Package ‘sclr’

October 14, 2022

Title Scaled Logistic Regression

Version 0.3.1

Description Maximum likelihood estimation of the scaled logit model parameters proposed in Dunning (2006) <doi:10.1002/sim.2282>.

License MIT + file LICENSE

URL `https://khvorov45.github.io/sclr/`

Encoding UTF-8

LazyData true

RoxygenNote 7.0.2

Depends R (>= 3.6.0)

Imports broom, tibble, dplyr, rlang, stats, purrr

Suggests knitr, rmarkdown, testthat (>= 2.1.0)

VignetteBuilder knitr

NeedsCompilation no

Author Arseniy Khvorov [aut, cre]

Maintainer Arseniy Khvorov <`khvorov45@gmail.com`>

Repository CRAN

Date/Publication 2020-03-02 12:30:02 UTC

R topics documented:

`check_baseline` *Check for baseline boundary*

Description

Fits the scaled logit model as well as logistic regression. Does a likelihood ratio test.

Usage

```
check_baseline(
  formula = NULL,
  data = NULL,
  fit_sclr = NULL,
  fit_lr = NULL,
  conf_lvl = 0.95,
  verbose = TRUE
)
```

Arguments

`formula`	Formula to use for model fitting.
`data`	Optional dataframe.
`fit_sclr`	Fit object returned by `sclr`.
`fit_lr`	Fit object returned by `glm`.
`conf_lvl`	Confidence level for the test
`verbose`	Whether to print message based on test result.

Value

A `tibble` with a summary.

Examples

```
library(sclr)
l1 <- sclr_ideal_data(n = 50, theta = 1e6, seed = 20191104)
check_baseline(status ~ logHI, l1)
```

coef.sclr *ML estimate components*

Description

coef returns MLE's. vcov returns the estimated variance-covariance matrix at MLE's. confint returns the confidence interval. model.matrix returns the model matrix (x). model.frame returns the model frame (x and y in one matrix).

Usage

```
## S3 method for class 'sclr'
coef(object, ...)

## S3 method for class 'sclr'
vcov(object, ...)

## S3 method for class 'sclr'
confint(object, parm, level = 0.95, ...)

## S3 method for class 'sclr'
model.matrix(object, ...)

## S3 method for class 'sclr'
model.frame(formula, ...)

## S3 method for class 'sclr'
logLik(object, ...)
```

Arguments

object, formula	An object returned by sclr.
...	Not used. Needed to match generic signature.
parm	Parameter name, if missing, all parameters are considered.
level	Confidence level.

find_prot_titre_val *Search function for scaled logit protection covariate levels*

Description

The search engine behind get_protection_level. Should not usually be necessary to call this directly.

Usage

```
find_prot_titre_val(
  fit,
  var_name,
  newdata = NULL,
  prot_var_name = "prot_point",
  lvl = 0.5,
  ci_level = 0.95,
  tol = 10^(-7)
)
```

Arguments

`fit` Object returned by `sclr`.

`var_name` Name of the covariate for which the protection values should be calculated. This name should appear in the formula of the call to `sclr` which was used to generate `fit`.

`newdata` A dataframe with all covariates except the one for which protection values should be calculated.

`prot_var_name` A variable name among those returned by `predict.sclr` which needs to equal `lvl` at the value of `var_name` that is supposed to be found.

`lvl` Protection level to find titre values for. Default is 0.5 (50%).

`ci_level` Confidence level for the calculated interval. Default is 0.95.

`tol` Tolerance. The values will be found numerically, once the algorithm converges within `tol` of `lvl` it stops looking. Default is $10^{(} - 7)$.

Value

A dataframe. Will have the same variables as `newdata` with the addition of the `var_name` variable.

`get_protection_level` *Protection level calculations*

Description

Calculates covariate values corresponding to a particular protection level. Only accepts one covariate at a time, fixed values of all the others should be provided. The search engine is `find_prot_titre_val`.

