Package ‘outstandR’

January 21, 2026

Title Model-Based Standardisation for Indirect Treatment Comparison with Limited Subject-Level Data

Version 1.0.0

Description For the problem of indirect treatment comparison with limited subject-level data, this package provides tools for model-based standardisation with several different computation approaches.
See Remiro-Azócar A, Heath A, Baio G (2022) ``Parametric G-computation for compatible indirect treatment comparisons with limited individual patient data'',
Res. Synth. Methods, 1–31. ISSN 1759-2879, <doi:10.1002/jrsm.1565>.

License GPL (>= 3)

Encoding UTF-8

Language en-GB

RoxygenNote 7.3.3

Depends R (>= 4.1.0)

Imports boot, cli, copula, crayon, dplyr, ggplot2, glue, pillar, purrr, rlang, rstanarm, stats, stringr, tibble, tidyr, tidyselect, Rdpack (>= 0.7)

Suggests causaldata, doSNOW, gridExtra, here, knitr, maicplus, marginaleffects, MASS, parallel, reshape2, rmarkdown, stdReg2, testthat (>= 3.0.0)

Config/testthat/edition 3

RdMacros Rdpack

URL `https://StatisticsHealthEconomics.github.io/outstandR/`

BugReports `https://github.com/StatisticsHealthEconomics/outstandR/issues/`

LazyData false

NeedsCompilation no

Author Nathan Green [aut, cre, cph] (ORCID:
<`https://orcid.org/0000-0003-2745-1736`>, ROR:
<`https://ror.org/02jx3x895`>),
Chengyang Gao [aut],
Antonio Remiro-Azocar [aut] (ORCID:
<`https://orcid.org/0000-0002-7263-4251`>)

Maintainer Nathan Green <n.green@ucl.ac.uk>

Repository CRAN

Date/Publication 2026-01-21 20:00:02 UTC

Contents

`AC_IPD_binY_contX` *Individual-level patient data for binary outcome, continuous covariates*

Description

This data set contains simulated patient covariate and outcome values.

Usage

```
data(AC_IPD_binY_contX)
```

Format

`y ~ PF_cont_1 + PF_cont_2 + trt + trt:(EM_cont_1 + EM_cont_2)`:

id Numeric unique identifier

PF_cont_1 Numeric prognostic factor continuous covariate

PF_cont_2 Numeric prognostic factor continuous covariate

EM_cont_1 Numeric effect modifier continuous covariate

EM_cont_2 Numeric effect modifier continuous covariate

trt Factor treatment identifier. Levels A, C

y Integer binary outcome

true_eta Numeric linear predictor

Source

Simulated data

References

Remiro-Azocar A, Heath A, Baio G (2022)

`AC_IPD_contY_mixedX` *Individual-level patient data for continuous outcome, mixed covariates*

Description

This data set contains simulated patient covariate and outcome values. Corresponds to ALD data set.

Usage

```
data(AC_IPD_contY_mixedX)
```

Format

`y ~ X1 + X3 + X4 + trt + trt:(X2 + X3 + X4):`

id Numeric unique identifier

X1 Numeric prognostic factor continuous covariate

X2 Numeric prognostic factor and effect modifier binary covariate

X3 Numeric prognostic factor and effect modifier continuous covariate

X4 Numeric effect modifier binary covariate

trt Factor treatment identifier. Levels A, C

y Integer binary outcome

true_eta Numeric linear predictor

Source

Simulated data

References

Remiro-Azocar A, Heath A, Baio G (2022)

`AC_IPD_countY_contX` *Individual-level patient data for count outcome, continuous covariates*

Description

This data set contains simulated patient covariate and outcome values. Corresponds to ALD data set.

Usage

```
data(AC_IPD_countY_contX)
```

Format

`y ~ PF_cont_1 + PF_cont_2 + trt + trt:(EM_cont_1 + EM_cont_2):`

id Numeric unique identifier

PF_cont_1 Numeric prognostic factor continuous covariate

PF_cont_2 Numeric prognostic factor continuous covariate

EM_cont_1 Numeric effect modifier continuous covariate

EM_cont_2 Numeric effect modifier continuous covariate

trt Factor treatment identifier. Levels A, C

y Integer non-negative count outcome

true_eta Numeric linear predictor

Source

Simulated data

References

Remiro-Azocar A, Heath A, Baio G (2022)

`BC_ALD_binY_contX` *Aggregate level patient data for binary outcome, continuous covariates*

Description

This data set contains summaries of simulated patient covariate and outcome values.

