Package: beeca (via r-universe)

September 3, 2024

Title Binary Endpoint Estimation with Covariate Adjustment

Version 0.1.3.9000

Description Performs estimation of marginal treatment effects for binary outcomes when using logistic regression working models with covariate adjustment (see discussions in Magirr et al (2024) <https://osf.io/9mp58/>). Implements the variance estimators of Ge et al (2011) <doi:10.1177/009286151104500409> and Ye et al (2023) <doi:10.1080/24754269.2023.2205802>.

Maintainer Alex Przybylski <alexander.przybylski@novartis.com>

License LGPL (>= 3)

Encoding UTF-8

Roxygen list(markdown = TRUE)

RoxygenNote 7.2.3

Suggests knitr, rmarkdown, testthat (>= 3.0.0), tidyr, marginaleffects, margins, RobinCar (== 0.3.0)

Config/testthat/edition 3

Depends R (>= 2.10)

LazyData true

Imports dplyr, lifecycle, sandwich, stats

VignetteBuilder knitr

URL https://openpharma.github.io/beeca/

BugReports https://github.com/openpharma/beeca/issues

Repository https://openpharma.r-universe.dev

RemoteUrl https://github.com/openpharma/beeca

RemoteRef HEAD

RemoteSha 6c269af55229edee14304df75ceca607b2c7b3de

Contents

`apply_contrast` *Apply contrast to calculate marginal estimate of treatment effect and corresponding standard error*

Description

Calculates the marginal estimate of treatment effect and its corresponding standard error based on a fitted GLM object using specified contrast (summary measure) methods

Usage

```
apply_contrast(
  object,
  contrast = c("diff", "rr", "or", "logrr", "logor"),
  reference
)
```

Arguments

`object`	a fitted `glm` object augmented with `counterfactual.predictions`, `counterfactual.means` and `robust_varcov`.
`contrast`	a string specifying the type of contrast to apply. Accepted values are "diff" (risk difference), "rr" (risk ratio), "or" (odds ratio), "logrr" (log risk ratio), "logor" (log odds ratio). Note: log-transformed ratios (logrr and logor) work better compared to rr and or when computing confidence intervals using normal approximation. The choice of contrast affects how treatment effects are calculated and interpreted. Default is `diff`.
`reference`	a string indicating which treatment group should be considered as the reference level. Accepted values are one of the levels in the treatment variable. Default to the first level used in the `glm` object. This parameter influences the calculation of treatment effects relative to the chosen reference group.

Details

The `apply_constrast()` functions computes the summary measure between two arms based on the estimated marginal effect and its variance-covariance matrix using the Delta method.

Note: Ensure that the `glm` object has been adequately prepared with `average_predictions()` and `estimate_varcov()` before applying `apply_contrast()`. Failure to do so may result in errors indicating missing components.

Value

An updated `glm` object with two additional components appended: `marginal_est` (marginal estimate of the treatment effect) and `marginal_se` (standard error of the marginal estimate). These appended component provide crucial information for interpreting the treatment effect using the specified contrast method.

See Also

`get_marginal_effect()` for estimating marginal effects directly from an original `glm` object

Examples

```
trial01$trtp <- factor(trial01$trtp)
fit1 <- glm(aval ~ trtp + bl_cov, family = "binomial", data = trial01) |>
  predict_counterfactuals(trt = "trtp") |>
  average_predictions() |>
  estimate_varcov(method = "Ye") |>
  apply_contrast("diff", reference = "0")

# Assuming `trial01` is a dataset with treatment (`trtp`)
# and baseline covariate (`bl_cov`)
trial01$trtp <- factor(trial01$trtp)
fit1 <- glm(aval ~ trtp + bl_cov, family = "binomial", data = trial01)

# Preprocess fit1 as required by apply_contrast
fit2 <- fit1 |>
  predict_counterfactuals(trt = "trtp") |>
  average_predictions() |>
  estimate_varcov(method = "Ye")

# Apply contrast to calculate marginal estimates
fit3 <- apply_contrast(fit2, contrast = "diff", reference = "0")

fit3$marginal_est
fit3$marginal_se
```

average_predictions *Average over counterfactual predictions*

Description

`average_predictions()` averages counterfactual predictions stored within a `glm` object. This is pivotal for estimating treatment contrasts and associated variance estimates using g-computation. The function assumes predictions are generated via `predict_counterfactuals()`.

