Package ‘melt’

July 10, 2022

Type Package

Title Multiple Empirical Likelihood Tests

Version 1.6.0

Description Performs multiple empirical likelihood tests for linear and generalized linear models. The core computational routines are implemented using the 'Eigen' C++ library and 'RcppEigen' interface, with OpenMP for parallel computation. Details of multiple testing procedures are given in Kim, MacEachern, and Peruggia (2021) <arxiv:2112.09206>.

License GPL (>= 2)


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

Depends R (>= 4.0.0)

Imports graphics, methods, Rcpp, stats

Suggests covr, ggplot2, knitr, microbenchmark, 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 7.2.0

Author Eunseop Kim [aut, cre],
        Steven MacEachern [ctb, ths],
        Mario Peruggia [ctb, ths]

Maintainer Eunseop Kim <kim.7302@osu.edu>

Repository CRAN

Date/Publication 2022-07-10 09:20:02 UTC
Description

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

Let \( l(\theta) \) denote the 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 \( \chi^2_q \) under the constraint of hypothesis, i.e.,

\[
\min_{\theta; L\theta = r} l(\theta) \rightarrow_d \chi^2_q.
\]

Minimization of \( l(\theta) \) with respect to \( \theta \) 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 \( \gamma \), the algorithm updates \( \theta \) 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 with the following optimization results:

- `par` A numeric vector of the parameter value that minimizes the empirical likelihood subject to the constraints.
- `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.

References


Examples

`showClass("CEL")`
clothianidin  

*Clothianidin concentration in maize plants*

**Description**

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

**Usage**

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

**Format**

A data frame with 102 rows 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**


---

**coef**  

*Model coefficients*

**Description**

Extracts maximum empirical likelihood estimates from a model.

**Usage**

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

**Arguments**

- **object**: An object that inherit from `EL`, including `CEL`, `LM`, and `GLM`.
- **...**: Further arguments passed to methods.
A numeric vector of the maximum empirical likelihood estimates.

Examples

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

<table>
<thead>
<tr>
<th>confint</th>
<th>Confidence interval for model parameters</th>
</tr>
</thead>
</table>

Description

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

Usage

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

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

Arguments

- **object**: An object that inherit from EL, including CEL, LM, and GLM.
- **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().

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 \((1 - \text{level}) / 2\) and \(1 - (1 - \text{level}) / 2\) in %, respectively.

References

See Also

confreg(), el_control(), elt()

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 = el_control()
)

Arguments

- **object**: An object that inherit from EL, including CEL, LM, and GLM.
- **parm**: A specification of which parameters are to be given a confidence region, either a vector of numbers or a vector of names. It should be a vector of length two of the form (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().

Value

An object of class ConfregEL.
References


See Also

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

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)

Description

S4 class for confidence region.

Slots

points A numeric matrix with two columns for boundary points of a confidence region.
estimates A numeric vector of length two for 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 $\theta$.
- `maxit_l` A single integer for the maximum number of iterations for the optimization with respect to $\lambda$.
- `tol` A single numeric for the convergence tolerance denoted by $\epsilon$. The iteration stops when 
  \[ \| P \nabla l(\theta^{(k)}) \| < \epsilon. \]
- `tol_l` A single numeric for the relative convergence tolerance denoted by $\delta$. The iteration stops when 
  \[ \| \lambda^{(k)} - \lambda^{(k-1)} \| < \delta \| \lambda^{(k-1)} \| + \delta^2. \]
- `step` A single numeric for the step size $\gamma$ 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 data.
- `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.
- `b` A single integer for the number of bootstrap replicates.
- `m` A single integer for the number of Monte Carlo samples.

**See Also**

- `el_control()`

**Examples**

```r
to showClass("ControlEL")
```
Convergence check

Description

Extracts convergence status from a model.

Usage

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

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

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

Arguments

- `object`: An object that inherit from EL, including CEL, LM, and GLM.
- `...`: Further arguments passed to methods.

Value

A single logical.

Methods (by class)

- EL: Extracts the convergence status of the model with respect to the Lagrange multiplier.
- CEL: Extracts the convergence status of the model with respect to the parameter.
- ELT: Extracts the convergence status of the model with respect to the parameter (or the Lagrange multiplier if lhs is NULL).

Examples

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

S4 class for empirical likelihood.