Usage

```
data(BC_ALD_binY_contX)
```

Format

`y ~ PF_cont_1 + PF_cont_2 + trt + trt:(EM_cont_1 + EM_cont_2):`

variable String covariate or outcome name. From EM_cont_1, EM_cont_2, PF_cont_1, PF_cont_2, y.

statistic String summary statistic name. From mean, sd, sum, N

value Numeric value

trt Treatment (arm) name. From B, C

Source

Simulated data

References

Remiro-Azocar A, Heath A, Baio G (2022)

BC_ALD_contY_mixedX *Aggregate level patient data for continuous outcome, mixed covariates*

Description

This data set contains summaries of simulated patient covariate and outcome values. Corresponds to IPD data set.

Usage

```
data(BC_ALD_contY_mixedX)
```

Format

`y ~ X1 + X3 + X4 + trt + trt:(X2 + X3 + X4)`:

variable String covariate or outcome name. From X1, X2, X3, X4, y.

statistic String summary statistic name. From mean, sd, prob, sum, N

value Numeric value

trt Treatment (arm) name. From B, C

Source

Simulated data

References

Remiro-Azocar A, Heath A, Baio G (2022)

BC_ALD_countY_contX *Aggregate level patient data for count outcome, continuous covariates*

Description

This data set contains summaries of simulated patient covariate and outcome values. Corresponds to IPD data set.

Usage

```
data(BC_ALD_countY_contX)
```

Format

`y ~ PF_cont_1 + PF_cont_2 + trt + trt:(EM_cont_1 + EM_cont_2):`

variable String covariate or outcome name. From EM_cont_1, EM_cont_2, PF_cont_1, PF_cont_2, y.

statistic String summary statistic name. From mean, sd, sum, N

value Numeric value

trt Treatment (arm) name. From B, C

Source

Simulated data

References

Remiro-Azocar A, Heath A, Baio G (2022)

`calculate_ate` *Calculate Average Treatment Effect*

Description

Computes the average treatment effect (ATE) based on the specified effect scale.

Usage

```
calculate_ate(mean_comp, mean_ref, effect)
```

Arguments

`mean_comp`, `mean_ref`
Mean of the outcome for the comparator and reference / common

`effect` A character string specifying the effect scale. Options are:

"log_odds" Log-odds difference.

"risk_difference" Risk difference.

"delta_z" Probit scale difference (z-scores).

"log_relative_risk_rare_events" Log relative risk for rare events.

"log_relative_risk" Log relative risk.

Value

Numeric computed average treatment effect on the specified scale.

Examples

```
calculate_ate(mean_comp = 0.7, mean_ref = 0.5, effect = "log_odds")
calculate_ate(mean_comp = 0.7, mean_ref = 0.5, effect = "risk_difference")
```

`calculate_trial_mean` *Calculate Trial Mean Wrapper*

Description

Calculate Trial Mean Wrapper

Usage

```
calculate_trial_mean(ald, tid, effect, family)
```

Arguments

`ald`	Aggregate level data. Data frame in long format.
`tid`	Treatment ID
`effect`	Effect name. String.
`family`	Family distribution

Value

Numeric mean value.

`calculate_trial_mean_binary` *Calculate Trial Mean Binary Data*

Description

Calculate Trial Mean Binary Data

Usage

```
calculate_trial_mean_binary(ald, tid, effect)
```

Arguments

`ald`	Aggregate level data. Data frame in long format.
`tid`	Treatment ID
`effect`	Effect name. String.

Value

Numeric mean value.

`calculate_trial_mean_continuous` *Calculate Trial Mean Continuous Data*

Description

Calculate Trial Mean Continuous Data

Usage

```
calculate_trial_mean_continuous(ald, tid, effect, verbatim = FALSE)
```

Arguments

`ald`	Aggregate level data. Data frame in long format.
`tid`	Treatment ID
`effect`	Effect name. String.
`verbatim`	Print messages, logical

Value

Numeric mean value.

`calculate_trial_mean_count` *Calculate Trial Mean Count Data*

Description

Calculate Trial Mean Count Data

Usage

```
calculate_trial_mean_count(ald, tid, effect, verbatim = FALSE)
```

Arguments

`ald`	Aggregate level data. Data frame in long format.
`tid`	Treatment ID
`effect`	Effect name. String.
`verbatim`	Print messages, logical

Value

Numeric mean value.

calculate_trial_variance

Calculate trial variance

Description

Computes the variance of treatment effects for a trial based on the specified family distribution.

Usage

```
calculate_trial_variance(ald, tid, effect, family)
```

Arguments

ald	Aggregate-level data. Data frame.
tid	Treatment identifier used to extract relevant columns from ald.
effect	A character string specifying the effect scale (e.g., "log_odds", "risk_difference").
family	A character string specifying the model family (e.g., "binomial", "gaussian").

Value

Numeric computed variance of treatment effects.

