

# Package ‘inference’

July 22, 2025

**Type** Package

**Title** Methods for Causal Inference with Interference

**Version** 1.0.3

**Date** 2025-05-14

**Maintainer** Bradley Saul <bradleysaul@fastmail.com>

**Description** Provides methods for estimating causal effects in the presence of interference described in B. Saul and M. Hugdens (2017) <[doi:10.18637/jss.v082.i02](https://doi.org/10.18637/jss.v082.i02)>. Currently it implements the inverse-probability weighted (IPW) estimators proposed by E.J. Tchetgen Tchetgen and T.J. Vanderweele (2012) <[doi:10.1177/0962280210386779](https://doi.org/10.1177/0962280210386779)>.

**Depends** R (>= 3.1)

**Imports** numDeriv (>= 2012.9-1), lme4 (>= 1.1-6), Formula (>= 1.1-2),  
methods, spaMM

**License** GPL (>= 2)

**Suggests** testthat, knitr, rmarkdown

**VignetteBuilder** knitr

**LazyData** true

**RoxygenNote** 7.3.2

**Encoding** UTF-8

**NeedsCompilation** no

**Author** Bradley Saul [aut, cre]

**Repository** CRAN

**Date/Publication** 2025-05-15 07:40:02 UTC

## Contents

|                            |   |
|----------------------------|---|
| diagnose_weights . . . . . | 2 |
| direct_effect . . . . .    | 3 |
| get_args . . . . .         | 3 |
| indirect_effect . . . . .  | 4 |
| inference . . . . .        | 4 |

|                              |    |
|------------------------------|----|
| interference . . . . .       | 5  |
| logit_integrand . . . . .    | 7  |
| log_likelihood . . . . .     | 8  |
| overall_effect . . . . .     | 9  |
| print.interference . . . . . | 9  |
| score_calc . . . . .         | 10 |
| score_matrix . . . . .       | 10 |
| total_effect . . . . .       | 11 |
| vaccinesim . . . . .         | 11 |
| voters . . . . .             | 12 |
| wght_calc . . . . .          | 13 |
| wght_deriv_array . . . . .   | 14 |
| wght_deriv_calc . . . . .    | 15 |
| wght_matrix . . . . .        | 15 |

|              |           |
|--------------|-----------|
| <b>Index</b> | <b>17</b> |
|--------------|-----------|

---

|                  |   |
|------------------|---|
| diagnose_weights | <i>Plot histograms of weights from an interference object</i> |
|------------------|---|

---

## Description

Plot histograms of weights from an interference object

## Usage

```
diagnose_weights(obj, allocations = NULL, ...)
```

## Arguments

|             |   |
|-------------|---|
| obj         | an interference object  |
| allocations | optional numeric vector of allocations for which to print histogram. If NULL (the default), five allocations selected evenly from the first allocation to the last are printed. |
| ...         | additional arguments passed to <a href="#">hist</a>   |

## Value

histogram of group-level weights

---

|               |   |
|---------------|---|
| direct_effect | <i>Retrieve Direct Effect estimates</i> |
|---------------|---|

---

**Description**

Retrieves the population average direct causal effect for a specified allocation:  $\hat{Y}(0, \alpha) - \hat{Y}(1, \alpha)$ .

**Usage**

```
direct_effect(object, allocation = NULL, trt.lv11 = 0)
```

**Arguments**

|            |   |
|------------|---|
| object     | an object of class interference   |
| allocation | the allocation scheme for which to estimate direct effects. If NULL, then returns all direct effects. |
| trt.lv11   | Defaults to 0.  |

**Value**

a data.frame with requested values

---

|          |                                      |
|----------|--------------------------------------|
| get_args | <i>Get arguments from a function</i> |
|----------|--------------------------------------|

---

**Description**

Extracts the names of the arguments from a function, and creates a list of those arguments where they exist in ... .