Usage

```
average_predictions(object)
```

Arguments

`object`	a fitted `glm` object augmented with counterfactual predictions named: `counterfactual.predictions`

Details

The `average_predictions()` function calculates the average over the counterfactual predictions which can then be used to estimate a treatment contrast and associated variance estimate.

The function appends a `glm` object with the averaged counterfactual predictions.

Note: Ensure that the `glm` object has been adequately prepared with `predict_counterfactuals()` before applying `average_predictions()`. Failure to do so may result in errors indicating missing components.

Value

an updated `glm` object appended with an additional component `counterfactual.means`.

See Also

`predict_counterfactuals()` for generating counterfactual predictions.

`estimate_varcov()` for estimating the variance-covariate matrix of mariginal effects

`get_marginal_effect()` for estimating marginal effects directly from an original `glm` object

Examples

```
# Use the trial01 dataset
data(trial01)

# ensure the treatment indicator is a factor
trial01$trtp <- factor(trial01$trtp)

# fit glm model for trial data
fit1 <- glm(aval ~ trtp + bl_cov, family = "binomial", data = trial01)
```

```
# Preprocess fit1 as required by average_predictions
fit2 <- fit1 |>
  predict_counterfactuals(trt = "trtp")

# average over the counterfactual predictions
fit3 <- average_predictions(fit2)

# display the average predictions
fit3$counterfactual.means
```

`estimate_varcov` *Estimate variance-covariance matrix for marginal estimand based on GLM model*

Description

Main variance estimation function. Estimates the variance-covariance matrix of a marginal estimand for a generalized linear model (GLM) object using specified methods. This function supports both Ge's and Ye's methods for variance estimation, accommodating different estimand specifications.

Usage

```
estimate_varcov(
  object,
  strata = NULL,
  method = c("Ge", "Ye"),
  type = c("HC0", "model-based", "HC3", "HC", "HC1", "HC2", "HC4", "HC4m", "HC5"),
  mod = FALSE
)
```

Arguments

`object`	a fitted `glm` object augmented with `counterfactual.predictions`, `counterfactual.predictions` and `counterfactual.means`
`strata`	an optional string or vector of strings specifying the names of stratification variables. Relevant only for Ye's method and used to adjust the variance-covariance estimation for stratification. If provided, each specified variable must be present in the model.
`method`	a string indicating the chosen method for variance estimation. Supported methods are `Ge` and `Ye`. The default method is `Ge` based on Ge et al (2011) which is suitable for the variance estimation of conditional average treatment effect. The method `Ye` is based on Ye et al (2023) and is suitable for the variance estimation of population average treatment effect. For more details, see Magirr et al. (2024).

`type` a string indicating the type of variance estimator to use (only applicable for Ge's method). Supported types include HC0 (default), model-based, HC3, HC, HC1, HC2, HC4, HC4m, and HC5. See vcovHC for heteroscedasticity-consistent estimators. This parameter allows for flexibility in handling heteroscedasticity and model specification errors.

`mod` For Ye's method, the implementation of open-source RobinCar package has an additional variance decomposition step when estimating the robust variance, which then utilizes different counterfactual outcomes than the original reference. Set `mod = TRUE` to use exactly the implementation method described in Ye et al (2022), default to `FALSE` to use the modified implementation in RobinCar and Bannick et al (2023) which improves stability.