Examples

```
ald <- data.frame(trt = c("B","C","B","C"),
                  variable = c(NA, NA, "y", "y"),
                  statistic = c("N", "N", "sum", "sum"),
                  value = c(100, 100, 50, 60))

calculate_trial_variance(
  ald, tid = "B", effect = "log_odds", family = "binomial")
```

calculate_trial_variance_binary

Calculate trial variance binary

Description

Calculate trial variance binary

Usage

```
calculate_trial_variance_binary(ald, tid, effect)
```

Arguments

`ald`	Aggregate level data
`tid`	Treatment ID
`effect`	Effect

Value

Numeric value of total variance.

`calculate_trial_variance_continuous`
Calculate trial variance continuous

Description

Calculate trial variance continuous

Usage

```
calculate_trial_variance_continuous(ald, tid, effect, verbatim = FALSE)
```

Arguments

`ald`	Aggregate level data. Data frame in long format.
`tid`	Treatment ID
`effect`	Effect name. String.
`verbatim`	Print messages, logical

Value

Numeric value of total variance.

`calculate_trial_variance_count`
Calculate trial variance count

Description

Calculate trial variance count

Usage

```
calculate_trial_variance_count(ald, tid, effect)
```

Arguments

`ald`	Aggregate level data. Data frame in long format.
`tid`	Treatment ID.
`effect`	Effect name. String.

Value

Numeric value of total variance.

`calc_ALD_stats` *Aggregate-level data mean and variance statistics*

Description

Computes the mean and variance of marginal treatment effects for aggregate-level trial data.

Usage

```
calc_ALD_stats(strategy, analysis_params)
```

Arguments

`strategy` A list containing the strategy details, including the family distribution.

`analysis_params`

A list containing:

- `ald` Aggregate-level trial data
- `ref_trt` Treatment labels reference (common; e.g. placebo)
- `ald_comp` Treatment labels comparator
- `scale` A scaling parameter for the calculation. From "log_odds", "risk_difference", "log_relative_risk".

Value

A list containing:

`mean` The marginal treatment effect mean.

`var` The marginal treatment effect variance.

See Also

`marginal_treatment_effect()`, `marginal_variance()`

Examples

```
strategy <- list(family = list(family = "binomial"))  # basic version

ald <- data.frame(trt = c("B","C","B","C"),
                  variable = c(NA, NA, "y", "y"),
                  statistic = c("N", "N", "sum", "sum"),
                  value = c(100, 100, 50, 60))

calc_ALD_stats(strategy = strategy,
               list(ald = ald,
                    ref_trt = "C",
                    ald_comp = "B",
                    scale = "log_odds"))
```

calc_gcomp_bayes *Bayesian G-computation using Stan*

Description

Calculate draws of binary responses from posterior predictive distribution from the Bayesian G-computation method using Hamiltonian Monte Carlo.

Usage

```
calc_gcomp_bayes(strategy, analysis_params, ...)
```

Arguments

`strategy` A list specifying the model strategy, including:

- `formula`: A linear regression `formula` object.
- `family`: A `family` object specifying the distribution and link function (e.g., `binomial`).
- `iter`: Number of iterations for the MCMC sampling.
- `warmup`: Number of warmup iterations for the MCMC sampling.
- `chains`: Number of MCMC chains.

`analysis_params` List of analysis parameters. Must contain `ipd` and `ald`.

`...` Additional arguments passed to `rstanarm::stan_glm()`.

Value

A list containing:

- `means`: A list containing:
  - `A`: Posterior means for comparator treatment group "A".
  - `C`: Posterior means for reference treatment group "C".
- `model`: A list containing the `fit` object (from `stan_glm`), `rho`, `N`, and `stan_args`.

Examples

```
strategy <- list(
  formula = y ~ trt:X1,
  family = binomial(),
  rho = NA,
  N = 1000L,
  marginal_distns = NA,
  marginal_params = NA,
  trt_var = "trt",
  iter = 2000,
  warmup = 500,
  chains = 4)

ipd <- data.frame(
   trt = sample(c("A", "C"), size = 100, replace = TRUE),
   X1 = rnorm(100, 1, 1),
   y = sample(c(1,0), size = 100, prob = c(0.7, 0.3), replace = TRUE))

ald <- data.frame(
  trt = c(NA, NA, "B", "C", "B", "C"),
  variable = c("X1", "X1", "y", "y", NA, NA),
  statistic = c("mean", "sd", "sum", "sum", "N", "N"),
  value = c(0.5, 0.1, 10, 12, 20, 25))

calc_gcomp_bayes(
  strategy,
  analysis_params = list(
    ipd = ipd, ald = ald,
    ref_trt = "C",
    ipd_comp = "A"))
```

`calc_gcomp_ml` *G-computation Maximum Likelihood Bootstrap*

Description

Computes the mean difference in treatment effects using bootstrap resampling.

Usage

```
calc_gcomp_ml(strategy, analysis_params)
```

Arguments

`strategy` A list specifying the model strategy, including:

- `R`: Number of bootstrap replications.
- `formula`: A linear regression `formula` object.

