Package ‘addhaz’

May 7, 2026

Title Binomial and Multinomial Additive Hazard Models

Version 0.5

Description Functions to fit the binomial and multinomial additive hazard models and to estimate the contribution of diseases/conditions to the disability prevalence, as proposed by Nusselder and Looman (2004) and extended by Yokota et al (2017).

Depends R (>= 3.1.0), stats, boot (>= 1.3-17), Matrix (>= 1.2-3), MASS

License GPL-3

Encoding UTF-8

LazyData true

RoxygenNote 6.1.0

NeedsCompilation no

Author Renata T C Yokota [cre, aut],
Caspar W N Looman [aut],
Wilma J Nusselder [aut],
Herman Van Oyen [aut],
Geert Molenberghs [aut]

Maintainer Renata T C Yokota <`renatayokota@gmail.com`>

Repository CRAN

Date/Publication 2018-09-25 22:40:03 UTC

Contents

`BinAddHaz` *Fit Binomial Additive Hazard Models*

Description

This function fits binomial additive hazard models subject to linear inequality constraints using the function `constrOptim` in the `stats` package for binary outcomes. Additionally, it calculates the cause-specific contributions to the disability prevalence based on the attribution method, as proposed by Nusselder and Looman (2004).

Usage

```
BinAddHaz(formula, data, subset, weights, na.action, model = TRUE,
          contrasts = NULL, start, attrib = TRUE,
          attrib.var, collapse.background = FALSE, attrib.disease = FALSE,
          type.attrib = "abs", seed, bootstrap = FALSE, conf.level = 0.95,
          nbootstrap, parallel = FALSE, type.parallel = "snow", ncpus = 4,...)
```

Arguments

`formula`	a formula expression of the form response ~ predictors, similar to other regression models. In case of `attrib = TRUE`, the first predictor in the formula should be the `attrib.var`. See example.
`data`	an optional data frame or matrix containing the variables in the model. If not found in `data`, the variables are taken from environment(formula), typically the environment from which `BinAddHaz` is called.
`subset`	an optional vector specifying a subset of observations to be used in the fitting process. All observations are included by default.
`weights`	an optional vector of survey weights to be used in the fitting process.
`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. The 'factory-fresh' default is `na.omit`.
`model`	logical. If `TRUE`, the model frame is included as a component of the returned object.
`contrasts`	an optional list to be used for some or all of the factors appearing as variables in the model formula.
`start`	an optional vector of starting values. If not provided by the user, it is automatically generated using `glm`, `family = poisson`.
`attrib`	logical. Should the attribution of disability to chronic diseases/conditions be estimated? Default is `TRUE`.
`attrib.var`	character indicating the name of the attribution variable to be used if `attrib = TRUE`. If missing, the attribution results will not be stratified by the levels of the attribution variable. The attribution variable must be the first variable included in the linear predictor in `formula`. See example.

`collapse.background`	logical. Should the background be collapsed across the levels of the `attrib.var`? If `FALSE`, the background will be estimated for each level of the `attrib.var`. If `TRUE`, only one background will be estimated. If `TRUE`, `attrib.var` must be specified. Default is `FALSE`.
`attrib.disease`	logical. Should the attribution of diseases be stratified by the levels of the attribution variable? If `FALSE`, the attribution of diseases will not be stratified by the levels of the `attrib.var`. If `TRUE`, the attribution of diseases will be estimated for each level of the `attrib.var`. If `TRUE`, interaction between diseases and the attribution variable must be provided in the `formula`. Default is `FALSE`.
`type.attrib`	type of attribution to be estimated. The options are `"abs"` for absolute contribution, `"rel"` for relative contribution, or `"both"` for both absolute and relative contributions. Default is `"abs"`.
`seed`	an optional integer indicating the random seed.
`bootstrap`	logical. Should bootstrap percentile confidence intervals be estimated for the model parameters and attributions? Default is `FALSE`. See details.
`conf.level`	scalar containing the confidence level of the bootstrap percentile confidence intervals. Default is `0.95`.
`nbootstrap`	integer. Number of bootstrap replicates.
`parallel`	logical. Should parallel calculations be used to obtain the bootstrap percentile confidence intervals? Only valid if `bootstrap = TRUE`. Default is `FALSE`.
`type.parallel`	type of parallel operation to be used (if `parallel = TRUE`), with options: `"multicore"` and `"snow"`. Default is `"snow"`. See details.
`ncpus`	integer. Number of processes to be used in the parallel operation: typically one would choose this to be the number of available CPUs. Default is 4.
`...`	other arguments passed to or from the other functions.