Details

Let $X_i$ be independent and identically distributed $p$-dimensional random variable from an unknown distribution $F$ for $i = 1, \ldots, n$. We assume that $F$ 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

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

We assume that there exists an unique $\theta_0$ that solves the above equation. Given a value of $\theta$, 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 $\lambda$ 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 $\{g(X_i, \theta)\}_{i=1}^n$.

Under some regularity conditions, it is known that $-2 \log R(\theta_0)$ converges in distribution to $\chi^2_p$, where $\chi^2_p$ has a chi-square distribution with $p$ degrees of freedom.

Slots

- optim: A list with the following optimization results:
  - par: A numeric vector of the specified parameters.
  - 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.
eld

logp  A numeric vector of the log probabilities obtained from empirical likelihood.
logl  A single numeric for the empirical log-likelihood.
loglr A single numeric for the empirical log-likelihood ratio.
statistic A single numeric for the minus twice the empirical log-likelihood ratio statistic that
          has 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.
npar  A single integer for the number of parameters.
weights A numeric vector of re-scaled weights used for model fitting.
data  A numeric matrix for the data used for 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.

References


Examples

showClass("EL")

eld

Empirical likelihood displacement

Description

Computes empirical likelihood displacement for model diagnostics and outlier detection.

Usage

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

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

Arguments

object       An object that inherit from EL, including CEL, LM, and GLM.
control      An object of class ControlEL constructed by el_control().
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 $i$th 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 $i$th 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


See Also

el_control(), el_eval(), plot()

Examples

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

Arguments

- **object**: An object that inherit from EL, including CEL, LM, and GLM.
- **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 numeric matrix or a list of numeric matrices for the left-hand sides of hypothesis. Each row of the matrices gives a linear combination of the parameters in `object`. The number of columns should 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()`.

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

See Also

el_control(), elt()

Examples

```r
## Example 1: bivariate mean (list `rhs` & no `lhs`)
data("women")
fit <- el_mean(women, par = c(65, 135))
rhs <- list(c(64, 133), c(66, 140))
set.seed(143)
elmt(fit, rhs = rhs)

## Example 2: pairwise comparison (no `rhs` & matrix `lhs`)
data("clothianidin")
fit2 <- el_lm(clo ~ -1 + trt, clothianidin)
lhs <- matrix(c(
  1, -1, 0, 0,
  0, 1, -1, 0,
  0, 0, 1, -1
), byrow = TRUE, nrow = 3)
set.seed(629)
elmt(fit2, lhs = lhs)

## Example 3: arbitrary hypotheses (list `rhs` & list `lhs`)
data("mtcars")
fit <- el_lm(mpg ~ wt + qsec, data = mtcars)
lhs <- list(rbind(c(1, 4, 0)), rbind(c(0, 1, 0), c(0, 0, 1)))
rhs <- list(0, c(-6, 1))
elmt(fit, rhs = rhs, lhs = lhs)
```

ELMT-class

ELMT class

Description

S4 class for empirical likelihood multiple tests.

Slots

- `alpha` A single numeric for the overall significance level.
- `statistic` A numeric vector for the minus twice the (constrained) empirical log-likelihood ratios.
- `cv` A single numeric for the multiplicity adjusted critical value.
- `pval` A numeric vector for the multiplicity adjusted p-values.
- `calibrate` A single character for the calibration method used.

Examples

```r
showClass("ELMT")
```
**elt**

*Empirical likelihood test*

**Description**

Tests a linear hypothesis.

**Usage**

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

**Arguments**

- **object**: An object that inherit from `EL`, including `CEL`, `LM`, and `GLM`.
- **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 hypothesis. Each row gives a linear combination of the parameters in `object`. The number of columns should be equal to the number of parameters. Defaults to `NULL`. See ‘Details’.
- **alpha**: A single numeric for the significance level. Defaults to `0.05`.
- **calibrate**: A single character for the calibration method. It is case-insensitive and must be one of "chisq", "boot", or "f". Defaults to "chisq". See ‘Details’.
- **control**: An object of class `ControlEL` constructed by `el_control()`.