- `family`: A `family` object specifying the distribution and link function (e.g., `binomial`).
- `N`: Synthetic sample size for g-computation.

`analysis_params`
List of analysis parameters.

Value

A list containing:

- `means`: A list containing:
  - `A`: Bootstrap estimates for comparator treatment group "A".
  - `C`: Bootstrap estimates for reference treatment group "C".
- `model`: A list containing the `fit` object, `rho`, and `N`.

Examples

```
strategy <- list(
  formula = y ~ trt:X1,
  family = binomial(),
  rho = NA,
  N = 1000L,
  n_boot = 100L,
  marginal_distns = NA,
  marginal_params = NA,
  trt_var = "trt")

ipd <- data.frame(trt = sample(c("A", "C"), size = 100, replace = TRUE),
                  X1 = rnorm(100, 1, 1),
                  y = sample(c(1,0), size = 100, prob = c(0.7,0.3), replace = TRUE))

ald <- data.frame(trt = c(NA, NA, "B", "C", "B", "C"),
                  variable = c("X1", "X1", "y", "y", NA, NA),
                  statistic = c("mean", "sd", "sum", "sum", "N", "N"),
                  value = c(0.5, 0.1, 10, 12, 20, 25))

calc_gcomp_ml(
  strategy,
  analysis_params =
    list(ipd = ipd, ald = ald,
         ref_trt = "C",
         ipd_comp = "A"))
```

calc_IPD_stats *Calculate individual-level patient data statistics*

Description

Computes mean and variance statistics for individual-level patient data using various approaches, including Matching-Adjusted Indirect Comparison (MAIC), Simulated Treatment Comparison (STC), and G-computation via Maximum Likelihood Estimation (MLE) or Bayesian inference.

Usage

```
calc_IPD_stats(strategy, analysis_params, ...)

## Default S3 method:
calc_IPD_stats(...)

## S3 method for class 'stc'
calc_IPD_stats(strategy, analysis_params, var_method = NULL, ...)

## S3 method for class 'maic'
calc_IPD_stats(strategy, analysis_params, var_method = NULL, ...)

## S3 method for class 'gcomp_ml'
calc_IPD_stats(strategy, analysis_params, var_method = NULL, ...)

## S3 method for class 'gcomp_bayes'
calc_IPD_stats(strategy, analysis_params, var_method = NULL, ...)

## S3 method for class 'mim'
calc_IPD_stats(strategy, analysis_params, var_method = NULL, ...)
```

Arguments

`strategy` A list corresponding to different modelling approaches

`analysis_params`

A list containing:

- `ipd`: Individual-level patient data (data frame)
- `ald`: Aggregate-level trial data (data frame)
- `ref_trt`: Treatment label for the reference arm (common; e.g., "C")
- `ipd_comp`: Treatment label for the comparator arm in the IPD (e.g., "A")
- `scale`: Scaling parameter ("log_odds", "risk_difference", "log_relative_risk")

`...` Additional arguments

`var_method` A string specifying the variance estimation method, either "sample" (default) or "sandwich".

Value

A list containing:

- `contrasts`: A list with elements `mean` and `var`.
- `absolute`: A list with elements `mean` and `var`.

Simulated treatment comparison statistics

IPD for reference "C" and comparator "A" trial arms are used to fit a regression model describing the observed outcomes y in terms of the relevant baseline characteristics x and the treatment variable z.

Matching-adjusted indirect comparison statistics

Marginal IPD comparator treatment "A" vs reference treatment "C" treatment effect estimates using bootstrapping sampling.

G-computation maximum likelihood statistics

Compute a non-parametric bootstrap with default $R = 1000$ resamples.

G-computation Bayesian statistics

Using Stan, compute marginal relative effects for IPD comparator "A" vs reference "C" treatment arms for each MCMC sample by transforming from probability to linear predictor scale.

Multiple imputation marginalisation

Using Stan, compute marginal relative treatment effect for IPD comparator "A" vs reference "C" arms for each MCMC sample by transforming from probability to linear predictor scale. Approximate by using imputation and combining estimates using Rubin's rules.

Examples

```
strategy <- strategy_maic(formula = as.formula(y~trt:X1), family = binomial())
ipd <- data.frame(trt = sample(c("A", "C"), size = 100, replace = TRUE),
                  X1 = rnorm(100, 1, 1),
                  y = sample(c(1,0), size = 100, prob = c(0.7,0.3), replace = TRUE))

ald <- data.frame(trt = c(NA, "B", "C", "B", "C"),
                  variable = c("X1", "y", "y", NA, NA),
                  statistic = c("mean", "sum", "sum", "N", "N"),
                  value = c(0.5, 10, 12, 20, 25))

calc_IPD_stats(strategy,
  analysis_params = list(ipd = ipd, ald = ald, scale = "log_odds"))
```

`estimate_var_sandwich` *Estimate Variance Sandwich Estimator*

Description

Computes the robust (sandwich) variance estimator for the treatment effect.

