# Package ‘RobinCID’

May 29, 2025

**Title** Robust Inference in Complex Innovative Trial Design

**Version** 1.0.0

**Description** Perform robust estimation and inference in platform trials and other master protocol trials. Yuhan Qian, Yifan Yi, Jun Shao, Yanyao Yi, Gregory Levin, Nicole Mayer-Hamblett, Patrick J. Heagerty, Ting Ye (2025) <doi:10.48550/arXiv.2411.12944>.

**License** MIT + file LICENSE

**Encoding** UTF-8

**RoxygenNote** 7.3.2

**URL** `https://github.com/Eureeca/RobinCID`

**BugReports** `https://github.com/Eureeca/RobinCID/issues`

**Suggests** testthat (>= 3.0.0)

**Config/testthat/edition** 3

**Imports** checkmate, numDeriv, stats, MASS

**Depends** R (>= 3.5)

**LazyData** true

**NeedsCompilation** no

**Author** Yuhan Qian [aut, cre, cph],
Ting Ye [aut],
Yanyao Yi [aut],
Marlena Bannick [aut]

**Maintainer** Yuhan Qian <`yhqian@uw.edu`>

**Repository** CRAN

**Date/Publication** 2025-05-29 08:50:13 UTC

## Contents

---

## `RobinCID-package` `RobinCID` *Package*

### Description

`RobinCID` implements unbiased prediction and robust inference in R.

### Author(s)

**Maintainer**: Yuhan Qian <`yhqian@uw.edu`> [copyright holder]

Authors:

- Ting Ye
- Yanyao Yi
- Marlena Bannick

### See Also

Useful links:

- `https://github.com/Eureeca/RobinCID`
- Report bugs at `https://github.com/Eureeca/RobinCID/issues`

---

## `assign_prob_and_strata` *Assign Probability according to Design*

### Description

Assign Probability according to Design

### Usage

```
assign_prob_and_strata(
  data,
  estimand,
  design = list(randomization_var_colnames = NULL, randomization_table = NULL),
  method,
  estimated_propensity = TRUE,
  stratify_by = NULL
)
```

### Arguments

| | |
|---|---|
| `data` | (`data.frame`) Input data frame. |
| `estimand` | (`list`) A list specifying the estimand. |
| `design` | (`list`) A list describing the randomization design. See `Details`. |
| `method` | estimation method. |
| `estimated_propensity` | Whether to use estimated propensity score. |
| `stratify_by` | The column name of stratification variable in `data`. |

### Details

`design` has two elements: `randomization_var_colnames` (`vector`) and `randomization_table` (`data.frame`)

### Value

A new `data` with columns of the treatment assignment probability.

---

## `estimate_effect` *Compute Estimates and Covariance Matrix*

### Description

Compute Estimates and Covariance Matrix

### Usage

```
estimate_effect(
  ret,
  y,
  treatment,
  treatments_for_compare,
  data,
  prob_mat,
  post_strata,
  stabilize
)
```

### Arguments

| | |
|---|---|
| `ret` | counterfactual prediction |
| `y` | Observed outcome |
| `treatment` | name of treatment |
| `treatments_for_compare` | description |
| `data` | (`data.frame`) data |
| `prob_mat` | (`data.frame`) treatment assignment probability |
| `post_strata` | (`character`) A string name of post-stratification variable |
| `stabilize` | (`logical`) whether to stabilize |

### Value

A list of "estimate_effect" object with following elements:

- `estimate`: estimate of the treatment effect.
- `inner_variance`: estimate of the covariance matrix.
- `method`: estimation method.

---

## `example` *Trial Data*

### Description

This dataset is generated based on the first three enrollment windows outlined in Figure 1(a) in our paper.

### Usage

```
example
```

### Format

A data frame with 500 rows and 17 columns:

**xc, xb, subtype** The covariates.

**t, substudy** The enrollment window and substudy assignment

**treatment** The treatment assignment, "1", "2", "3" and "4"

**y** The continuous response.

**y_b** The binary response, I(y>3).

**s12, s12.2, s12.error, s13, s14** The stratification variables.

**trt.1, trt.2, trt.3, trt.4** The assgiment probabilities of trt 1, 2, 3, and 4.

### Source

The data is generated by Yuhan.

---

`find_data` *Find Data in a Fit*

---

### Description

Find Data in a Fit

### Usage

```
find_data(fit, ...)
```

### Arguments

| | |
|---|---|
| `fit` | A fit object. |
| `...` | Additional arguments. |

### Value

A data frame used in the fit.