**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 as $\inf_{\theta: \theta = r} l(\theta)$.
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. Three methods are available: "chisq" (chi-square calibration), "boot" (bootstrap calibration), and "f" ($F$ calibration). "boot" is applicable only when `lhs` is `NULL`. The nthreads, seed, and B slots in control apply to the bootstrap procedure. "f" is applicable only to the mean parameter when `lhs` is `NULL`.

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


See Also

`el_control()`, `elmt()`

Examples

```r
## F calibration for the mean
set.seed(533414)
x <- rnorm(100)
fit <- el_mean(x, 0)
elt(fit, rhs = 0.3, calibrate = "f")

## Test of no treatment effect
data("clothianidin")
lhs <- 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 = lhs)
```

ELT-class

S4 class for empirical likelihood test.
Slots

- optim A list with the optimization results.
- alpha A single numeric for the significance level.
- logl A single numeric for the (constrained) empirical log-likelihood.
- loglr A single numeric for the (constrained) empirical log-likelihood ratio.
- statistic A single numeric for the minus twice the (constrained) empirical log-likelihood ratio.
- cv A single numeric for the critical value.
- pval A single numeric for the \( p \) -value of the statistic.
- calibrate A single character for the calibration method used.

Examples

showClass("ELT")

---

el_control

Control parameters for computation

Description

Specifies computational details of (constrained) empirical likelihood.

Usage

```r
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 = sample.int(.Machine$integer.max, 1L), 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 set to the reciprocal of 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 $200 \times d$, 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 kept in the data slot 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 the half of the available threads. For better performance, it is generally recommended to limit the number of threads to the number of physical cores. Note that it only applies to the following functions that involve multiple evaluations or minimizations: 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 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.

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

```r
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
el_eval evaluates empirical likelihood with a \( n \times p \) numeric matrix argument g, whose \( i \)th 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 with the following components:
- **optim**: A list with the following optimization results:
  - **lambda**: Lagrange multiplier of the dual problem.
  - **iterations**: Number of iterations performed.
  - **convergence**: Convergence status.
- **logp**: Log probabilities obtained from empirical likelihood.
- **logl**: Empirical log-likelihood.
- **loglr**: Empirical log-likelihood ratio.
- **statistic**: Minus twice the empirical log-likelihood ratio statistic that has an asymptotic chi-square distribution.
- **df**: Degrees of freedom of the statistic.
- **pval**: \( p \)-value of the statistic.
- **npar**: Number of parameters.
- **weights**: Re-scaled weights used for model fitting.
References


See Also

el_control()

Examples

set.seed(3271)
x <- rnorm(50)
par <- 0
g <- x - par
el_eval(g, weights = rep(c(1, 2), each = 25))

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,
  control = el_control(),
  start = NULL,
  etastart = NULL,
  mustart = NULL,
  ...
)

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 NAs. The default is set by the na.action setting of options, and is na.fail if that is unset.

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

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().

...  Additional arguments to be passed to glm.control().

Details

The available families and link functions are as follows:

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

Included in the tests are the overall test with

\[ H_0 : \beta_1 = \beta_2 = \cdots = \beta_{p-1} = 0, \]

and the tests for each parameter with

\[ H_{0j} : \beta_j = 0, \ j = 0, \ldots, p - 1. \]

The test results are returned as optim and parTests, respectively.

Value

An object of class of GLM.

References


See Also

el_control(), el_lm(), elt()
Examples

```r
cat("set.seed(20010)\n")
n <- 50
x <- rnorm(n)
x2 <- rnorm(n)
l <- -2 + 0.2 * x + 3 * x2
mu <- 1 / (1 + exp(-l))
y <- rbinom(n, 1, mu)
df <- data.frame(y, x, x2)
fit <- el_glm(y ~ x + x2,
             family = binomial, df, weights = NULL,
             control = el_control(na.action = na.omit, start = NULL, etastart = NULL, mustart = NULL))
save(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, 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.
- `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 \((X_i, Y_i)\) 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 regression 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 \( \epsilon_i \) are independent random variables that satisfy \( \mathbb{E}(\epsilon_i | X_i) = 0 \). We assume that the errors have finite conditional variance. Then the least square estimator of \( \theta \) solves the following estimating equation:

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

\textit{el_lm()} first computes the parameter estimates by calling \texttt{lm.fit()} (with ... if any) with the \texttt{model.frame} and \texttt{model.matrix} obtained from the formula. Note that the maximum empirical likelihood estimator is the same as the least square estimator in our model. Next, it performs hypothesis tests based on an asymptotic chi-squared distribution of 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 test results are returned as \texttt{optim} and \texttt{parTests}, respectively.

Value

An object of class of \texttt{LM}.

References


See Also

\textit{el_control()}, \textit{el_glm()}, \textit{elt()}

Examples