Usage

```
estimate_var_sandwich(strategy, analysis_params, ...)
```

Arguments

`strategy` An object of class `strategy` created by functions such as `strategy_maic()`, `strategy_stc()`, or `strategy_mim()`. Contains modelling details like the formula and family.

`analysis_params`
List of analysis parameters (ipd, ald, etc.)

`...` Additional arguments

Value

Numeric variance estimate for the treatment contrast

`get_treatment_effect` *Get treatment effect scale corresponding to a link function*

Description

Maps a given link function to its corresponding treatment effect scale.

Usage

```
get_treatment_effect(link)
```

Arguments

`link` A character string specifying the link function. Options are:

"logit" Log-odds scale.
"identity" Risk difference.
"probit" Probit scale.
"cloglog" Log relative risk for rare events.
"log" Log relative risk.

Value

A character string representing the treatment effect scale.

Examples

```
get_treatment_effect(link = "logit")
get_treatment_effect(link = "identity")
```

`marginal_treatment_effect`

Marginal treatment effect from reported event counts

Description

Computes the relative treatment effect from aggregate-level data using event counts.

Usage

```
marginal_treatment_effect(ald, ref_trt = NA, comp_trt = NA, scale, family)
```

Arguments

`ald`	Aggregate-level data
`ref_trt`	Treatment labels reference (common; e.g. placebo)
`comp_trt`	Treatment labels comparator
`scale`	A scaling parameter for the calculation.
`family`	A character string specifying the family distribution (e.g., "binomial").

Value

Numeric relative treatment effect.

Examples

```
ald <- data.frame(trt = c("B","C","B","C"),
                  variable = c(NA, NA, "y", "y"),
                  statistic = c("N", "N", "sum", "sum"),
                  value = c(100, 100, 50, 60))

marginal_treatment_effect(ald, ref_trt = "C", comp_trt = "B",
                          scale = "log_odds", family = "binomial")
```

marginal_variance *Marginal effect variance using the delta method*

Description

Computes the total variance of marginal treatment effects using the delta method.

Usage

```
marginal_variance(ald, ref_trt = NA, comp_trt = NA, scale, family)
```

Arguments

ald	Aggregate-level data
ref_trt	Treatment labels reference (common; e.g. placebo)
comp_trt	Treatment labels comparator
scale	A scaling parameter for the calculation.
family	A character string specifying the family distribution (e.g., "binomial").

Value

Numeric total variance of marginal treatment effects.

Examples

```
ald <- data.frame(trt = c("B","C","B","C"),
                  variable = c(NA, NA, "y", "y"),
                  statistic = c("N", "N", "sum", "sum"),
                  value = c(100, 100, 50, 60))

marginal_variance(ald, ref_trt = "C", comp_trt = "B",
                  scale = "log_odds", family = "binomial")
```

outstandR *Calculate the difference between treatments using all evidence*

Description

This is the main, top-level wrapper for `{outstandR}`. Methods taken from (Remiro-Azócar et al. 2022).

Usage

```
outstandR(
  ipd_trial,
  ald_trial,
  strategy,
  ref_trt = NA,
  CI = 0.95,
  scale = NULL,
  var_method = NULL,
  seed = NULL,
  ...
)
```

Arguments

`ipd_trial`	Individual-level patient data. For example, suppose between studies *A* and *C*. In a long format and must contain a treatment column and outcome column consistent with the formula object. The labels in the treatment are used internally so there must be a common treatment with the aggregate-level data trial.
`ald_trial`	Aggregate-level data. For example, suppose between studies *B* and *C*. The column names are • `variable`: Covariate name. In the case of treatment arm sample size this is `NA`, • `statistic`: Summary statistic name from "mean", standard deviation "sd", probability "prop", or "sum", • `value`: Numerical value of summary statistic, • `trt`: Treatment label. Because we assume a common covariate distribution between treatment arms this is `NA`.
`strategy`	Computation strategy function. These can be `strategy_maic()`, `strategy_stc()`, `strategy_gcomp_ml()` and `strategy_gcomp_bayes()`.
`ref_trt`	Reference / common / anchoring treatment name.
`CI`	Confidence interval level; between 0,1 with default 0.95.
`scale`	Relative treatment effect scale. If `NULL`, the scale is automatically determined from the model. Choose from "log-odds", "log_relative_risk", "risk_difference", "delta_z", "mean_difference", "rate_difference" depending on the data type.
`var_method`	Variance estimation method.
`seed`	Random seed.
`...`	Additional arguments. Currently, can pass named arguments to `rstanarm::stan_glm()` via `strategy_gcomp_bayes()`.