Usage

```
get_protection_level(
  fit,
  var_name,
  newdata = NULL,
  lvl = 0.5,
  ci_level = 0.95,
  tol = 10^(-7)
)
```

Arguments

`fit` Object returned by `sclr`.

`var_name` Name of the covariate for which to find values corresponding to a protection level. This name should appear in the formula in the call to `sclr` which was used to generate `fit`.

`newdata` A dataframe with all covariates except the one for which protection values should be calculated. If there is only one covariate, can be left as `NULL` (the default)

`lvl` Protection level to find covariate values for. Default is 0.5 (50%)

`ci_level` Confidence level for the calculated interval. Default is 0.95.

`tol` Tolerance. The values will be found numerically, once the algorithm converges within `tol` of `lvl` it stops looking. Default is $10^(- 7)$.

Value

A `tibble`. Will have the same variables as `newdata` with the addition of the `var_name` variable.

`new_sclr` *Create a new* `sclr` *object*

Description

`new_sclr` creates the object `sclr` returns. `is_sclr` checks if the object is of class `sclr`.

Usage

```
new_sclr(fit, x, y, cl, mf, mt)

is_sclr(fit)
```

Arguments

`fit` A list returned by `sclr_fit`.

`x` Model matrix.

`y` Model response.

`cl` Call.

`mf` Model frame.

`mt` Model terms.

Value

`sclr` object

`one_titre_data` *Simulated one-titre antibody data*

Description

A simulated dataset containing 5000 independent observations on antibody titres and the corresponding infection status. The data was simulated to resemble real influenza infection and haemagglutinin titre data.

Usage

```
one_titre_data
```

Format

A data frame with 5000 observations and 2 variables:

logHI haemagglutinin-inhibiting (HI) titre. True simulated titre on a log scale.

status influenza infection status. 1 - infected. 0 - not infected

Model

The model behind the simulation was

$$\lambda * (1 - f(\beta_0 + \beta_1 * HI))$$

Where

- f - Inverse logit function
- λ = 0.5
- β_0 = -5
- β_1 = 2

`predict.sclr` *Predict method for scaled logit model x.*

Description

Returns only the protection estimates. The only supported interval is a confidence interval (i.e. the interval for the estimated expected value).

Usage

```
## S3 method for class 'sclr'
predict(object, newdata, ci_lvl = 0.95, ...)
```

Arguments

`object` Object returned by `sclr`.

`newdata` A dataframe with all covariates. Names should be as they appear in the formula in the call to `sclr`.

`ci_lvl` Confidence level for the calculated interval.

`...` Not used. Needed to match generic signature.

Details

The model is

$$P(Y = 1) = \lambda(1 - logit^{-1}(\beta_0 + \beta_1 X_1 + \beta_2 X_2 + ... + \beta_k X_k))$$

Where Y is the binary outcome indicator, (e.g. 1 - infected, 0 - not infected). X - covariate. k - number of covariates. This function calculates

$$\beta_0 + \beta_1 X_1 + \beta_2 X_2 + ... + \beta_k X_k$$

transformations at the covariate values found in `newdata` as well as the variance-covariance matrices of those transformations. This is used to calculate the confidence intervals at the given parameter values. The inverse logit transformation is then applied to point estimates and interval bounds.

Value

A `tibble` obtained by adding the following columns to `newdata`:

`prot_point_lin` `prot_l_lin` `prot_u_lin`
Point estimate, low and high bounds of the linear transformation.

`prot_sd_lin` Estimated standard deviation of the linear transformation.

`prot_point` `prot_l` `prot_u`
Inverse logit-transformed point estimate, low and high bounds of the linear transformation.

`print.sclr` *Print a* `sclr` *object.*

Description

Summarises a `sclr` object for printing. For a dataframe summary, see `tidy`.

Usage

```
## S3 method for class 'sclr'
print(x, level = 0.95, ...)

## S3 method for class 'sclr'
summary(object, level = 0.95, ...)
```

Arguments

`x, object` An object returned by `sclr`.

`level` Confidence level for the intervals.

`...` Not used. Needed to match generic signature.