```r
set.seed(5649)
df <- data.frame(y = rnorm(50), x = rnorm(50))
fit <- el_lm(y ~ x, df)
summary(fit)

fit2 <- el_lm(y ~ x, df, weights = rep(c(1, 2), each = 25))
summary(fit2)

df[1, 2] <- NA
fit3 <- el_lm(y ~ x, df, na.action = na.omit)
summary(fit3)
```
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()`.

Value
An object of class `EL`.

References

See Also
- `el_control()`, `el_eval()`, `elt()`

Examples
```r
## Scalar mean
set.seed(414)
x <- rnorm(100)
par <- 0
el_mean(x, par)

## Vector mean
x <- matrix(rnorm(100), ncol = 2)
par <- c(0, 0)
```
el_sd

el_mean(x, par)

## Weighted data
x <- matrix(rnorm(100), ncol = 2)
par <- c(0, 0)
w <- rep(c(1, 2), each = 25)
el_mean(x, par, w)

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.

ccontrol
An object of class ControlEL constructed by el_control().

Value
An object of class EL.

See Also
el_control(), el_mean(), elt()

Examples
set.seed(4097)
x <- rnorm(100, mean = -2, sd = 3)
el_sd(x, mean = -2, sd = 3.5)
GLM-class

**Description**

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

**Examples**

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

LM-class

**Description**

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

**Details**

If there is no intercept in a model, optim slot need to be understood in terms of EL class since constrained optimization is not involved in the overall test.

**Methods (by generic)**

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

**Slots**

- `parTests`: A list with the test results for each parameter:
  - `statistic`: A numeric vector of the empirical likelihood ratio statistics.
  - `convergence`: A logical vector of the convergence status of tests for each parameter.

- `misc`: A list with miscellaneous outputs from a model fitting function. They are used in other generics and methods.

**Examples**

```r
showClass("LM")
```
logLik

Description

Extracts empirical log-likelihood from a model evaluated at the estimated coefficients.

Usage

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

Arguments

- `object`: An object that inherit from `EL`.
- `...`: Further arguments passed to methods.

Value

An object of class `logLikEL`.

Examples

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

logLikEL-class

Description

S4 class for empirical log-likelihood.

Slots

- `logLik`: A single numeric for the empirical log-likelihood.
- `df`: A single integer for the degrees of freedom or the number of (estimated) parameters in the model.

Examples

```r
showClass("logLikEL")
```
logLR

**Empirical log-likelihood ratio**

**Description**
Extracts empirical log-likelihood ratio from a model.

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

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

**Arguments**
- **object**: An object that inherit from `EL` or `ELT`.
- **...**: Further arguments passed to methods.

**Value**
A single numeric.

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

**Arguments**
- **object**: An object that inherit from `EL`.
- **...**: Further arguments passed to methods.
Value

A single integer.

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 'ELD'
plot(x, y, ...)

Arguments

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

Methods (by class)

- ConfregEL: Plots a two-dimensional confidence region for model parameters.
- ELD: Plots empirical likelihood displacement values versus observation index.

See Also

confreg(), eld()
## Examples

```r
## 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)
```

## Print methods

**Description**

Provides print methods for objects.

**Usage**

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

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

## S4 method for signature 'logLikEL'
print(x, digits = getOption("digits"), ...)

## S4 method for signature 'ELMT'
print(
  x,
  digits = getOption("digits"),
  signif.stars = getOption("show.signif.stars"),
  ...
)

## S4 method for signature 'ELT'
print(x, digits = getOption("digits"), ...)
```
**SD-class**

**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.

**Examples**

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

---

**SD-class** *SD class*

**Description**

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

**Examples**

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

---

**summary** *Summary methods*

**Description**

Provides summary methods for objects.

**Usage**

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

**Arguments**

- `object` An object to be summarized.
- `...` Further arguments passed to methods.

**Methods (by class)**

- `LM`: Summarizes the results of the overall test and the tests for each parameter.
Examples

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

weights

Description

S4 class for a summary of LM objects.