Details

The model is a generalized linear model with a non-canonical link function, which requires a restriction on the linear predictor ($\eta \geq 0$) to produce valid probabilities. This restriction is implemented in the optimization procedure, with an adaptive barrier algorithm, using the function `constrOptim` in the `stats` package.

The variance-covariance matrix is based on the observed information matrix.

This version of the package only allows the calculation of non-parametric bootstrap percentile confidence intervals (CI). Also, the function gives the user the option to do parallel calculation of the bootstrap CI. The `snow` parallel option is available for all operating systems (Windows, Linux, and Mac OS) while the `multicore` option is only available for Linux and Mac OS systems. These two calculations are done by calling the `boot` function in the `boot` package. For more details, see the documentation of the `boot` package.

More information about the binomial additive hazard model and the calculation of the contribution of chronic conditions to the disability prevalence can be found in the references.

Value

A list with arguments:

`coefficients`	numerical vector with the regression coefficients.
`ci`	confidence intervals calculated via bootstraping (if `bootstrap = TRUE`) or as the inverse of the observed information matrix.
`resDeviance`	residual deviance.
`df`	degrees of freedom.
`pvalue`	numerical vector of p-values based on the Wald test. Only provided if `bootstrap = FALSE`.
`stdError`	numerical vector with the standard errors for the parameter estimates based on the inverse of the observed information matrix. Only provided if `bootstrap = FALSE`.
`vcov`	variance-covariance (inverse of the observed information matrix). Only provided if `bootstrap = FALSE`.
`contribution`	for `type.attrib = "abs"` or `"rel"`, a matrix is provided. For `type.attrib = "both"`, a list with two matrices (`"abs"` and `"rel"`) is provided. This represents the contribution of each predictor in the model (usually diseases) to the disability prevalence. Percentile bootstrap confidence intervals are provided if `bootstrap = TRUE`.
`bootsRep`	matrix with the bootstrap replicates of the regression coefficients and contributions (if `attrib = TRUE`). If `attrib = FALSE`, it returns a logical, `FALSE`.
`conf.level`	confidence level of the bootstrap CI. Only provided if `bootstrap = TRUE`.
`bootstrap`	logical. Was bootstrap CI requested?
`fitted.values`	the fitted mean values, obtained by transforming the linear predictor by the inverse of the link function.
`residuals`	difference between the observed response and the fitted values.
`call`	the matched call.

Author(s)

Renata T C Yokota. This function is based on the R code developed by Caspar W N Looman and Wilma J Nusselder for non R-users, with modifications. Original R code is available upon request to Wilma J Nusselder (w.nusselder@erasmusmc.nl).

References

Nusselder, W.J., Looman, C.W.N. (2004). Decomposition of differences in health expectancy by cause. Demography, 41(2), 315-334.

Nusselder, W.J., Looman, C.W.N. (2010). WP7: Decomposition tools: technical report on attribution tool. European Health Expectancy Monitoring Unit (EHEMU). Available at <http://www.eurohex.eu/pdf/Reports_2010/2010TR7.2_TR%20on%20attribution%20tool.pdf>.

Yokota, R.T.C., Van Oyen, H., Looman, C.W.N., Nusselder, W.J., Otava, M., Kifle, Y.W., Molenberghs, G. (2017). Multinomial additive hazard model to assess the disability burden using cross-sectional data. Biometrical Journal, 59(5), 901-917.