---

`h_diff` *Contrast Functions and Jacobians*

---

### Description

Contrast Functions and Jacobians

### Usage

```
h_diff(x)

h_jac_diff(x)

h_ratio(x)

h_jac_ratio(x)

h_odds_ratio(x)

h_jac_odds_ratio(x)
```

### Arguments

| | |
|---|---|
| `x` | (`numeric`) Vector of values. |

## Value

Vector of contrasts, or matrix of jacobians.

## Examples

```
h_diff(1:3)
h_jac_ratio(1:3)
```

---

# `predict_counterfactual` *Counterfactual Prediction*

## Description

Obtain counterfactual prediction of a fit.

## Usage

```
predict_counterfactual(
  fit.j,
  fit.k,
  treatment,
  treatments_for_compare,
  prob_mat,
  post_strata,
  data,
  stabilize,
  settings
)
```

## Arguments

| | |
|---|---|
| `fit.j` | fitted object for trt j. |
| `fit.k` | fitted object for trt k. |
| `treatment` | name of treatment column |
| `treatments_for_compare` | (`character`) Treatments for comparison |
| `prob_mat` | (`data.frame`) treatment assignment probabilities |
| `post_strata` | (`character`) A string name of post-stratification variable. |
| `data` | (`data.frame`) raw dataset. |
| `stabilize` | stabilize |
| `settings` | estimation setting |

### Value

A list of `prediction_cf` object with following elements:

- `estimation`: = estimation,
- `sample_size`: sample size of ECE population.
- `fit.j`: fitted model for the treatment specified in tx_to_compare.
- `fit.k`: fitted model for the treatment specified in tx_to_compare.

---

## `robin_ps` *Post-Stratification Based Inference*

### Description

Provides robust inference via post stratification.

### Usage

```
robin_ps(
  data,
  estimand = list(tx_colname = NULL, tx_to_compare = NULL),
  design = list(randomization_var_colnames = NULL, randomization_table = NULL),
  stratify_by = NULL,
  outcome_model = list(formula = NULL, family = gaussian()),
  contrast_specs = list(contrast = "difference", contrast_jac = NULL),
  alpha = 0.05,
  ...
)
```

### Arguments

| | |
|---|---|
| `data` | (`data.frame`) A data frame containing the dataset. |
| `estimand` | (`list`) A list specifying the estimand, with two elements:<br>• `tx_colname` (`character`): The column name of the treatment variable in `data`.<br>• `tx_to_compare` (`character vector`): A vector specifying exactly two treatment levels to compare. |
| `design` | (`list`) A list specifying randomization information, with two elements:<br>• `randomization_var_colnames` (`character vector`): Column names of randomization variables in `data`.<br>• `randomization_table` (`data.frame`, default: `NULL`): A data frame containing treatment assignment probabilities for each level of the randomization variables. See *Details*. |
| `stratify_by` | (`character`, optional) The column name of the stratification variable in `data`. If provided, `stratify_by` overrides `design`. |

| | |
|---|---|
| `outcome_model` | (`list`) A list specifying the outcome working model, with two elements:<br>- `formula` (`formula`): The regression formula for the analysis.<br>- `family` A description of the error distribution and link function for the model. Default: `gaussian()`. |
| `contrast_specs` | (`list`) A list specifying the contrast function and its Jacobian:<br>- `contrast` (`function` or `character`): A function to compute the treatment effect, or one of `"difference"`, `"risk_ratio"`, or `"odds_ratio"` for default contrasts.<br>- `contrast_jac` (`function`, optional): A function to compute the Jacobian of the contrast function. Ignored if using default contrasts. |
| `alpha` | (`numeric`) The nominal significance level. Default: `0.05`. |
| `...` | Additional arguments passed to `glm`. |

### Details

If family is `MASS::negative.binomial(NA)`, the function will use `MASS::glm.nb` instead of `glm`.

### Value

A treatment_effect object.

### Examples

```
data_sim <- RobinCID::example
tx_colname <- "treatment"
treatment_levels <- unique(data_sim[[tx_colname]])
tx_to_compare <- c("trt.1", "trt.3")
randomization_var_colnames <- c("t", "subtype")
df <- data_sim[c("xb", "xc", tx_colname, randomization_var_colnames, "y")]
randomization_table <- unique(data_sim[c(randomization_var_colnames, treatment_levels)])
robin_ps(
  data = df,
  estimand = list(tx_colname = tx_colname,
                  tx_to_compare = tx_to_compare),
  design = list(randomization_var_colnames = randomization_var_colnames,
                randomization_table = randomization_table),
  stratify_by = NULL,
  outcome_model = list(formula = y ~ 1,
                       family = gaussian())
)
```

---

## `robin_wt` *Inverse Probability Weighting Based Inference*

### Description

Provides robust inference via inverse probability weighting.