`sclr` *Fits the scaled logit model*

Description

Used to fit the scaled logit model from Dunning (2006).

Usage

```
sclr(
  formula,
  data = NULL,
  ci_lvl = 0.95,
  tol = 10^(-7),
  algorithm = c("newton-raphson", "gradient-ascent"),
  nr_iter = 2000,
  ga_iter = 2000,
  n_conv = 3,
  conventional_names = FALSE,
  seed = NULL
)
```

Arguments

`formula` an object of class "formula": a symbolic description of the model to be fitted.

`data` a data frame.

`ci_lvl` Confidence interval level for the parameter estimates.

`tol` Tolerance.

`algorithm` Algorithms to run. "newton-raphson" or "gradient-ascent". If a character vector, the algorithms will be applied in the order they are present in the vector.

`nr_iter` Maximum allowed iterations for Newton-Raphson.

`ga_iter` Maximum allowed iterations for gradient ascent.

`n_conv` Number of times the algorithm has to converge (to work around local maxima).

`conventional_names`
If `TRUE`, estimated parameter names will be (Baseline), (Intercept) and the column names in the model matrix. Otherwise - lambda, beta_0 and beta_ prefix in front of column names in the model matrix.

`seed` Seed for the algorithms.

Details

The model is logistic regression with an added parameter for the top asymptote. That parameter is reported as `theta` (or `(Baseline)` if `conventional_names = TRUE`). Note that it is reported on the logit scale. See `vignette("sclr-math")` for model specification, log-likelihood, scores and second derivatives. The main default optimisation algorithm is Newton-Raphson. Gradient ascent is used as a fallback by default. Computing engine behind the fitting is `sclr_fit`.

Value

An object of class `sclr`. This is a list with the following elements:

`parameters` Maximum likelihood estimates of the parameter values.

`covariance_mat` The variance-covariance matrix of the parameter estimates.

`algorithm` Algorithm used.

`algorithm_return` Everything the algorithm returned.

`n_converge` The number of Newton-Raphson iterations (including resets) that were required for convergence.

`x` Model matrix derived from `formula` and `data`.

`y` Response matrix derived from `formula` and `data`.

`call` The original call to `sclr`.

`model` Model frame object derived from `formula` and `data`.

`terms` Terms object derived from model frame.

`ci` Confidence intervals of the parameter estimates.

`log_likelihood` Value of log-likelihood calculated at the ML estimates of parameters.

`formula` Passed formula.

`data` Passed data.

Methods supported: `print`, `vcov`, `coef`, `model.frame`, `model.matrix`, `summary`, `predict`, `tidy` (`broom` package), `logLik`.

References

Dunning AJ (2006). "A model for immunological correlates of protection." Statistics in Medicine, 25(9), 1485-1497. `https://doi.org/10.1002/sim.2282`.

Examples

```
library(sclr)
fit1 <- sclr(status ~ logHI, one_titre_data)
summary(fit1)
```

sclr_fit *Fitter function for the scaled logit model*

Description

Computing engine behind sclr.

Usage

```
sclr_fit(
  y,
  x,
  tol = 10^(-7),
  algorithm = c("newton-raphson", "gradient-ascent"),
  nr_iter = 2000,
  ga_iter = 2000,
  n_conv = 3,
  conventional_names = FALSE,
  seed = NULL
)
```

Arguments

`y` A vector of observations.

`x` A design matrix.

`tol` Tolerance.

`algorithm` Algorithms to run. "newton-raphson" or "gradient-ascent". If a character vector, the algorithms will be applied in the order they are present in the vector.

`nr_iter` Maximum allowed iterations for Newton-Raphson.

`ga_iter` Maximum allowed iterations for gradient ascent.

`n_conv` Number of times the algorithm has to converge (to work around local maxima).

`conventional_names`
If `TRUE`, estimated parameter names will be (Baseline), (Intercept) and the column names in the model matrix. Otherwise - lambda, beta_0 and beta_ prefix in front of column names in the model matrix.

`seed` Seed for the algorithms.