Value

List of length 11 of statistics as a `outstandR` class object. Containing statistics between each pair of treatments. These are the mean, variances and confidence intervals, for contrasts and absolute values.

References

Remiro-Azócar A, Heath A, Baio G (2022). “Parametric G-computation for compatible indirect treatment comparisons with limited individual patient data.” *Res. Synth. Methods*, 1–31. ISSN 1759-2879, doi:10.1002/jrsm.1565, 2108.12208.

See Also

`strategy_maic()` `strategy_stc()` `strategy_gcomp_ml()` `strategy_gcomp_bayes()`

Examples

```
data(AC_IPD_binY_contX)  # A vs C individual patient-level data
data(BC_ALD_binY_contX)  # B vs C aggregate-level data

# linear formula
lin_form <- as.formula("y ~ PF_cont_1 + PF_cont_2 + trt*EM_cont_1 + trt*EM_cont_2")

# sampling values of additional arguments picked for speed
# select appropriate to specific analysis

# matching-adjusted indirect comparison
outstandR_maic <- outstandR(
  AC_IPD_binY_contX, BC_ALD_binY_contX,
  strategy = strategy_maic(formula = lin_form, n_boot = 100))

# simulated treatment comparison
outstandR_stc <- outstandR(
  AC_IPD_binY_contX, BC_ALD_binY_contX,
  strategy = strategy_stc(lin_form))


# G-computation with maximum likelihood
outstandR_gcomp_ml <- outstandR(
  AC_IPD_binY_contX, BC_ALD_binY_contX,
  strategy = strategy_gcomp_ml(lin_form, n_boot = 100, N =100))

# G-computation with Bayesian inference
outstandR_gcomp_bayes <- outstandR(
  AC_IPD_binY_contX, BC_ALD_binY_contX,
  strategy = strategy_gcomp_bayes(lin_form),
  chains = 1, iter = 1000, warmup = 20)

# Multiple imputation marginalization
outstandR_mim <- outstandR(
  AC_IPD_binY_contX, BC_ALD_binY_contX,
  strategy = strategy_mim(lin_form,
                          N = 100), # size of pseudo-population
  chains = 1, iter = 1000, warmup = 20)
```

outstandR-class *outstandR class*

Description

The `outstandR` class contains the results from running a model with the function `outstandR()`.

Details

Objects of class `outstandR` have the following

contrasts A list containing statistics for relative treatment effects:

- `means`: Estimated relative effects (e.g., log-odds ratios, risk differences).
- `variances`: Variance-covariance matrix of the relative effects.
- `contrast_ci`: Confidence intervals for the relative effects.

absolute A list containing statistics for absolute treatment outcomes:

- `means`: Estimated absolute outcomes (e.g., probabilities, mean response).
- `variances`: Variance-covariance matrix of the absolute outcomes.
- `ci`: Confidence intervals for the absolute outcomes.

CI The confidence level used (e.g., 0.95).

ref_trt The name of the reference treatment.

scale The scale of the outcome (e.g., "log odds", "probability").

model A list containing details of the underlying statistical model. Contents vary by strategy:

- `family`: The error distribution and link function.
- `fit`: The underlying model object (e.g., for STC, G-Comp ML, or Bayesian G-Comp).
- `weights`, `ESS`: (MAIC only) The estimated weights and Effective Sample Size.
- `stan_args`: (Bayesian G-Comp, MIM) Arguments passed to Stan.
- `rho`: (G-Comp ML, MIM, Bayesian G-Comp) Correlation coefficient.
- `N`: (G-Comp ML, MIM, Bayesian G-Comp) Number of iterations.
- `nu`, `hats.v`, `M`: (MIM only) Imputation parameters and matrices.

plot.outstandR *Default Plot Method for outstandR Objects*

Description

Default Plot Method for outstandR Objects

Usage

```
## S3 method for class 'outstandR'
plot(x, ..., type = c("both", "contrasts", "absolute"), labels = NULL)
```

Arguments

`x`	An object of class 'outstandR' or a list of 'outstandR' objects.
`...`	Additional 'outstandR' objects for comparison.
`type`	Character, one of "both" (default), "contrasts", or "absolute".
`labels`	Optional character vector of names for the models.

Value

A `ggplot2::ggplot()` object representing the forest plot of the results.