## Usage

```
robin_wt(
  data,
  estimand = list(tx_colname = NULL, tx_to_compare = NULL),
  design = list(randomization_var_colnames = NULL, randomization_table = NULL),
  estimated_propensity = TRUE,
  outcome_model = list(formula = NULL, family = gaussian()),
  contrast_specs = list(contrast = "difference", contrast_jac = NULL),
  alpha = 0.05,
  ...
)
```

## Arguments

| | |
|---|---|
| `data` | (`data.frame`) A data frame containing the dataset. |
| `estimand` | (`list`) A list specifying the estimand, with two elements:<br>• `tx_colname` (`character`): The column name of the treatment variable in `data`.<br>• `tx_to_compare` (`character vector`): A vector specifying exactly two treatment levels to compare. |
| `design` | (`list`) A list specifying randomization information, with two elements:<br>• `randomization_var_colnames` (`character vector`): Column names of randomization variables in `data`.<br>• `randomization_table` (`data.frame`, default: `NULL`): A data frame containing treatment assignment probabilities for each level of the randomization variables. See *Details*. |
| `estimated_propensity` | (`logical`, default: `TRUE`) Whether to use estimated propensity scores. |
| `outcome_model` | (`list`) A list specifying the outcome working model, with two elements:<br>• `formula` (`formula`): The regression formula for the analysis.<br>• `family` A description of the error distribution and link function for the model. Default: `gaussian()`. |
| `contrast_specs` | (`list`) A list specifying the contrast function and its Jacobian:<br>• `contrast` (`function` or `character`): A function to compute the treatment effect, or one of `"difference"`, `"risk_ratio"`, or `"odds_ratio"` for default contrasts.<br>• `contrast_jac` (`function`, optional): A function to compute the Jacobian of the contrast function. Ignored if using default contrasts. |
| `alpha` | (`numeric`) The nominal significance level. Default: `0.05`. |
| `...` | Additional arguments passed to `glm`. |

## Details

If `randomization_table` is provided, it must include columns corresponding to `randomization_var_colnames`, as well as treatment assignment probability columns named after the treatment levels in `tx_colname` from `data`.

If `family` is `MASS::negative.binomial(NA)`, the function will use `MASS::glm.nb` instead of `glm`.

### Value

A treatment_effect object.

### Examples

```
data_sim <- RobinCID::example
tx_colname <- "treatment"
treatment_levels <- unique(data_sim[[tx_colname]])
tx_to_compare <- c("trt.1", "trt.3")
randomization_var_colnames <- c("t", "subtype")
df <- data_sim[c("xb", "xc", tx_colname, randomization_var_colnames, "y")]
randomization_table <- unique(data_sim[c(randomization_var_colnames, treatment_levels)])
robin_wt(
  data = df,
  estimand = list(tx_colname = tx_colname,
                  tx_to_compare = tx_to_compare),
  design = list(randomization_var_colnames = randomization_var_colnames,
                randomization_table = randomization_table),
  estimated_propensity = FALSE,
  outcome_model = list(formula = y ~ 1,
                       family = gaussian())
)
```

---

## `treatment_effect` *Treatment Effect*

### Description

Obtain treatment effect and variance from counter-factual prediction

### Usage

```
treatment_effect(object, pair, eff_measure, eff_jacobian, alpha, ...)

difference(object, ...)

risk_ratio(object, ...)

odds_ratio(object, ...)
```

### Arguments

| | |
|---|---|
| `object` | Object from which to obtain treatment effect. Must be obtained from `estimate_effect()`. |
| `pair` | (`integer` or `character`) Names or index of the treatment levels. |
| `eff_measure` | (`function`) Treatment effect measurement function. |

| | |
|---|---|
| `eff_jacobian` | (`function`) Treatment effect jacobian function. |
| `alpha` | Nominal level |
| `...` | Additional arguments passed to `glm` |

## Value

A list of `treatment_effect` object with following elements:

- `mm_name`: name of the treatments to compare.
- `marginal_mean`: estimate of the treatment effect.
- `mmvariance`: estimate of the covariance matrix.
- `trt_effect`: estimate of the contrast.
- `variance`: estimate of the variance of contrast.
- `contrast`: name of the contrast function.
- `settings`: estimation settings.

# Index