Details

The `estimate_varcov` function facilitates robust variance estimation techniques for GLM models, particularly useful in clinical trial analysis and other fields requiring robust statistical inference. It allows researchers to account for complex study designs, including stratification and different treatment contrasts, by providing a flexible interface for variance-covariance estimation.

Note: Ensure that the `glm` object has been adequately prepared with `predict_counterfactuals` and `average_predictions` before applying `estimate_varcov()`. Failure to do so may result in errors indicating missing components.

Value

an updated `glm` object appended with an additional component `robust_varcov`, which is the estimated variance-covariance matrix of the marginal effect. The matrix format and estimation method are indicated in the matrix attributes.

References

Ye T. et al. (2023) Robust variance estimation for covariate-adjusted unconditional treatment effect in randomized clinical trials with binary outcomes. Statistical Theory and Related Fields

Ge M. et al. (2011) Covariate-Adjusted Difference in Proportions from Clinical Trials Using Logistic Regression and Weighted Risk Differences. Drug Information Journal.

Bannick, M. S., et al. A General Form of Covariate Adjustment in Randomized Clinical Trials. arXiv preprint arXiv:2306.10213 (2023).

See Also

`average_predictions()` for averaging counterfactual predictions.

`apply_contrast()` for computing a summary measure.

`get_marginal_effect()` for estimating marginal effects directly from an original `glm` object

Examples

```
# Example usage with a binary outcome GLM model
trial01$trtp <- factor(trial01$trtp)
fit1 <- glm(aval ~ trtp + bl_cov, family = "binomial", data = trial01)
```

```
#' # Preprocess fit1 as required by estimate_varcov
fit2 <- fit1 |>
  predict_counterfactuals(trt = "trtp") |>
  average_predictions()

# Estimate variance-covariance using Ge's method
fit3_ge <- estimate_varcov(fit2, method = "Ge")
print(fit3_ge$robust_varcov)


# Estimate variance-covariance using Ye's method with stratification
fit4 <- glm(aval ~ trtp + bl_cov_c, family = "binomial", data = trial01) |>
  predict_counterfactuals(trt = "trtp") |>
  average_predictions()
fit4_ye <- estimate_varcov(fit4, method = "Ye", strata = "bl_cov_c")
print(fit4_ye$robust_varcov)
```

`get_marginal_effect` *Estimate marginal treatment effects using a GLM working model*

Description

Estimates the marginal treatment effect from a logistic regression working model using a specified choice of variance estimator and contrast.

Usage

```
get_marginal_effect(
  object,
  trt,
  strata = NULL,
  method = "Ge",
  type = "HC0",
  contrast = "diff",
  reference,
  mod = FALSE
)
```

Arguments

`object`	a fitted glm object.
`trt`	a string specifying the name of the treatment variable in the model formula. It must be one of the linear predictor variables used in fitting the `object`.
`strata`	an optional string or vector of strings specifying the names of stratification variables. Relevant only for Ye's method and used to adjust the variance-covariance estimation for stratification. If provided, each specified variable must be present in the model.

method	a string indicating the chosen method for variance estimation. Supported methods are Ge and Ye. The default method is Ge based on Ge et al (2011) which is suitable for the variance estimation of conditional average treatment effect. The method Ye is based on Ye et al (2023) and is suitable for the variance estimation of population average treatment effect. For more details, see Magirr et al. (2024).
type	a string indicating the type of variance estimator to use (only applicable for Ge's method). Supported types include HC0 (default), model-based, HC3, HC, HC1, HC2, HC4, HC4m, and HC5. See vcovHC for heteroscedasticity-consistent estimators.
contrast	a string indicating choice of contrast. Defaults to 'diff' for a risk difference. See apply_contrast.
reference	a string indicating which treatment group should be considered as the reference level. Accepted values are one of the levels in the treatment variable. Default to the first level used in the glm object. This parameter influences the calculation of treatment effects relative to the chosen reference group.
mod	for Ye's method, the implementation of open-source RobinCar package has an additional variance decomposition step when estimating the robust variance, which then utilizes different counterfactual outcomes than the original reference. Set mod = TRUE to use exactly the implementation method described in Ye et al (2022), default to FALSE to use the modified implementation in RobinCar and Bannick et al (2023) which improves stability.