See Also

`MultAddHaz`

Examples

```
  data(disabData)

  ## Model without bootstrap CI and no attribution

  fit1 <- BinAddHaz(dis.bin ~ diab + arth + stro , data = disabData, weights = wgt,
                    attrib = FALSE)
  summary(fit1)

 ## Model with bootstrap CI and attribution without stratification, no parallel calculation
  # Warning message due to the low number of bootstrap replicates
## Not run:
  fit2 <- BinAddHaz(dis.bin ~ diab + arth + stro , data = disabData, weights = wgt,
                    attrib = TRUE, collapse.background = FALSE, attrib.disease = FALSE,
                   type.attrib = "both", seed = 111, bootstrap = TRUE, conf.level = 0.95,
                    nbootstrap = 5)
  summary(fit2)

  ## Model with bootstrap CI and attribution of diseases and background stratified by
  ## age, with parallel calculation of bootstrap CI
  # Warning message due to the low number of bootstrap replicates

  diseases <- as.matrix(disabData[,c("diab", "arth", "stro")])
  fit3 <- BinAddHaz(dis.bin ~ factor(age) -1 + diseases:factor(age), data = disabData,
                    weights = wgt, attrib = TRUE, attrib.var = age,
                 collapse.background = FALSE, attrib.disease = TRUE, type.attrib = "both",
                    seed = 111,  bootstrap = TRUE, conf.level = 0.95, nbootstrap = 10,
                    parallel = TRUE, type.parallel = "snow", ncpus = 4)
  summary(fit3)
## End(Not run)
```

---

disabData *Example of disability data*

Description

The disabData is a subset of the data from the 2013 National Health Survey in Brazil ("Pesquisa Nacional de Saude, 2013"). The data are restricted to women aged 60 years or older, resulting in 6294 individuals.

Usage

```
data(disabData)
```

Format

This dataset has information about disability and chronic conditions. The disability outcomes were defined as limitations on instrumental activities of daily living (IADL). Individuals with missing data were excluded. The data frame contains 7 variables:

- **dis.bin:** disability as a binary variable, with 2 categories: 0 (no disability), 1 (disability).
- **dis.mult:** disability as a multinomial variable, with 3 categories: 0 (no disability), 1 (mild disability), and 2 (severe disability).
- **wgt:** survey weights.
- **age:** binary variable for age: 0 (60-79 years) or 1 (80+ years).
- **diab:** binary variable for diabetes: 0 (no) or 1 (yes).
- **arth:** binary variable for arthritis: 0 (no) or 1 (yes).
- **stro:** binary variable for stroke: 0 (no) or 1 (yes).

Source

The data were obtained from the National Health Survey 2013, Brazil. For more information about the survey, see references.

References

Szwarcwald, C.L., Malta, D.C., Pereira, C.A., Vieira, M.L., Conde, W.L., Souza Junior, P.R., et al. (2013). National Health Survey in Brazil: design and methodology of application. Cien Saude Colet., 19(2): 333:42 [Article in Portuguese].

Instituto Brasileiro de Geografia e Estatistica (IBGE). Pesquisa Nacional de Saude 2013. Available at <`http://www.ibge.gov.br/home/estatistica/populacao/pns/2013/`>.

Examples

```
data(disabData)
str(disabData)
```

`LRTest` *Perform likelihood ratio test*

Description

This function performs the likelihood ratio test to compare two nested binomial or multinomial additive hazard models. It can be used for model selection.

Usage

```
LRTest(model1, model2)
```

Arguments

`model1, model2`	objects of class "binaddhazmod" or "multaddhazmod" to be compared. See example.

Details

The likelihood ratio test is defined as -2*log(likelihood model 1/likelihood model 2).The resulting test statistic is assumed to follow a chi-squared distribution, with degrees of freedom (df) equal to the difference of the df between the models. If the test is statistically significant, the model with more variables fits the data significantly better than the model with less variables.

Value

A data frame with columns:

`Res.df`	degrees of freedom for each model.
`Res.dev`	2*log-likelihood of each model.
`df`	difference in the degrees of freedom between models 1 and 2.
`Deviance`	difference between the 2*log-likelihood of models 1 and 2, representing the value of the likelihood ratio test statistic.
`Pr(>Chi)`	p-value, based on the chi-squared distribution.

