Package 'MCPModBC'

May 16, 2023

Title Improved Inference in Multiple Comparison Procedure – Modelling

Version 1.1

Date 2023-05-15

Description Implementation of Multiple Comparison Procedures with Modeling (MCP-Mod) procedure with bias-corrected estimators and second-order covariance matrices as described in Diniz, Gallardo and Magalhaes (2023) <doi:10.1002/pst.2303>.

License GPL (>= 2)

Encoding UTF-8

Depends R (>= 4.0.0), stats

Imports doRNG, survival, doParallel, nleqslv, foreach, DoseFinding, dplyr, rlang

RoxygenNote 7.2.1

NeedsCompilation no

Author Marcio Diniz [aut], Diego Gallardo [aut, cre], Tiago Magalhaes [aut]

Maintainer Diego Gallardo <diego.gallardo@uda.cl>

Repository CRAN

Date/Publication 2023-05-15 23:40:13 UTC

R topics documented:

`data_generator` *Data Generator*

Description

It generates data for a dose-finding trial.

Usage

```
data_generator(doses, sample.size, distr, parm,
censoring.rate = NULL)
```

Arguments

`doses`	a numeric vector indicating the doses that will be considered in the clinical trial.
`sample.size`	a numeric value indicating the sample size per dose in the clinical trial.
`distr`	a character value indicating the distribution of the response variable. Currently, the only option available is 'weibull'.
`parm`	a named list of true values for the simulation. See mode in Details.
`censoring.rate`	a numeric value between 0 and 1 indicating the censoring rate when generated data. It is required when `distr = "weibull"`.

Details

If `distr = "weibull"`, the list `parm` should contain two components - `lambda` and `sigma` - that are the scale and shape parameters in the following parametrization of the Weibull distribution:

$$f(t;\lambda,\sigma) = \frac{1}{\sigma\lambda^{1/\sigma}} t^{1/\sigma-1} \exp\left\{-(t/\lambda)^{1/\sigma}\right\},\ t>0,$$

with hazard rate given by

$$h(t) = \frac{1}{\lambda^{1/\sigma}\sigma} t^{1/\sigma-1}$$

and regression structure

$$\log(\lambda_i) = d_i\beta_i.$$

where $\log(\lambda_i)$ represents the model effect for dose i, `doses[i]`.

Value

a data frame of dimension [length(doses)$\times$sample.size] $\times$ 3 when `distr = "weibull"` containing time-to-event, censoring indicator and dose.

References

Diniz, Márcio A. and Gallardo, Diego I. and Magalhães, Tiago M. (2023). Improved inference for MCP-Mod approach for time-to-event endpoints with small sample sizes. arXiv <doi.org/10.48550/arXiv.2301.00325>

Examples

```
library(DoseFinding)
library(MCPModBC)

## doses scenarios
doses <- c(0, 5, 25, 50, 100)
nd <- length(doses)

# median survival time for placebo dose
mst.control <- 4

# shape parameter
sigma.true <- 0.5

# maximum hazard ratio between active dose and placebo dose
hr.ratio <- 4
# minimum hazard ratio between active dose and placebo dose
hr.Delta <- 2

# hazard rate for placebo dose
placEff <- log(mst.control/(log(2)^sigma.true))

# maximum hazard rate for active dose
maxEff <- log((mst.control*(hr.ratio^sigma.true))/(log(2)^sigma.true))

# minimum hazard rate for active dose
minEff.Delta <- log((mst.control*(hr.Delta^sigma.true))/(log(2)^sigma.true))
Delta <- (minEff.Delta - placEff)

## MCP Parameters
emax <- guesst(d = doses[4], p = 0.5, model="emax")
exp <- guesst(d = doses[4], p = 0.1, model="exponential", Maxd = doses[nd])
logit <- guesst(d = c(doses[3], doses[4]), p = c(0.1,0.8), "logistic",
Maxd= doses[nd])
betam <- guesst(d = doses[2], p = 0.3, "betaMod", scal=120, dMax=50,
Maxd= doses[nd])

models.candidate <- Mods(emax = emax, linear = NULL,
                         exponential = exp, logistic = logit,
                         betaMod = betam, doses = doses,
                         placEff = placEff, maxEff = (maxEff- placEff))
plot(models.candidate)

## True Model
model.true <- "emax"
response <- model_response(doses = doses,
                           distr = "weibull",
                           model.true = model.true,
                           models.candidate = models.candidate)
lambda.true <- response$lambda
parm <- list(lambda = lambda.true, sigma = sigma.true)
```

```
## Scenario: Censoring 10%
censoring.rate <- 0.1

dt <- data_generator(doses = doses,
                     sample.size = 20,
                     distr = "weibull",
                     parm = parm,
                     censoring.rate = censoring.rate)
## Print data
#dt
```

