Package `cprobit’

May 27, 2020

Type Package
Title Conditional Probit Model for Analysing Continuous Outcomes
Version 1.0.2
Author Ning Yilin, Tan Chuen Seng
Maintainer Ning Yilin <ningyilinnyl@gmail.com>
Depends R (>= 3.5.0)
Imports car, nortest, ggplot2, gridExtra
Description Implements the three-step workflow for robust analysis of change in
two repeated measurements of continuous outcomes, described in Ning et al.
(in press), `Robust estimation of the effect of an exposure on the change in
a continuous outcome’, BMC Medical Research Methodology.
License LGPL-3
Encoding UTF-8
LazyData true
RoxygenNote 7.1.0
NeedsCompilation no
Repository CRAN
Date/Publication 2020-05-27 10:00:08 UTC

R topics documented:

bg_variability ................................................................. 2
compile_est ................................................................. 2
cprobit ................................................................. 3
cprobit_step1 ............................................................. 5
estimate_sd_error ......................................................... 6
geom_mean ................................................................. 6
get_v ................................................................. 7
make_design_mat .......................................................... 7
profile_llh .............................................................. 8
update_estimate .......................................................... 8

Index 10
**bg_variability**  
*Inpatient blood glucose data for 1200 patients*

**Description**
A simulated dataset containing the variability of inpatient point-of-care blood glucose (BG) measurements from 1200 non-critical care adult patients in medical ward. BG variability is measured as the standard deviation of the BG readings within a day. Data was simulated based on real data.

**Usage**
```r
bg_variability
```

**Format**
A data frame with 1200 rows and 7 variables:

- `subject_id` Subject ID of each patient.
- `case_id` Case ID, with 1 and 2 referring to the first and second follow-up respectively.
- `y` BG variability of the first and second follow-up.
- `t` Binary indicator for the second follow-up.
- `sd0` Baseline BG variability.
- `age` Patients' age.
- `female` Binary indicator for being female.

**compile_est**  
*Inpernal function: generate commonly used summary statistics for estimates.*

**Description**
Inpernal function: generate commonly used summary statistics for estimates.

**Usage**
```r
compile_est(
  var,  
est,  
se = NULL,  
z_score = NULL,  
pval = NULL,  
value_null = 0,  
ci_lower = NULL,  
ci_upper = NULL,  
prefix = NULL,  
postfix = NULL
)
```
cprobit

Arguments

- **var**: Names of variables.
- **est**: Estimated regression coefficients.
- **se**: SE of estimates.
- **z_score**: Z score of estimates, i.e., \( \frac{\text{est}}{\text{se}} \).
- **pval**: P-value of estimates.
- **value_null**: Null effects for estimates, either with length 1 or length of \( \text{est} \). Default is 0.
- **ci_lower**: Lower bound of 95% CI of estimates.
- **ci_upper**: Upper bound of 95% CI of estimates.
- **prefix**: Prefix to the column names in the data.frame returned.
- **postfix**: Postfix to the column names in the data.frame returned.

Details

Vectorised, as long as the length of the input match.

---

**cprobit**

*Apply the three-step workflow for the analysis of two repeated outcomes from each subject*

---

Description

Apply the three-step workflow for the analysis of two repeated outcomes from each subject

Usage

```r
cprobit(
  formula,
  dat,
  index,
  transform = NULL,
  lambda = NA,
  resid_pval_threshold = 0.05
)
```

## S3 method for class 'cprobit'
summary(object, plot = FALSE, ...)

## S3 method for class 'cprobit'
print(x, ...)
Arguments

- **formula**: Formula for the model. Do not convert data type within the formula (e.g., `factor(x)` is not supported in `formula`). See Details.
- **dat**: A `data.frame` in the long format, with each row corresponding to one measurement from one subject, and two columns indicating the subject and case ID respectively. Variable names must not contain space or special characters.
- **index**: Names of variables indicating subject and case ID. Case ID must be coded as integers 1 and 2.
- **transform**: Whether a Box-Cox transformation should be applied to the outcome, taking value `NULL` (the default), `TRUE` or `FALSE`.
- **lambda**: Value of the Box-Cox transformation parameter to use. Default is `NA`, in which case it will be estimated from data.
- **resid_pval_threshold**: The threshold for the Lilliefors p-value of the residuals to determine whether a Box-Cox transformation on the outcome is necessary. Default is 0.05.
- **object**: Model fitted using `cprobit` function.
- **plot**: Whether residual qq-plots should be plotted. Default is `FALSE`.
- **...**: Additional arguments affecting the summary produced (not yet implemented).
- **x**: Model fitted using `cprobit` function.

Details

Specify the formula for the repeated measurements instead of the change in the outcome, but without any time-invariant component that would have been eliminated after taking the difference. Interaction between two variables can be specified in the formula using `*` or `:`; but users need to create their own variable for interaction involving three or more variables.

If `transform = NULL`, the workflow will determine the need for a Box-Cox transformation on the outcome (i.e., Step 3) based on the residual diagnostics in Step 2. A Box-Cox transformation will be used if the p-value of the Lilliefors test is smaller than `resid_pval_threshold` (default is 0.05). If `transform = TRUE`, analyses will always be performed on both the observed and Box-Cox transformed outcomes. If `transform = FALSE`, analysis will only be performed on the observed outcomes.

Value

Returns a list.