Examples

```
  data(disabData)

  ## Comparing two binomial models

  fit1 <- BinAddHaz(dis.bin ~ diab + arth + stro , data = disabData, weights = wgt,
                    attrib = FALSE)

  diseases <- as.matrix(disabData[,c("diab", "arth", "stro")])
  fit2 <- BinAddHaz(dis.bin ~ factor(age) -1 + diseases:factor(age), data = disabData,
                    weights = wgt, attrib = FALSE)

  LRTest(fit2, fit1)

  ## Comparing two multinomial models
## Not run:
  fit3 <- MultAddHaz(dis.mult ~ diab + arth + stro , data = disabData, weights = wgt,
                     attrib = FALSE)
  fit4 <- MultAddHaz(dis.mult ~ factor(age) -1 + diseases: factor(age), data = disabData,
                     weights = wgt, attrib = FALSE)
  LRTest(fit4, fit3)
## End(Not run)
```

`MultAddHaz` *Fit Multinomial Additive Hazard Models*

Description

This function fits multinomial additive hazard models subject to linear inequality constraints using the function `constrOptim` in the `stats` package for multinomial (multi-category) outcomes. It also calculates the cause-specific contributions to the disability prevalence for each category of the response variable based on the extension of the attribution method, as proposed by Yokota et al (2017).

Usage

```
MultAddHaz(formula, data, subset, weights, na.action, model = TRUE,
           contrasts = NULL, start, attrib = TRUE, attrib.var,
           collapse.background = FALSE, attrib.disease = FALSE,
           type.attrib = "abs", seed, bootstrap = FALSE, conf.level = 0.95,
           nbootstrap, parallel = FALSE, type.parallel = "snow", ncpus = 4,...)
```

Arguments

`formula`	a formula expression of the form response ~ predictors, similar to other regression models. In case of `attrib` = TRUE, the first predictor in the formula should be the attrib.var. See example.
`data`	an optional data frame or matrix containing the variables in the model. If not found in `data`, the variables are taken from environment(formula), typically the environment from which `MultAddHaz` is called.
`subset`	an optional vector specifying a subset of observations to be used in the fitting process. All observations are included by default.
`weights`	an optional vector of survey weights to be used in the fitting process.
`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. The 'factory-fresh' default is `na.omit`.
`model`	logical. If TRUE, the model frame is included as a component of the returned object.
`contrasts`	an optional list to be used for some or all of the factors appearing as variables in the model formula.
`start`	an optional vector of starting values. If not provided by the user, it is randomly generated.
`attrib`	logical. Should the attribution of chronic diseases/conditions for each disability level be estimated? Default is TRUE.
`attrib.var`	character indicating the name of the attribution variable to be used if `attrib` = TRUE. If missing, the attribution results will not be stratified by the levels of the attribution variable. The attribution variable must be the first variable included in the linear predictor in `formula`. See example.

`collapse.background`	logical. Should the background be collapsed across the levels of the `attrib.var` for each disability level? If `FALSE`, the background will be estimated for each level of the `attrib.var`. If `TRUE`, only one background will be estimated. If `TRUE`, `attrib.var` must be specified. Default is `FALSE`.
`attrib.disease`	logical. Should the attribution of diseases be stratified by the levels of the attribution variable for each disability level? If `FALSE`, the attribution of diseases will not be stratified by the levels of the `attrib.var`. If `TRUE`, the attribution of diseases will be estimated for each level of the `attrib.var`. If `TRUE`, interaction between diseases and the attribution variable must be provided in the `formula`. Default is `FALSE`.
`type.attrib`	type of attribution to be estimated. The options are `"abs"` for absolute contribution, `"rel"` for relative contribution, or `"both"` for both absolute and relative contributions. Default is `"abs"`.
`seed`	integer indicating the random seed.
`bootstrap`	logical. Should bootstrap percentile confidence intervals be estimated for the model parameters and attributions? Default is `FALSE`. See details.
`conf.level`	scalar containing the confidence level of the bootstrap percentile confidence intervals. Default is `0.95`.
`nbootstrap`	integer. Number of bootstrap replicates.
`parallel`	logical. Should parallel calculations be used to obtain the bootstrap percentile confidence intervals? Default is `FALSE`.
`type.parallel`	type of parallel operation to be used (if `parallel = TRUE`), with options: `"multicore"` and `"snow"`. Default is `"snow"`. See details.
`ncpus`	integer. Number of processes to be used in the parallel operation: typically one would choose this to be the number of available CPUs. Default is 4.
`...`	other arguments passed to or from the other functions.