`print.outstandR` *Print a Summary of a outstandR Object*

Description

This is a method for the function `print()` for objects of the class "outstandR" created by a call to `outstandR()`

Usage

```
## S3 method for class 'outstandR'
print(x, ...)
```

Arguments

`x`	Objects of the class "outstandR"
`...`	Additional arguments passed to other methods

Value

No return value, called for side effects

See Also

`outstandR()`

reshape_ald_to_long *Convert aggregate data from wide to long format*

Description

Convert aggregate data from wide to long format

Usage

```
reshape_ald_to_long(df)
```

Arguments

df	A dataframe of ALD

Value

Data frame in long format

reshape_ald_to_wide *Convert aggregate data from long to wide format*

Description

Convert aggregate data from long to wide format

Usage

```
reshape_ald_to_wide(df)
```

Arguments

df	A Dataframe of ALD

Value

Data frame in wide format

Examples

```
df <-
  data.frame(
    variable = c("age", "age", "y", "y", "y", "y", "y", "y", "y", "y"),
    statistic = c("mean", "sd", "sum", "bar", "sd", "N", "sum", "bar", "sd", "N"),
    trt = c(NA, NA, "B", "B", "B", "B", "C", "C", "C", "C"),
    value = c(1,1,1,1,1,1,1,1,1,1))
```

strategy-class *Strategy class and subclasses*

Description

The strategy class is a virtual class that defines the statistical approach for population adjustment in indirect treatment comparisons These objects are constructors that validate hyperparameters and encapsulate modelling settings before execution by outstandR()

Details

Objects of class strategy have a common structure but carry different subclasses to trigger specific S3 method dispatch

General fields Shared by all strategies:

- formula: The linear regression formula for the outcome model
- family: A base R family object specifying the distribution and link
- trt_var: The name of the treatment variable.

maic subclass Additional fields for Matching-Adjusted Indirect Comparison:

- n_boot: Number of bootstrap resamples for variance estimation.

stc subclass Additional fields for Simulated Treatment Comparison:

- N: Synthetic sample size for the target population.

gcomp_ml subclass Additional fields for Maximum Likelihood G-computation:

- rho: Named square matrix of covariate correlations.
- marginal_distns: Names of the marginal distributions for covariates.
- marginal_params: Parameters for the marginal distributions.
- N: Synthetic sample size for the pseudo-population.
- n_boot: Number of bootstrap resamples.

gcomp_bayes subclass Additional fields for Bayesian G-computation:

- rho, marginal_distns, marginal_params, N: Same as gcomp_ml.
- ...: Additional arguments passed to the Stan engine via rstanarm::stan_glm().

mim subclass Additional fields for Multiple Imputation Marginalization:

- rho: Correlation matrix.
- N: Number of iterations/simulated individuals.

strategy_maic *New strategy objects*

Description

Create a type of strategy class for each modelling approach.

Usage

```
strategy_maic(
  formula = NULL,
  family = gaussian(link = "identity"),
  trt_var = NULL,
  n_boot = 1000L
)

strategy_stc(
  formula = NULL,
  family = gaussian(link = "identity"),
  trt_var = NULL
)

strategy_gcomp_ml(
  formula = NULL,
  family = gaussian(link = "identity"),
  trt_var = NULL,
  rho = NA,
  marginal_distns = NA,
  marginal_params = NA,
  n_boot = 1000L,
  N = 1000L
)

strategy_gcomp_bayes(
  formula = NULL,
  family = gaussian(link = "identity"),
  trt_var = NULL,
  rho = NA,
  marginal_distns = NA,
  marginal_params = NA,
  N = 1000L
)

strategy_mim(
  formula = NULL,
  family = gaussian(link = "identity"),
  trt_var = NULL,
  rho = NA,
  N = 1000L
)

new_strategy(strategy, ...)
```

Arguments

formula Linear regression formula object. Prognostic factors (PF) are main effects and effect modifiers (EM) are interactions with the treatment variable, e.g., y ~ X1 + trt + trt:X2. For covariates as both PF and EM use * syntax.

family A 'family' object specifying the distribution and link function (e.g., 'binomial'). See stats::family() for more details.

trt_var Treatment variable name; string

n_boot The number of resamples used for the non-parametric bootstrap; integer

rho A named square matrix of covariate correlations; default NA

marginal_distns

Marginal distributions names; vector default NA. Available distributions are given in stats::Distributions. See copula::Mvdc() for details

marginal_params

Marginal distributions parameters; list of lists, default NA. See copula::Mvdc() for details

N Synthetic sample size for g-computation

strategy Class name from strategy_maic, strategy_stc, strategy_gcomp_ml, strategy_gcomp_bayes, strategy_mim

... Additional arguments

Value

maic class object

stc class object

gcomp_ml class object

gcomp_bayes class object

mim class object

Strategy list object

Matching-adjusted indirect comparison (MAIC)

MAIC is a form of non-parametric likelihood reweighting method which allows the propensity score logistic regression model to be estimated without IPD in the *AC* population. The mean outcomes $\mu_{t(AC)}$ on treatment $t = A, B$ in the *AC* target population are estimated by taking a weighted average of the outcomes Y of the N individuals in arm t of the *AB* population.

Used to compare marginal treatment effects where there are cross-trial differences in effect modifiers and limited patient-level data.

$$\hat{Y} = \frac{\sum_{i=1}^{N} Y_{it(AB)} w_{it}}{\sum_{i=1}^{N} w_{it}}$$

where the weight w_{it} assigned to the i-th individual receiving treatment t is equal to the odds of being enrolled in the *AC* trial vs the *AB* trial.