Details

The likelihood maximisation can use the Newton-Raphson or the gradient ascent algorithms.

sclr_ideal_data *Generate ideal data for the scaled logit model*

Description

Allows variation of all parameters and the creation of an arbitrary number of covariates.

Usage

```
sclr_ideal_data(
  n = 1000,
  theta = 0,
  beta_0 = -5,
 covariate_list = list(logHI = list(gen_fun = function(n) rnorm(n, 2, 2), true_par =
   2)),
  outcome_name = "status",
  seed = NULL,
  attach_true_vals = FALSE,
  attach_seed = FALSE
)
```

Arguments

`n` Number of observations.

`theta` Baseline risk parameter on the logit scale.

`beta_0` Intercept of the linear part.

`covariate_list` A list in the form of `name = list(gen_fun,true_par)` where `gen_fun` is a function that takes `n` as an argument and returns a vector of observations, `true_par` is the true parameter value of that covariate. See examples.

`outcome_name` Name to give to the outcome

`seed` Seed to set. If `NULL`, no seed will be set.

`attach_true_vals, attach_seed`
Whether to attach additional attributes.

Value

A `tibble`.

Examples

```
# One titre
one_titre <- sclr_ideal_data(
  covariate_list = list(
    logHI = list(gen_fun = function(n) rnorm(n, 2, 2), true_par = 2)
  )
)
```

```
sclr(status ~ logHI, one_titre) # Verify

# Two titres
two_titre <- sclr_ideal_data(
  covariate_list = list(
    logHI = list(gen_fun = function(n) rnorm(n, 2, 2), true_par = 2),
    logNI = list(gen_fun = function(n) rnorm(n, 2, 2), true_par = 1)
  )
)
sclr(status ~ logHI + logNI, two_titre) # Verify
```

`sclr_log_likelihood` *Log-likelihood*

Description

Computes the log-likelihood of the scaled logit model at a given set of parameter estimates (or the MLE if `pars` is not supplied). Either `fit` or `x`, `y` and `pars` need to be supplied.

Usage

```
sclr_log_likelihood(fit = NULL, x = NULL, y = NULL, pars = NULL)
```

Arguments

`fit` An object returned by `sclr`. Or a list with parameters, x and y entries corresponding to the parameter matrix, model matrix and model response.

`x` Model matrix. Will be taken from `fit` if `fit` is provided.

`y` Model response. Will be taken from `fit` if `fit` is provided.

`pars` A named vector of parameter values. Will be taken from `fit` if `fit` is provided.

`tidy.sclr` *Tidy a* `sclr` *object.*

Description

Summarises the objects returned by `sclr` into a `tibble`.

Usage

```
## S3 method for class 'sclr'
tidy(x, ci_level = 0.95, ...)
```

Arguments

`x`	An object returned by `sclr`.
`ci_level`	Confidence level for the intervals.
`...`	Not used. Needed to match generic signature.

Value

A `tibble` with one row per model parameter. Columns:

`term`	Name of model parameter.
`estimate`	Point estimate.
`std_error`	Standard error.
`conf_low`	Lower bound of the confidence interval.
`conf_high`	Upper bound of the confidence interval.

`two_titre_data` *Simulated two-titre antibody data*

Description

A simulated dataset containing 5000 independent observations on antibody titres and the corresponding infection status. The data was simulated to resemble real influenza infection and haemagglutinin + neuraminidase titre data.

Usage

```
two_titre_data
```

Format

A data frame with 5000 observations and 3 variables:

logHI haemagglutinin-inhibiting (HI) titre. True simulated titre on a log scale.

logNI neuraminidase-inhibiting titre. True simulated titre on a log scale.

status influenza infection status. 1 - infected. 0 - not infected

Model

The model behind the simulation was

$$\lambda * (1 - f(\beta_0 + \beta_1 * HI + \beta_2 * NI))$$

Where

- f - Inverse logit function

- λ = 0.5
- β_0 = -7.5
- β_1 = 2
- β_2 = 2

Index