Slots

statistic A single numeric for the minus twice the empirical log-likelihood ratio for the overall
test of the model.
df A single integer for the degrees of freedom of the statistic.
convergence A single logical for the convergence status of the constrained minimization.
parMatrix A numeric matrix of the test results of the parameters.
weighted A single logical for whether the given model is weighted or not.
na.action Information returned by model.frame on the special handling of NAs.
call Matched call.
terms terms object used.
aliased A named logical vector showing if the original coefficients are aliased.

Examples

showClass("SummaryLM")

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, ...)

weights
weights

Arguments

object     An object that inherit from EL, including CEL, LM, and GLM.

...        Further arguments passed to methods.

Value

A numeric vector of the re-scaled weights.

References


Examples

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

* datasets
  clothianidin, 4
  as.data.frame(), 20, 22

CEL, 2, 4–6, 9, 11, 13, 15, 16, 26, 33
CEL (CEL-class), 2
CEL-class, 2
clothianidin, 4
coeff, 4
coeff,EL-method (coeff), 4
confint, 5
confint(), 7, 18
confint,EL-method (confint), 5
confint,SD-method (confint), 5
confreg, 6
confreg(), 6, 18, 29
confreg,EL-method (confreg), 6
ConfregEL, 6, 7
ConfregEL (ConfregEL-class), 7
ConfregEL-class, 7
ControlEL, 5, 6, 8, 11, 13, 15, 18, 19, 21, 22, 24, 25
ControlEL (ControlEL-class), 8
ControlEL-class, 8
conv, 9
conv,CEL-method (conv), 9
conv,EL-method (conv), 9
conv,ELT-method (conv), 9

EL, 2–6, 9–11, 13, 15, 16, 18, 19, 24–28, 31, 33
EL (EL-class), 10
EL-class, 10
el_control, 17
el_control(), 5–8, 11–16, 19–25
el_eval, 19
el_eval(), 12, 18, 24
el_glm, 20
el_glm(), 18, 23, 26
el_lm, 22

el_lm(), 18, 21, 23, 26
el_mean, 24
el_mean(), 25
el_sd, 25
ELD, 12
ELD (ELD-class), 12
eIcl, 11
eIcl(), 18, 29
eIcl,EL-method (elcl), 11
eIcl,GLM-method (elcl), 11
eIcl,missing-method (elcl), 11
ELD-class, 12
ELMT, 13, 14
ELMT (ELMT-class), 14
eImlt, 13
eImlt(), 13, 16, 18
eImlt,EL-method (elmlt), 13
ELMT-class, 14
ELT, 16, 28
ELT (ELT-class), 16
eIlt, 15
eIlt(), 6, 7, 13–15, 18, 21, 23–25
eIlt,EL-method (elit), 15
eIlt,missing-method (elit), 15
eIlt,SD-method (elit), 15
ELT-class, 16

format(), 31
formula, 20, 22
formula,LM-method (LM-class), 26

GLM, 4–6, 9, 11, 13, 15, 21, 26, 33
GLM (GLM-class), 26
GLM-class, 26
glm.control(), 21
glm.fit(), 21

LM, 4–6, 9, 11, 13, 15, 23, 26, 32, 33
LM (LM-class), 26
LM-class, 26
lm.fit(), 23
logLik, 27
logLik,EL-method (logLik), 27
logLikEL, 27
logLikEL (logLikEL-class), 27
logLikEL-class, 27
logLR, 28
logLR,EL-method (logLR), 28
logLR,ELT-method (logLR), 28

model.frame, 32

nobs, 28
nobs,EL-method (nobs), 28

options, 21, 22

par, 29
plot, 29
plot(), 7, 12
plot,ConfregEL-method (plot), 29
plot,ELD-method (plot), 29
print, 30
print,EL-method (print), 30
print,ELMT-method (print), 30
print,ELT-method (print), 30
print,logLikEL-method (print), 30
print,SummaryLM-method (print), 30

SD, 31
SD (SD-class), 31
SD-class, 31
set.seed(), 18
summary, 31
summary,LM-method (summary), 31
SummaryLM, 32
SummaryLM (SummaryLM-class), 32
SummaryLM-class, 32

terms, 32

weights, 32
weights,EL-method (weights), 32