Details

The model is a generalized linear model with a non-canonical link function, which requires a restriction on the linear predictor ($\eta \geq 0$) to produce valid probabilities. This restriction is implemented in the optimization procedure, with an adaptive barrier algorithm, using the function `constrOptim` in the `stats` package.

The variance-covariance matrix is based on the observed information matrix.

This version of the package only allows the calculation of non-parametric bootstrap percentile confidence intervals (CI). Stratified bootstrap is applied to each category of the outcome. Also, the function gives the user the option to do parallel calculation of the bootstrap CI. The `snow` parallel option is available for all operating systems (Windows, Linux, and Mac OS) while the `multicore` option is only available for Linux and Mac OS systems. These two calculations are done by calling the `boot` function in the `boot` package. For more details see the documentation of the `boot` package.

More information about the multinomial additive hazard model and the estimation of the contribution of chronic conditions to the disability prevalence can be found in the references.

Value

A list with arguments:

`coefficients`	column matrix with the regression coefficients.
`ci`	matrix with confidence intervals calculated via bootstraping (if `bootstrap = TRUE`) or as the inverse of the observed information matrix.
`resDeviance`	residual deviance.
`df`	degrees of freedom.
`pvalue`	column matrix of p-values based on the Wald test. Only provided if `bootstrap = FALSE`.
`stdError`	column matrix with the standard errors for the parameter estimates based on the inverse of the observed information matrix. Only provided if `bootstrap = FALSE`.
`vcov`	variance-covariance matrix (inverse of the observed information matrix). Only provided if `bootstrap = FALSE`.
`contribution`	for `type.attrib = "abs"` or `"rel"`, a matrix is provided. For `type.attrib = "both"`, a list with two matrices (`"abs"` and `"rel"`) is provided. This represents the contribution of each predictor in the model (usually diseases) to the disability prevalence. Percentile bootstrap confidence intervals are provided if `bootstrap = TRUE`. If `attrib = FALSE`, it returns a logical, `FALSE`.
`bootsRep`	matrix with the bootstrap replicates of the regression coefficients and contributions (if `attrib = TRUE`).
`conf.level`	confidence level of the bootstrap CI. Only provided if `bootstrap = TRUE`.
`bootstrap`	logical. Was bootstrap CI requested?
`call`	the matched call.

Author(s)

Renata T. C. Yokota. This function is based on the R code developed by Caspar W. N. Looman and Wilma J. Nusselder for the binomial additive hazard model with modifications and adaptations for the multinomial case.

References

Yokota, R.T.C., Van Oyen, H., Looman, C.W.N., Nusselder, W.J., Otava, M., Kifle, Y.W., Molenberghs, G. (2017). Multinomial additive hazard model to assess the disability burden using cross-sectional data. Biometrical Journal, 59(5), 901-917.

See Also

`BinAddHaz`

Examples

```
  data(disabData)

  ## Model without bootstrap CI and no attribution
## Not run:
  fit1 <- MultAddHaz(dis.mult ~ diab + arth + stro , data = disabData, weights = wgt,
                     attrib = FALSE)
  summary(fit1)

 ## Model with bootstrap CI and attribution without stratification, no parallel calculation
  # Warning message due to the low number of bootstrap replicates

  fit2 <- MultAddHaz(dis.mult ~ diab + arth + stro , data = disabData, weights = wgt,
                     attrib = TRUE, collapse.background = FALSE, attrib.disease = FALSE,
                   type.attrib = "both", seed = 111, bootstrap = TRUE, conf.level = 0.95,
                     nbootstrap = 5)
  summary(fit2)

  ## Model with bootstrap CI and attribution of diseases and background stratified by
  ## age, with parallel calculation of bootstrap CI
  # Warning message due to the low number of bootstrap replicates

  diseases <- as.matrix(disabData[,c("diab", "arth", "stro")])
  fit3 <- MultAddHaz(dis.mult ~ factor(age) -1 + diseases: factor(age), data = disabData,
                     weights = wgt, attrib = TRUE, attrib.var = age,
                collapse.background = FALSE, attrib.disease = TRUE, type.attrib = "both",
                     seed = 111, bootstrap = TRUE, conf.level = 0.95, nbootstrap = 5,
                     parallel = TRUE, type.parallel = "snow", ncpus = 4)

  summary(fit3)
## End(Not run)
```

Index