`mcpmod_simulation` *Simulation to obtain operating characteristics for MCP-Mod design*

Description

It simulates dose-finding trials using MCP-Mod design with Maximum Likelihood Estimator and Fisher Information (MLE), Maximum Likelihood Estimator and second-order Fisher Information (MLE2), Cox and Snell's Bias-Corrected Estimator and Fisher Information (BCE), Cox and Snell's Bias-Corrected Estimator and second-order Fisher Information (BCE2), and Firth Bias-Corrected estimators (Firth) as discussed in Diniz, Magalhães and Gallardo.

Usage

```
mcpmod_simulation(doses, parm, sample.size, model.true,
  models.candidate, selModel = "AIC",significance.level = 0.025,
  Delta, distr = "weibull", censoring.rate = NULL,
  sigma.estimator = NULL, n.cores, seed,n.sim)
```

Arguments

`doses`	a numeric vector indicating the doses that will be considered in the clinical trial.
`parm`	a named list of true values for the simulation. See more details in `data_generator`.
`sample.size`	a numeric vector indicating the sample sizes per dose in the clinical trial to be evaluated in the simulation study.
`model.true`	a character value indicating the true functional form of dose-response curve. See more details in `model_response`.
`models.candidate`	an object of class 'Mods'. See more details in `Mods`.
`selModel`	a character value indicating the model selection criterion for dose estimation. See more details in `MCPMod`.
`significance.level`	a numeric value indicating the significance level when evaluating proof-of-concept based on an one-sided Wald test.
`Delta`	a numerical value indicating the target effect size used for the target dose. See `TD`.

`distr`	a character value indicating the distribution of the response variable. Currently, the only option available is 'weibull'.
`censoring.rate`	a numeric value between 0 and 1 indicating the censoring rate when generated data. It is required when `distr = "weibull"`.
`sigma.estimator`	a character value indicating whether the estimator for sigma should be a maximum likelihood or jackknife estimator. It is required when `distr = "weibull"`. Options are "mle" and "jackknife".
`n.cores`	a numeric value indicating the number of cores to be used in the 'simulation performed in parallel. Use parallel::detectCores() to check the number of cores available.
`seed`	an integer value, containing the random number generator (RNG) state for random number generation.
`n.sim`	a numerical value indicating the number of simulated trials.

Value

An object of class mcpmod_simulation with the following components:

`mle`: a matrix of dimension n.sim $\times$ 4 with results when using the MCP-Mod approach with MLE;

`mle2`: a matrix of dimension n.sim $\times$ 4 with results when using the MCP-Mod approach with MLE2;

`bce`: a matrix of dimension n.sim $\times$ 4 with results when using the MCP-Mod approach with BCE;

`bce2`: a matrix of dimension n.sim $\times$ 4 with results when using the MCP-Mod approach with BCE2;

`firth`: a matrix of dimension n.sim $\times$ 4 with results when using the MCP-Mod approach with Firth's estimator;

All matrices contain the following columns: (1) the first column indicates whether proof-of-concept (1 = "yes", 0 = "no"), in other words, the p-value of Wald test was statistically significant; (2) the second column indicates whether the true model was selected to estimate the dose-response curve (1 = "yes", 0 = "no") when proof-of-concept is demonstrated; (3) the third column contains the estimated target dose; (4) the fourth column contains the sample size considered in the trial.

`conditions`: a list containing the conditions of the simulation.

Author(s)

Diniz, M.A., Gallardo D.I., Magalhaes, T.M.

References

Bretz F, Pinheiro JC, Branson M. Combining multiple comparisons and modeling techniques in dose-response studies. Biometrics. 2005 Sep;61(3):738-48.

Bornkamp B, Pinheiro J, Bretz F. MCPMod: An R package for the design and analysis of dose-finding studies. Journal of Statistical Software. 2009 Feb 20;29:1-23.