References

Examples

# Apply the three-step workflow to assess the association between the
# baseline glucose variability and the change in the glucose variability in
# the subsequent two days.
# Although age and gender are available, they do not need to be explicitly
# adjusted for in the cprobit model.
data(bg_variability)
head(bg_variability)
model <- cprobit(formula = y ~ t + t:sd0, dat = bg_variability,
                 index = c("subject_id", "case_id"))
summary(model, plot = TRUE)

Description

Implements the Step 1 of the proposed workflow, where a cprobit model is applied to analyse
whether there is an increase in the outcome within each subject.

Usage

cprobit_step1(y_name, x_names, dat_diff, var_names = NULL)

Arguments

  y_name            Name of outcome variable for Step 1.
  x_names           Names of covariates for Step 1.
  dat_diff          A data.frame containing the difference data.
  var_names         Variable names for the estimates.

Value

  Returns a data.frame summarising the Step 1 estimates (coef) and the covariance matrix for the
  Step 1 estimates (vcov).
estimate_sd_error

*Inpernal function: estimate the SD of error terms in the difference model*

**Description**

Inpernal function: estimate the SD of error terms in the difference model

**Usage**

```r
estimate_sd_error(beta_c, y1, y2, lambda = NA, design_mat_diff)
```

**Arguments**

- `beta_c`: Numeric vector of Step 1 estimates.
- `y1`: Numeric vector of the observed outcome at observation time 1.
- `y2`: Numeric vector of the observed outcome at observation time 2.
- `lambda`: The Box-Cox transformation parameter. Default is `NA`, indicating no need for a transformation. See Details.
- `design_mat_diff`: Numeric matrix of the design matrix for difference.

**Value**

Returns the estimate for `sigma_delta` if `lambda = NULL`, or `sigma_delta_lambda` on the transformed scale.

---

geom_mean

*Inpernal function: compute geometric mean of a positive variable*

**Description**

Inpernal function: compute geometric mean of a positive variable

**Usage**

```r
geom_mean(x)
```

**Arguments**

- `x`: A numeric vector.
get_v

Inpernal function: compute difference in the (transformed) outcome

Description
Inpernal function: compute difference in the (transformed) outcome

Usage
get_v(y1, y2, lambda = NA, scaled = TRUE)

Arguments
- y1: Numeric vector of the observed outcome at observation time 1.
- y2: Numeric vector of the observed outcome at observation time 2.
- lambda: The Box-Cox transformation parameter. Default is NA, indicating no need for a transformation. See Details.
- scaled: Whether the difference in the transformed outcomes should be scaled by the Jacobian.

Value
Returns the difference in the observed outcomes if lambda = NA, or the difference in the scaled transformed outcomes with transformation parameter lambda.

make_design_mat

Inpernal function: construct design matrix without the intercept term.

Description
Inpernal function: construct design matrix without the intercept term.

Usage
make_design_mat(lp, dat, remove_intercept = TRUE)

Arguments
- lp: Formula for the linear predictor part, as a string.
- dat: Data to construct the design matrix from.
- remove_intercept: Whether the first column should be removed. Default is TRUE (to remove the intercept term).
update_estimate

**Value**

Returns a list containing the constructed design matrix and the original variable names. In the column names of the design matrix returned, any : in variable names are replaced with . to avoid computational issues when using the design matrix to fit model.

---

**profile_llh**

*Inpernal function: profile log-likelihood of lambda*

---

**Description**

Inpernal function: profile log-likelihood of lambda

**Usage**

`profile_llh(lambda, beta_c, y1, y2, design_mat_diff)`

**Arguments**

- `lambda` A numeric vector of the estimated parameter. Default is `NA`, indicating no need for a transformation. See **Details**.
- `beta_c` Numeric vector of Step 1 estimates.
- `y1` Numeric vector of the observed outcome at observation time 1.
- `y2` Numeric vector of the observed outcome at observation time 2.
- `design_mat_diff` Numeric matrix of the design matrix for difference.

**Value**

Returns the profile log likelihood (not the negative value).

---

**update_estimate**

*Inpernal function: update Step 1 estimates to obtain linear exposure effect on (transformed) outcome*

---

**Description**

Inpernal function: update Step 1 estimates to obtain linear exposure effect on (transformed) outcome
update_estimate

Usage

update_estimate(
  y1_name,
  y2_name,
  var_names = NULL,
  dat_diff,
  res_step1,
  transform = FALSE
)

Arguments

y1_name Name of observed outcome at observation time 1.
y2_name Name of observed outcome at observation time 2.
var_names Variable names for the estimates.
dat_diff A \texttt{data.frame} containing the difference data.
res_step1 Results from Step 1 of the workflow.
transform Whether the outcome should be transformed. Default is \texttt{FALSE}.

Value

Returns a list: a \texttt{data.frame} summarising the estimated linear exposure effect, the estimated standard deviation of the error terms from the difference model, the covariance matrix of the estimated exposure effects, a \texttt{data.frame} summarising the estimated transformation parameter, and the residuals.
Index

*Topic datasets
  bg_variability, 2
bg_variability, 2
compile_est, 2
cprobit, 3
cprobit_step1, 5
estimate_sd_error, 6
geom_mean, 6
get_v, 7
make_design_mat, 7
print.cprobit(cprobit), 3
profile_1lh, 8
summary.cprobit(cprobit), 3
update_estimate, 8