Details

The get_marginal_effect function is a wrapper that facilitates advanced variance estimation techniques for GLM models with covariate adjustment targeting a population average treatment effect. It is particularly useful in clinical trial analysis and other fields requiring robust statistical inference. It allows researchers to account for complex study designs, including stratification and treatment contrasts, by providing a flexible interface for variance-covariance estimation.

Value

an updated glm object appended with marginal estimate components: counterfactual.predictions (see predict_counterfactuals), counterfactual.means (see average_predictions), robust_varcov (see estimate_varcov), marginal_est, marginal_se (see apply_contrast) and marginal_results. A summary is shown below

counterfactual.predictions	Counterfactual predictions based on the working model. For each subject in the input glm data, th
counterfactual.means	Average of the counterfactual predictions for each level of the treatment variable.
robust_varcov	Variance-covariance matrix of the marginal effect estimate for each level of treatment variable, wi
marginal_est	Marginal treatment effect estimate for a given contrast.
marginal_se	Standard error estimate of the marginal treatment effect estimate.
marginal_results	Analysis results data (ARD) containing a summary of the analysis for subsequent reporting.

Examples

```
trial01$trtp <- factor(trial01$trtp)
fit1 <- glm(aval ~ trtp + bl_cov, family = "binomial", data = trial01) |>
  get_marginal_effect(trt = "trtp", method = "Ye", contrast = "diff", reference = "0")
fit1$marginal_results
```

`ge_macro_trial01` *Output from the Ge et al (2011) SAS macro applied to the trial01 dataset*

Description

For purposes of implementation comparisons, these are the result outputs from the SAS macro provided with the Ge et al (2011) publication (https://doi.org/10.1177/009286151104500409), applied to the trial01 dataset included with beeca, adjusting for treatment (trtp) and a single covariate (bl_cov) and targeting a risk difference contrast.

Usage

```
ge_macro_trial01
```

Format

`ge_macro_trial01` A tibble with 1 row and 6 columns:

diff Marginal risk difference estimate

se Standard error of marginal risk difference estimate

pt Marginal risk in treated

pC Marginal risk in controls

lower Lower bound of 95 percent confidence interval of risk difference estimate

upper Upper bound of 95 percent confidence interval of risk difference estimate

---

`margins_trial01` *Output from the Margins SAS macro applied to the trial01 dataset*

Description

For purposes of implementation comparisons, these are the result outputs from the SAS Margins macro (https://support.sas.com/kb/63/038.html), applied to the trial01 dataset included with beeca, adjusting for treatment (trtp) and a single covariate (bl_cov) and targeting a risk difference contrast.

Usage

```
margins_trial01
```

Format

`margins_trial01` A tibble with 1 row and 11 columns:

Estimate Marginal risk difference estimate

ChiSq Wald Chi-Square statistic

Row Row number

StdErr Standard error of marginal risk difference estimate

Lower Lower bound of 95 percent confidence interval of estimate

Upper Upper bound of 95 percent confidence interval of estimate

Contrast Descriptive label for contrast

df Degrees of freedom

Pr p-value

Alpha Significance level alpha

label Label for contrast

`predict_counterfactuals` *Predict counterfactual outcomes in GLM models*

Description

This function calculates counterfactual predictions for each level of a specified treatment variable in a generalized linear model (GLM). It is designed to aid in the assessment of treatment effects by predicting outcomes under different treatments under causal inference framework.

