
getOptim,EL-method *(getOptim)*, 33
getOptim,ELT-method *(getOptim)*, 33
getOptim,SummaryEL-method *(getOptim)*, 33
getOptim,SummaryELT-method *(getOptim)*, 33
getOptim,SummaryLM-method *(getOptim)*, 33
GLM, *20*, *34*, *45*, *49*, *51*
GLM *(GLM-class)*, 34
GLM-class, 34
glm.control(), *19*
glm.fit(), *19*, *20*

LM, *22*, *34*, *36*, *43*, *47*, *49*, *52*
LM *(LM-class)*, 36
LM-class, 36
lm.fit(), *22*
logL, 37
logL,EL-method *(logL)*, 37
logL,ELT-method *(logL)*, 37
logL,SummaryEL-method *(logL)*, 37
logL,SummaryELT-method *(logL)*, 37
logL,SummaryLM-method *(logL)*, 37
logLR, 38
logLR,EL-method *(logLR)*, 38
logLR,ELT-method *(logLR)*, 38
logLR,SummaryEL-method *(logLR)*, 38
logLR,SummaryELT-method *(logLR)*, 38
logLR,SummaryLM-method *(logLR)*, 38
logProb, 39
logProb,EL-method *(logProb)*, 39
logProb,ELT-method *(logProb)*, 39

model.frame, *51–53*

nobs, 40
nobs,EL-method *(nobs)*, 40
nobs,SummaryEL-method *(nobs)*, 40
nobs,SummaryLM-method *(nobs)*, 40

offset, *19*, *21*
options, *19*, *21*

par, *41*
plot, 41
plot(), *9*, *26*
plot,ConfregEL-method *(plot)*, 41
plot,EL-method *(plot)*, 41
plot,ELD-method *(plot)*, 41
print, 42
print,EL-method *(print)*, 42
print,ELMT-method *(print)*, 42
print,ELT-method *(print)*, 42
print,LM-method *(print)*, 42
print,SummaryEL-method *(print)*, 42
print,SummaryELMT-method *(print)*, 42
print,SummaryELT-method *(print)*, 42
print,SummaryGLM-method *(print)*, 42
print,SummaryLM-method *(print)*, 42
pVal, 43
pVal(), *6*, *44*
pVal,EL-method *(pVal)*, 43
pVal,ELMT-method *(pVal)*, 43
pVal,ELT-method *(pVal)*, 43
pVal,SummaryEL-method *(pVal)*, 43
pVal,SummaryELMT-method *(pVal)*, 43
pVal,SummaryELT-method *(pVal)*, 43
pVal,SummaryLM-method *(pVal)*, 43

QGLM, *45*, *49*, *53*
QGLM *(QGLM-class)*, 45

QGLM-class, 45

SD, *25*, *46*
SD *(*SD-class*)*, 46
SD-class, 46
set.seed(), *16*
sigTests, 47
sigTests(), *34*, *47*
sigTests,LM-method *(*sigTests*)*, 47
summary, 48
summary(), *48*
summary,EL-method *(*summary*)*, 48
summary,ELMT-method *(*summary*)*, 48
summary,ELT-method *(*summary*)*, 48
summary,GLM-method *(*summary*)*, 48
summary,LM-method *(*summary*)*, 48
summary,QGLM-method *(*summary*)*, 48
SummaryEL, *49*
SummaryEL *(*SummaryEL-class*)*, 49
SummaryEL-class, 49
SummaryELMT, *50*
SummaryELMT *(*SummaryELMT-class*)*, 50
SummaryELMT-class, 50
SummaryELT, *50*
SummaryELT *(*SummaryELT-class*)*, 50
SummaryELT-class, 50
SummaryGLM, *51*, *53*
SummaryGLM *(*SummaryGLM-class*)*, 51
SummaryGLM-class, 51
SummaryLM, *51, 52*
SummaryLM *(*SummaryLM-class*)*, 52
SummaryLM-class, 52
SummaryQGLM, *53*
SummaryQGLM *(*SummaryQGLM-class*)*, 53
SummaryQGLM-class, 53

terms, *35, 36*, *45*, *51–53*
thiamethoxam, 54

weights, 55
weights,EL-method *(*weights*)*, 55