Diniz, Márcio A. and Gallardo, Diego I. and Magalhães, Tiago M. (2023). Improved inference for MCP-Mod approach for time-to-event endpoints with small sample sizes. arXiv <doi.org/10.48550/arXiv.2301.00325>

Pinheiro J, Bornkamp B, Glimm E, Bretz F. Model-based dose finding under model uncertainty using general parametric models. Statistics in medicine. 2014 May 10;33(10):1646-61.

Examples

```
library(DoseFinding)
library(MCPModBC)

## doses scenarios
doses <- c(0, 5, 25, 50, 100)
nd <- length(doses)
sample.size <- 25

# shape parameter
sigma.true <- 0.5

# median survival time for placebo dose
mst.control <- 4

# maximum hazard ratio between active dose and placebo dose
hr.ratio <- 4
# minimum hazard ratio between active dose and placebo dose
hr.Delta <- 2

# hazard rate for placebo dose
placEff <- log(mst.control/(log(2)^sigma.true))

# maximum hazard rate for active dose
maxEff <- log((mst.control*(hr.ratio^sigma.true))/(log(2)^sigma.true))

# minimum hazard rate for active dose
minEff.Delta <- log((mst.control*(hr.Delta^sigma.true))/(log(2)^sigma.true))
Delta <- (minEff.Delta - placEff)

## MCP Parameters
significance.level <- 0.05
selModel <- "AIC"

emax <- guesst(d = doses[4], p = 0.5, model="emax")
exp <- guesst(d = doses[4], p = 0.1, model="exponential", Maxd = doses[nd])
logit <- guesst(d = c(doses[3], doses[4]), p = c(0.1,0.8), "logistic",
Maxd= doses[nd])
betam <- guesst(d = doses[2], p = 0.3, "betaMod", scal=120, dMax=50,
Maxd= doses[nd])

models.candidate <- Mods(emax = emax, linear = NULL,
                         exponential = exp, logistic = logit,
                         betaMod = betam, doses = doses,
                         placEff = placEff, maxEff = (maxEff- placEff))
plot(models.candidate)

## Simulation Parameters
n.sim <- 10
seed <- 1234
n.cores <- 1
```

```
## True Model
model.true <- "emax"
response <- model_response(doses = doses,
                           distr = "weibull",
                           model.true = model.true,
                           models.candidate = models.candidate)
lambda.true <- response$lambda
parm <- list(lambda = lambda.true, sigma = sigma.true)

## Scenario: Censoring 10%
censoring.rate <- 0.1

test <- mcpmod_simulation(doses = doses,
           parm = parm, sample.size = sample.size,
           model.true = model.true,
           models.candidate = models.candidate,
           selModel = selModel,
           significance.level = significance.level,
           Delta = Delta, distr = "weibull",
           censoring.rate = censoring.rate,
           sigma.estimator = "jackknife",
           n.cores = n.cores, seed = seed, n.sim = n.sim)

test
summary(test)
```

`model_response` *Model Responser*

Description

It calculates the model response and parameters of interest for a given distribution.

Usage

```
model_response(doses, distr, model.true, models.candidate)
```

Arguments

`doses`	a numeric vector indicating the doses that will be considered in the clinical trial.
`distr`	a character value indicating the distribution of the response variable. Currently, the only option available is 'weibull'.
`model.true`	a character value indicating the functional form of the true dose-response curve. Options are "constant", "linear", "linlog", "quadratic", "exponential", "emax", "sigmaEmax", "betaMod", "logistic", "linInt".
`models.candidate`	an object of class Mods. See more details in `Mods`.

Value

a data frame with dimension length(doses) $\times$ 3 with the following columns: (1) model response (2) model parameter and (3) doses

Author(s)

Diniz, M.A., Gallardo D.I., Magalhaes, T.M.