Usage

```
predict_counterfactuals(object, trt)
```

Arguments

`object`	a fitted `glm` object for which counterfactual predictions are desired.
`trt`	a string specifying the name of the treatment variable in the model formula. It must be one of the linear predictor variables used in fitting the `object`.

Details

The function works by creating two new datasets from the original data used to fit the GLM model. In these datasets, the treatment variable is set to each of its levels across all records (e.g., patients).

Predictions are then made for each dataset based on the fitted GLM model, simulating the response variable under each treatment condition.

The results are stored in a tidy format and appended to the original model object for further analysis or inspection.

For averaging counterfactual outcomes, apply `average_predictions()`.

Value

an updated `glm` object appended with an additional component `counterfactual.predictions`.

This component contains a tibble with two columns: `cf_pred_0` and `cf_pred_1`, representing counterfactual predictions for each level of the treatment variable. A descriptive `label` attribute explains the counterfactual scenario associated with each column.

See Also

`average_predictions()` for averaging counterfactual predictions.

`get_marginal_effect()` for estimating marginal effects directly from an original `glm` object

Examples

```
# Preparing data and fitting a GLM model
trial01$trtp <- factor(trial01$trtp)
fit1 <- glm(aval ~ trtp + bl_cov, family = "binomial", data = trial01)

# Generating counterfactual predictions
fit2 <- predict_counterfactuals(fit1, "trtp")

# Accessing the counterfactual predictions
fit2$counterfactual.predictions
attributes(fit2$counterfactual.predictions)
```

`trial01` *Example trial dataset 01*

Description

A simplified example of a simulated trial dataset, with missing data.

Usage

```
trial01
```

Format

`trial01` A data frame with 268 rows and 9 columns:

usubjid Unique subject identifier

aval Primary outcome variable (1 = yes/0 = no)

trtp Planned treatment

bl_cov Baseline covariate (numeric)

bl_cov_c Dichotomized version of bl_cov (category of 1 or 0)

region_2, ..., region_5 Indicators for region (1 = yes/0 = no)

`trial02_cdisc` *Example CDISC Clinical Trial Dataset in ADaM Format*

Description

This dataset is a simplified, binary outcome version of a sample Phase 2 clinical trial dataset formatted according to the Analysis Data Model (ADaM) standards set by the Clinical Data Interchange Standards Consortium (CDISC). It is designed for training and educational purposes, showcasing how clinical trial data can be structured for statistical analysis.

Usage

```
trial02_cdisc
```

Format

A data frame with 254 rows and 13 columns, representing trial participants and key variables:

USUBJID Unique subject identifier (alphanumeric code). A code unique to the clinical trial

PARAM Parameter name indicating the specific measurement or outcome assessed.

AGE Age of the participant at study enrollment, in years.

AGEGR1 Categorical representation of age groups.

AGEGR1N Numeric code representing age groups, used for statistical modeling.

RACE Self-identified race of the participant

RACEN Numeric representation of race categories, used for statistical modeling.

SEX Participant's sex at birth.

TRTP Planned treatment assignment, indicating the specific intervention or control condition.

TRTPN Numeric code for the planned treatment, simplifying data analysis procedures.

AVAL Analysis value, representing the primary outcome measure for each participant.

AVALC Character representation of the analysis value, used in descriptive summaries.

FASFL Full analysis set flag, indicating if the participant's data is included in the full analysis set.

Details

This dataset serves as an illustrative example for those learning about the ADaM standard in clinical trials. It includes common variables like demographic information, treatment assignments, and outcome measures.

Data privacy and ethical considerations have been addressed through the anonymization of subject identifiers and other sensitive information. The dataset is intended for educational and training purposes only.

Note

The numeric codes for categorical variables such as RACEN and TRTPN are arbitrary and should be interpreted within the context of this dataset. For example, refer to the categorical representations for additional context.

Source

This dataset has been reformatted for educational use from the safetyData package, specifically adam_adtte. For the original data and more detailed information, please refer to the safetyData documentation.

Index