

Package ‘polyreg’

July 24, 2025

Title Direct polytomous regression for estimating multiplicative effects on cumulative incidence probabilities from competing risks data

Version 0.3.0

Maintainer ST <gestimation@gmail.com>

Description Direct polytomous regression is a competing risks model that jointly analyzes multiple competing events and estimates estimate multiplicative effects of a binary exposure. This model naturally enforces a sum restriction to cumulative incidence probabilities by reparameterizing nuisance parameters using polytomous log odds products. Risk ratios, odds ratios or sub-distribution hazard ratios for each of competing events are estimated by stratified inverse probability of censoring weighted estimators under adjustment for covariates in the nuisance and censoring models. Richardson model for binomial regression that uses a log odds product is also covered by this function.

License MIT + file LICENSE

Encoding UTF-8

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.2

Suggests knitr,
rmarkdown,
modelsummary,
testthat (>= 3.0.0)

Config/testthat/edition 3

LinkingTo Rcpp, RcppArmadillo, roptim

Imports nleqslv, Rcpp, RcppArmadillo, roptim, boot

VignetteBuilder knitr

URL <https://github.com/gestimation/polyreg>

BugReports <https://github.com/gestimation/polyreg/issues>

R topics documented:

polyreg	2
Index	5

polyreg	<i>Direct polynomial regression for competing risks, survival and binomial analysis</i>
---------	---

Description

Direct polynomial regression for competing risks, survival and binomial analysis

Usage

```
polyreg(
  nuisance.model,
  exposure,
  strata = NULL,
  data,
  subset.condition = NULL,
  na.action = na.omit,
  code.event1 = 1,
  code.event2 = 2,
  code.censoring = 0,
  code.exposure.ref = 0,
  effect.measure1 = "RR",
  effect.measure2 = "RR",
  time.point = NULL,
  outcome.type = "COMPETINGRISK",
  conf.level = 0.95,
  report.nuisance.parameter = FALSE,
  report.optim.convergence = FALSE,
  report.boot.conf = FALSE,
  boot.bca = TRUE,
  boot.parameter1 = 200,
  boot.parameter2 = 46,
  outer.optim.method = "nleqslv",
  inner.optim.method = "optim",
  optim.parameter1 = 1e-05,
  optim.parameter2 = 20,
  optim.parameter3 = 100,
  optim.parameter4 = 1e-05,
  optim.parameter5 = 200,
  optim.parameter6 = 1e-10,
  optim.parameter7 = 200,
  data.initial.values = NULL,
  should.normalize.covariate = TRUE,
  should.sort.data = TRUE,
  prob.bound = 1e-05
)
```

Arguments

`nuisance.model` formula Model formula representing outcome and covariates other than exposure

<code>exposure</code>	character Column name representing the binary exposure variable.
<code>strata</code>	character Column name representing the stratification variable for adjustment for dependent censoring. Defaults to NULL.
<code>data</code>	data.frame Input dataset containing the outcome, the exposure and covariates.
<code>subset.condition</code>	character Specifies a condition for subsetting the data. Defaults to NULL.
<code>na.action</code>	character Specifies a missing-data filter function, applied to the model frame, after any subset argument has been used. Defaults to na.omit.
<code>code.event1</code>	integer Specifies the code of event 1. Defaults to 1.
<code>code.event2</code>	integer Specifies the code of event 2. Defaults to 2.
<code>code.censoring</code>	integer Specifies the code of censoring. Defaults to 0.
<code>code.exposure.ref</code>	integer Specifies the code of the reference category of exposure. Defaults to 0.
<code>effect.measure1</code>	character Specifies the effect measure for event (RR, OR, SHR).
<code>effect.measure2</code>	character Specifies the effect measure for competing risk (RR, OR, SHR).
<code>time.point</code>	numeric The time point for exposure effects to be estimated.
<code>outcome.type</code>	character Specifies the type of outcome (COMPETINGRISK, SURVIVAL, BINOMIAL, PROPORTIONAL and POLY-PROPORTIONAL).
<code>conf.level</code>	numeric The level for confidence intervals.
<code>report.nuisance.parameter</code>	logical Specifies contents of return. (TRUE = report estimates of nuisance parameters, FALSE = otherwise). Defaults to FALSE.
<code>report.optim.convergence</code>	logical Specifies contents of return. (TRUE = report indicators of convergence of parameter estimation, FALSE = otherwise). Defaults to FALSE.
<code>report.boot.conf</code>	logical Specifies contents of return. (TRUE = report bootstrap confidence intervals, FALSE = otherwise). Defaults to FALSE.
<code>boot.bca</code>	logical Specifies the method of bootstrap confidence intervals (TRUE = BCA method, FALSE = normal approximation).
<code>boot.parameter1</code>	integer Number of replications for bootstrap confidence intervals.
<code>boot.parameter2</code>	numeric Seed used for bootstrap confidence intervals.
<code>outer.optim.method</code>	character Specifies the method of optimization (nleqslv, Broyden, Newton, optim, BFGS, SANN).
<code>inner.optim.method</code>	character Specifies the method of optimization (optim, BFGS, SANN).
<code>optim.parameter1</code>	numeric A threshold for initial value search in outer loop. Defaults to 1e-5.
<code>optim.parameter2</code>	integer Maximum number of iterations. Defaults to 20.
<code>optim.parameter3</code>	numeric Constraint range for parameters. Defaults to 100.

`optim.parameter4`
 numeric A threshold for determining convergence in outer loop. Defaults to 1e-5.

`optim.parameter5`
 integer Maximum number of iterations for nleqslv/optim in outer loop. Defaults to 200.

`optim.parameter6`
 numeric A threshold for determining convergence in inner loop. Defaults to 1e-10.

`optim.parameter7`
 integer Maximum number of iterations for optim in inner loop. Defaults to 200.

`should.normalize.covariate`
 logical Indicates whether covariates are normalized (TRUE = normalize, FALSE = otherwise). Defaults to TRUE.

`should.sort.data`
 logical Indicates whether data are initially sorted to reduce computation steps (TRUE = sort, FALSE = otherwise). Defaults to TRUE.

`prob.bound`
 numeric A threshold for clamping probabilities. Defaults to 1e-5.

`data.initlal.values`
 data.frame A dataset containing initial values. Defaults to NULL.

Value

A list of results from direct polynomial regression. `coefficient` and `cov` are estimated regression coefficients of exposure and covariates and their variance covariance matrix. `summary` meets requirement of `msummary` function. `diagnosis.statistics` includes inverse probability weights, influence functions, and predicted potential outcomes

Examples

```
data(diabetes.complications)
output <- polyreg(nuisance.model = Event(t,epsilon)~+1, exposure = 'fruitq1', data = diabetes.complications, e
library(modelsummary)
msummary(output$summary, statistic = c("conf.int"), exponentiate = TRUE)
```

Index

polyreg, [2](#)