References

Diniz, Márcio A. and Gallardo, Diego I. and Magalhães, Tiago M. (2023). Improved inference for MCP-Mod approach for time-to-event endpoints with small sample sizes. arXiv <doi.org/10.48550/arXiv.2301.00325>

Examples

```
library(DoseFinding)
library(MCPModBC)

## doses scenarios
doses <- c(0, 5, 25, 50, 100)
nd <- length(doses)

# median survival time for placebo dose
mst.control <- 4

# shape parameter
sigma.true <- 0.5

# maximum hazard ratio between active dose and placebo dose
hr.ratio <- 4
# minimum hazard ratio between active dose and placebo dose
hr.Delta <- 2

# hazard rate for placebo dose
placEff <- log(mst.control/(log(2)^sigma.true))

# maximum hazard rate for active dose
maxEff <- log((mst.control*(hr.ratio^sigma.true))/(log(2)^sigma.true))

# minimum hazard rate for active dose
minEff.Delta <- log((mst.control*(hr.Delta^sigma.true))/(log(2)^sigma.true))
Delta <- (minEff.Delta - placEff)

## MCP Parameters
emax <- guesst(d = doses[4], p = 0.5, model="emax")
exp <- guesst(d = doses[4], p = 0.1, model="exponential", Maxd = doses[nd])
logit <- guesst(d = c(doses[3], doses[4]), p = c(0.1,0.8), "logistic",  Maxd= doses[nd])
betam <- guesst(d = doses[2], p = 0.3, "betaMod", scal=120, dMax=50, Maxd= doses[nd])

models.candidate <- Mods(emax = emax, linear = NULL,
                         exponential = exp, logistic = logit,
```

```
                          betaMod = betam, doses = doses,
                          placEff = placEff, maxEff = (maxEff- placEff))
plot(models.candidate)

## True Model
model.true <- "emax"
response <- model_response(doses = doses,
                           distr = "weibull",
                           model.true = model.true,
                           models.candidate = models.candidate)
response

lambda.true <- response$lambda
parm <- list(lambda = lambda.true, sigma = sigma.true)
```

`summary.mcpmod_simulation` *Summary of simulation results*

Description

It summarizes results of simulations of dose-finding trials following the MCP-Mod approach with bias-corrected and second-order covariance matrices.

Usage

```
## S3 method for class 'mcpmod_simulation'
summary(object, ...)
```

Arguments

`object`	an object of the "mcpmod_simulation" class.
`...`	additional arguments affecting the summary produced.

Value

A data frame with a summary with the information provided by `mcpmod_simulation`.

References

Diniz, Márcio A. and Gallardo, Diego I. and Magalhães, Tiago M. (2023). Improved inference for MCP-Mod approach for time-to-event endpoints with small sample sizes. arXiv <doi.org/10.48550/arXiv.2301.00325>

Examples

```
library(DoseFinding)
library(MCPModBC)

## doses scenarios
doses <- c(0, 5, 25, 50, 100)
nd <- length(doses)
sample.size <- 25

# shape parameter
sigma.true <- 0.5

# median survival time for placebo dose
mst.control <- 4

# maximum hazard ratio between active dose and placebo dose
hr.ratio <- 4
# minimum hazard ratio between active dose and placebo dose
hr.Delta <- 2

# hazard rate for placebo dose
placEff <- log(mst.control/(log(2)^sigma.true))

# maximum hazard rate for active dose
maxEff <- log((mst.control*(hr.ratio^sigma.true))/(log(2)^sigma.true))

# minimum hazard rate for active dose
minEff.Delta <- log((mst.control*(hr.Delta^sigma.true))/(log(2)^sigma.true))
Delta <- (minEff.Delta - placEff)

## MCP Parameters
significance.level <- 0.05
selModel <- "AIC"

emax <- guesst(d = doses[4], p = 0.5, model="emax")
exp <- guesst(d = doses[4], p = 0.1, model="exponential", Maxd = doses[nd])
logit <- guesst(d = c(doses[3], doses[4]), p = c(0.1,0.8), "logistic",
Maxd= doses[nd])
betam <- guesst(d = doses[2], p = 0.3, "betaMod", scal=120, dMax=50,
Maxd= doses[nd])

models.candidate <- Mods(emax = emax, linear = NULL,
                         exponential = exp, logistic = logit,
                         betaMod = betam, doses = doses,
                         placEff = placEff, maxEff = (maxEff- placEff))
plot(models.candidate)

## Simulation Parameters
n.sim <- 10
seed <- 1234
n.cores <- 1
```

```
## True Model
model.true <- "emax"
response <- model_response(doses = doses,
                           distr = "weibull",
                           model.true = model.true,
                           models.candidate = models.candidate)
lambda.true <- response$lambda
parm <- list(lambda = lambda.true, sigma = sigma.true)

## Scenario: Censoring 10%
censoring.rate <- 0.1

test <- mcpmod_simulation(doses = doses,
           parm = parm,
           sample.size = sample.size,
           model.true = model.true,
           models.candidate = models.candidate,
           selModel = selModel,
           significance.level = significance.level,
           Delta = Delta,
           distr = "weibull",
           censoring.rate = censoring.rate,
           sigma.estimator = "jackknife",
           n.cores = n.cores, seed = seed, n.sim = n.sim)
summary(test)
```

`summary.weibreg` *Print a summary for a object of the* `weibreg` *class.*

Description

Summarizes the results for a object of the `weibreg` class.