Simulated treatment comparison (STC)

Outcome regression-based method which targets a conditional treatment effect. STC is a modification of the covariate adjustment method. An outcome model is fitted using IPD in the *AB* trial. For example,

$$g(\mu_{t(AB)}(X)) = \beta_0 + \beta_1^T X + (\beta_B + \beta_2^T X^{EM}) I(t = B)$$

where β_0 is an intercept term, β_1 is a vector of coefficients for prognostic variables, β_B is the relative effect of treatment *B* compared to *A* at $X = 0$, β_2 is a vector of coefficients for effect modifiers X^{EM} subvector of the full covariate vector X), and $\mu_{t(AB)}(X)$ is the expected outcome of an individual assigned treatment t with covariate values X which is transformed onto a chosen linear predictor scale with link function $g(\cdot)$.

G-computation maximum likelihood

G-computation marginalizes the conditional estimates by separating the regression modelling from the estimation of the marginal treatment effect for *A* versus *C*. For example, a regression model of the observed outcome y on the covariates x and treatment z is fitted to the *AC* IPD:

$$g(\mu_n) = \beta_0 + \boldsymbol{x}_n \boldsymbol{\beta_1} + (\beta_z + \boldsymbol{x_n^{EM}} \boldsymbol{\beta_2}) \mathrm{I}(z_n = 1)$$

In the context of G-computation, this regression model is called the “Q-model". Having fitted the Q-model, the regression coefficients are treated as nuisance parameters. The parameters are applied to the simulated covariates $x*$ to predict hypothetical outcomes for each subject under both possible treatments. Namely, a pair of predicted outcomes, also called potential outcomes, under *A* and under *C*, is generated for each subject.

By plugging treatment *C* into the regression fit for every simulated observation, we predict the marginal outcome mean in the hypothetical scenario in which all units are under treatment *C*:

$$\hat{\mu}_0 = \int_{x^*} g^{-1}(\hat{\beta}_0 + x^* \hat{\beta}_1) p(x^*) dx^*$$

To estimate the marginal or population-average treatment effect for *A* versus *C* in the linear predictor scale, one back-transforms to this scale the average predictions, taken over all subjects on the natural outcome scale, and calculates the difference between the average linear predictions:

$$\hat{\Delta}_{10}^{(2)} = g(\hat{\mu}_1) - g(\hat{\mu}_0)$$

G-computation Bayesian

The difference between Bayesian G-computation and its maximum-likelihood counterpart is in the estimated distribution of the predicted outcomes. The Bayesian approach also marginalizes, integrates or standardizes over the joint posterior distribution of the conditional nuisance parameters of the outcome regression, as well as the joint covariate distribution.

Draw a vector of size $N*$ of predicted outcomes $y * z$ under each set intervention $z* \in \{0, 1\}$ from its posterior predictive distribution under the specific treatment. This is defined as $p(y *_{z*} |\mathcal{D}_{AC}) = \int_\beta p(y *_{z*} |\beta)p(\beta|\mathcal{D}_{AC})d\beta$ where $p(\beta|\mathcal{D}_{AC})$ is the posterior distribution of the outcome regression coefficients β, which encode the predictor-outcome relationships observed in the *AC* trial IPD.

This is given by:

$$p(y*_{z*} \mid \mathcal{D}_{AC}) = \int_{x*} p(y* \mid z*, x*, \mathcal{D}_{AC})p(x* \mid \mathcal{D}_{AC})dx*$$

$$= \int_{x*} \int_\beta p(y* \mid z*, x*, \beta)p(x* \mid \beta)p(\beta \mid \mathcal{D}_{AC})d\beta dx*$$

In practice, the integrals above can be approximated numerically, using full Bayesian estimation via Markov chain Monte Carlo (MCMC) sampling.

Multiple imputation marginalization (MIM)

TODO

Note

While current implementations focus on binary, continuous, and count outcomes, support for survival data (using the `survival` package) is under active development and scheduled for version 1.1.0.

See Also

`strategy_gcomp_bayes()`

`strategy_gcomp_ml()`,`copula::Mvdc()`

`summary.outstandR` *Summary method for outstandR*

Description

Summary method for outstandR

Usage

```
## S3 method for class 'outstandR'
summary(object, CI = NA, ...)

## S3 method for class 'summary.outstandR'
print(x, digits = 3, ...)
```

Arguments

`object`	`outstandR()` output object.
`CI`	Confidence interval level.
`...`	Additional arguments.
`x`	An object used to select a method.
`digits`	Minimal number of significant digits, see `print.default`.

Value

List of class `summary.outstandR`

Original argument, but mainly called for side effects

Index