Usage

```
## S3 method for class 'weibreg'
summary(object, ...)

## S3 method for class 'weibreg'
print(x, digits = max(3L, getOption("digits") - 3L), ...)
```

Arguments

`object`	an object of the `weibreg` class.
`...`	additional arguments affecting the summary produced.
`x`	an object of class `summary.weibreg`, usually, a result of a call to `summary.weibreg`.
`digits`	the number of significant digits to use when printing.

Value

A complete summary for the coefficients extracted from a `weibreg` object.

Functions

- `print(weibreg)`:

References

Diniz, Márcio A. and Gallardo, Diego I. and Magalhães, Tiago M. (2023). Improved inference for MCP-Mod approach for time-to-event endpoints with small sample sizes. arXiv <doi.org/10.48550/arXiv.2301.00325>

Examples

```
require(survival)
set.seed(2100)

##Generating covariates
n=20
x<-runif(n, max=10)
lambda<-exp(1.2-0.5*x)
sigma<-1.5

##Drawing T from Weibull model and fixing censoring at 1.5
T<-rweibull(n, shape=1/sigma, scale=lambda)
L<-rep(1.5, n)

##Defining the observed times and indicators of failure
t<-pmin(T,L)
delta<-ifelse(T<=L, 1, 0)
data=data.frame(t=t, delta=delta, x=x)

##Fitting for Weibull regression model

##Traditional MLE with corrected variance
ex1=weibfit(Surv(t,delta)~x, data=data, L=L, estimator="MLE",
corrected.var=TRUE)
summary(ex1)

##BCE without corrected variance
ex2=weibfit(Surv(t,delta)~x, data=data, L=L, estimator="Firth",
corrected.var=FALSE)
summary(ex2)

##BCE with corrected variance
ex3=weibfit(Surv(t,delta)~x, data=data, L=L, estimator="BCE",
corrected.var=TRUE)
summary(ex3)

##Firth's correction without corrected variance
ex4=weibfit(Surv(t,delta)~x, data=data, L=L, estimator="BCE",
corrected.var=FALSE)
```

```
summary(ex4)
```

`weibfit` *Computes different estimators for the censored Weibull regression model*

Description

Computes the maximum likelihood estimators (MLE) for the censored Weibull regression model. The bias-corrected estimators based on the Cox and Snell's and Firth's methods also are available. In addition, for the covariance matrix the corrected estimators discussed in Magalhaes et al. 2021 also are available.

Usage

```
weibfit(formula, data, L = Inf, estimator = "MLE",
corrected.var = FALSE)
```

Arguments

`formula`	A formula that contains on the left hand side an object of the type Surv and on the right hand side the covariates definition
`data`	A data.frame in which the formula argument can be evaluated
`L`	the prefixed censoring times. $L = \infty$ by default.
`estimator`	the class of estimator used: MLE (maximum likelihood estimator, by default), BCE (bias-corrected estimator based on the Cox and Snell's method) and Firth (bias-corrected estimator based on the Firth's method).
`corrected.var`	should the covariance-corrected estimator be used? (FALSE by default). See details.

Details

The Weibull distribution considered here has probability density function

$$f(t;\lambda,\sigma) = \frac{1}{\sigma\lambda^{1/\sigma}} t^{1/\sigma-1} \exp\left\{-\left(\frac{t}{\lambda}\right)^{1/\sigma}\right\}, \quad t,\sigma,\lambda > 0.$$

The regression structure is incorporated as

$$\log(\lambda_i) = \boldsymbol{x}_i^\top \boldsymbol{\beta}, \quad i = 1,\ldots,n.$$

For the computation of the bias-corrected estimators, σ is assumed as fixed in the jackknife estimator based on the traditional MLE.

The Fisher information matrix for $\boldsymbol{\beta}$ is given by $\boldsymbol{K} = \sigma^{-2}\boldsymbol{X}^\top \boldsymbol{W}\boldsymbol{X}$, where $\boldsymbol{X} = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n)^\top$, $\boldsymbol{W} = \text{diag}(w_1, \ldots, w_n)$, and

$$w_i = E\left[\exp\left(\frac{y_i - \log\lambda_i}{\sigma}\right)\right] = q \times \left\{1 - \exp\left[-L_i^{1/\sigma}\exp(-\mu_i/\sigma)\right]\right\} + (1-q)\times(r/n)\,,$$

with $q = P\left(W_{(r)} \leq \log L_i\right)$ and $W_{(r)}$ denoting the rth order statistic from $W_1, \ldots, W_n$, with $q = 1$ and $q = 0$ for types I and II censoring, respectively. (See Magalhaes et al. 2019 for details).

The bias-corrected maximum likelihood estimator based on the Cox and Snell's method (say $\widetilde{\boldsymbol{\beta}}$) is based on a corrective approach given by $\widetilde{\beta} = \widehat{\beta} - B(\widehat{\beta})$, where

$$B(\boldsymbol{\beta}) = -\frac{1}{2\sigma^3}\boldsymbol{P}\boldsymbol{Z}_d\left(\boldsymbol{W} + 2\sigma\boldsymbol{W}'\right)\mathbf{1},$$

with $\boldsymbol{P} = \boldsymbol{K}^{-1}\boldsymbol{X}^\top$, $\boldsymbol{Z} = \boldsymbol{X}\boldsymbol{K}^{-1}\boldsymbol{X}^\top$, $\boldsymbol{Z}_d$ is a diagonal matrix with diagonal given by the diagonal of $\boldsymbol{Z}$, $\boldsymbol{W}' = \text{diag}(w_1', \ldots, w_n')$, $w_i' = -\sigma^{-1}L_i^{1/\sigma}\exp\{-L_i^{1/\sigma}\exp(-\mu_i/\sigma) - \mu_i/\sigma\}$ and $\mathbf{1}$ is a n-dimensional vector of ones.

The bias-corrected maximum likelihood estimator based on the Firth's method (say $\check{\boldsymbol{\beta}}$) is based on a preventive approach, which is the solution for the equation $\boldsymbol{U}_{\boldsymbol{\beta}}^\star = \mathbf{0}$, where

$$\boldsymbol{U}_{\boldsymbol{\beta}}^\star = \boldsymbol{U}_{\boldsymbol{\beta}} - \boldsymbol{K}_{\boldsymbol{\beta}\boldsymbol{\beta}}B(\boldsymbol{\beta}).$$

The covariance correction is based on the general result of Magalhaes et al. 2021 given by

$$\mathbf{Cov}_{\mathbf{2}}^{\boldsymbol{\tau}}(\boldsymbol{\beta}^\star) = \boldsymbol{K}^{-1} + \boldsymbol{K}^{-1}\left\{\boldsymbol{\Delta} + \boldsymbol{\Delta}^\top\right\}\boldsymbol{K}^{-1} + \mathcal{O}(n^{-3})$$

where $\Delta = -0.5\Delta^{(1)} + 0.25\Delta^{(2)} + 0.5\tau_2\Delta^{(3)}$, with

$$\Delta^{(1)} = \frac{1}{\sigma^4}\boldsymbol{X}^\top\boldsymbol{W}^\star\boldsymbol{Z}_d\boldsymbol{X},$$

$$\Delta^{(2)} = -\frac{1}{\sigma^6}\boldsymbol{X}^\top\left[\boldsymbol{W}\boldsymbol{Z}^{(2)}\boldsymbol{W} - 2\sigma\boldsymbol{W}\boldsymbol{Z}^{(2)}\boldsymbol{W}' - 6\sigma^2\boldsymbol{W}'\boldsymbol{Z}^{(2)}\boldsymbol{W}'\right]\boldsymbol{X},$$

and

$$\Delta^{(3)} = \frac{1}{\sigma^5}\boldsymbol{X}^\top\boldsymbol{W}'\boldsymbol{W}^{\star\star}\boldsymbol{X},$$

where $\boldsymbol{W}^\star = \text{diag}(w_1^\star, \ldots, w_n^\star)$, $w_i^\star = w_i(w_i - 2) - 2\sigma w_i' + \sigma\tau_1(w_i' + 2\sigma w_i'')$, $\boldsymbol{Z}^{(2)} = \boldsymbol{Z}\odot\boldsymbol{Z}$, with $\odot$ representing a direct product of matrices (Hadamard product), $\boldsymbol{W}^{\star\star}$ is a diagonal matrix, with $\boldsymbol{Z}(\boldsymbol{W} + 2\sigma\boldsymbol{W}')\boldsymbol{Z}_d\mathbf{1}$ as its diagonal, $\boldsymbol{W}'' = \text{diag}(w_1'', \ldots, w_n'')$, $w_i'' = -\sigma^{-1}w_i'\left[L_i^{1/\sigma}\exp(-\mu_i/\sigma) - 1\right]$, $\boldsymbol{\tau} = (\tau_1, \tau_2) = (1, 1)$ indicating the second-order covariance matrix of the MLE $\boldsymbol{\beta}^\star = \widehat{\boldsymbol{\beta}}$ denoted by $\mathbf{Cov_2}(\widehat{\boldsymbol{\beta}})$ and $\boldsymbol{\tau} = (0, -1)$ indicating the second-order covariance matrix of the BCE $\boldsymbol{\beta}^\star = \widetilde{\boldsymbol{\beta}}$ denoted by $\mathbf{Cov_2}(\widetilde{\boldsymbol{\beta}})$.

Value

`coefficients`	a vector with the estimated coefficients for $\boldsymbol{\beta}$.
`var`	a matrix with the estimated covariance matrix for the estimates of the regression coefficients $\boldsymbol{\beta}$

scale	the estimated scale parameter σ
loglik	the value for the logarithm of the likelihood function evaluated in the estimates of $\boldsymbol{\beta}$ and σ
linear.predictors	a vector with the estimated linear predictor $\boldsymbol{x}_i^\top \boldsymbol{\beta}$
y	a vector with the observed times (possibly censored)
estimator	the estimator used for $\boldsymbol{\beta}$: MLE, BCE or Firth.
corrected.var	logical. TRUE if a correction for the covariance was used, FALSE otherwise.

Author(s)

Gallardo D.I., Diniz, M.A., Magalhaes, T.M.

References

Cox, D.R., Snell E.J. A general definition of residuals Journal of the Royal Statistical Society. Series B (Methodological). 1968;30:248-275.

Diniz, Márcio A. and Gallardo, Diego I. and Magalhães, Tiago M. (2023). Improved inference for MCP-Mod approach for time-to-event endpoints with small sample sizes. arXiv <doi.org/10.48550/arXiv.2301.00325>

Firth, D. Bias reduction of maximum likelihood estimates Biometrika. 1993;80:27-38.

Magalhaes Tiago M., Botter Denise A., Sandoval Monica C. A general expression for second-order covariance matrices - an application to dispersion models Brazilian Journal of Probability and Statistics. 2021;35:37-49.

Examples

```
require(survival)
set.seed(2100)

##Generating covariates
n=20;
x<-runif(n, max=10)
lambda<-exp(1.2-0.5*x); sigma<-1.5

##Drawing T from Weibull model and fixing censoring at 1.5
T<-rweibull(n, shape=1/sigma, scale=lambda); L<-rep(1.5, n)

##Defining the observed times and indicators of failure
y<-pmin(T,L);
delta<-ifelse(T<=L, 1, 0)
data=data.frame(y=y, delta=delta, x=x)

##Fitting for Weibull regression model

##Traditional MLE with corrected variance
ex1=weibfit(Surv(y,delta)~x, data=data, L=L, estimator="MLE",
corrected.var=TRUE)
summary(ex1)
```

```
##BCE without corrected variance
ex2=weibfit(Surv(y,delta)~x, data=data, L=L, estimator="BCE",
corrected.var=FALSE)
summary(ex2)

##BCE with corrected variance
ex3=weibfit(Surv(y,delta)~x, data=data, L=L, estimator="BCE",
corrected.var=TRUE)
summary(ex3)

##Firth's correction without corrected variance
ex4=weibfit(Surv(y,delta)~x, data=data, L=L, estimator="BCE",
corrected.var=FALSE)
summary(ex4)
```

Index