**Usage**

```
get_args(FUN, args_list = NULL, ...)
```

**Arguments**

|           |   |
|-----------|---|
| FUN       | function for which to find arguments  |
| args_list | a list of arguments. Defaults to NULL.                                      |
| ...       | any arguments. Those necessary for FUN must be named as appropriate for FUN |

**Value**

list of arguments for FUN

**Examples**

```
myargs <- get_args(lm, formula = Sepal.Length ~ Sepal.Width, data = iris )
summary(do.call('lm', myargs))
```

---

|                 |   |
|-----------------|---|
| indirect_effect | <i>Retrieve Indirect Effect estimates</i> |
|-----------------|---|

---

**Description**

Retrieves the population average indirect causal effect for specified allocations:  $\hat{Y}(0, \alpha_1) - \hat{Y}(0, \alpha_2)$ . This is the effect due to the coverage (allocation) levels.

**Usage**

```
indirect_effect(object, allocation1, allocation2 = NULL, trt.lvl = 0)
ie(object, allocation1, allocation2 = NULL, trt.lvl = 0)
```

**Arguments**

|             |   |
|-------------|---|
| object      | an object of class interference   |
| allocation1 | the allocation scheme for which to estimate indirect effects  |
| allocation2 | the allocation scheme for which to estimate indirect effects. If NULL, then returns all indirect effects compared to allocation1. |
| trt.lvl     | Defaults to 0.  |

**Value**

a data.frame with requested values

---

|           |   |
|-----------|---|
| inference | <i>Methods for causal inference with interference</i> |
|-----------|---|

---

**Description**

Interference occurs when the treatment of one unit affects outcomes of other units. This package provides methods for estimating causal effects in the presence of interference. Currently it implements the IPW estimators proposed by Tchetgen Tchetgen and Vanderweele (2012) ([doi:10.1177/0962280210386779](https://doi.org/10.1177/0962280210386779)) and developed further in Heydrich-Perez et al. (2014) ([doi:10.1111/biom.12184](https://doi.org/10.1111/biom.12184)).

**Author(s)**

**Maintainer:** Bradley Saul <[bradleysaul@fastmail.com](mailto:bradleysaul@fastmail.com)>

## References

Saul, B. and Huggens, M. G. (2017). A Recipe for inference: Start with Causal Inference. Add Interference. Mix Well with R. *Journal of Statistical Software*, 82(2), 1-21. doi:10.18637/jss.v082.i02

---

 interference

*Estimate Causal Effects in presence of interference*


---

## Description

Estimate Causal Effects in presence of interference

## Usage

```
interference(
  formula,
  propensity_integrand = "logit_integrand",
  loglikelihood_integrand = propensity_integrand,
  allocations,
  data,
  model_method = "glmer",
  model_options = list(family = stats::binomial(link = "logit")),
  causal_estimation_method = "ipw",
  causal_estimation_options = list(variance_estimation = "robust"),
  conf.level = 0.95,
  rescale.factor = 1,
  integrate_allocation = TRUE,
  runSilent = TRUE,
  ...
)
```

## Arguments

**formula** The formula used to define the causal model. Has a minimum of 4 parts, separated by | and ~ in a specific structure: outcome | exposure ~ propensity covariates | group. The order matters, and the pipes split the data frame into corresponding pieces. The part separated by ~ is passed to the chosen model\_method used to estimate or fix propensity parameters.

**propensity\_integrand** A function, which may be created by the user, used to compute the IP weights. This defaults to logit\_integrand, which calculates the product of inverse logits for individuals in a group:  $\prod_{j=1}^{n_i} \{r \times h_{ij}(b_i)^{A_{ij}}\} \{1 - r \times h_{ij}(b_i)\}^{1 - A_{ij}} f_b(b_i; \theta_s)$  where

$$h_{ij}(b_i) = \text{logit}^{-1}(\mathbf{X}_{ij}\theta_a + b_i)$$

and  $b_i$  is a group-level random effect,  $f_b$  is a  $N(0, \theta_s)$  density, and  $r$  is a known randomization probability which may be useful if a participation vector is included in the formula. If no random effect was included in the formula,

|                           |  |
|---------------------------|--|
|                           | logit_integrand essentially ignores the random effect and $f_b(b_i, \theta_s)$ integrates to 1. See details for arguments that can be passed to logit_integrand  |
| loglikelihood_integrand   | A function, which may be created by the user, that defines the log likelihood of the logit model used for robust variance estimation. Generally, this will be the same function as propensity_integrand. Indeed, this is the default.  |
| allocations               | a vector of values in (0, 1). Increasing the number of elements of the allocation vector greatly increases computation time; however, a larger number of allocations will make plots look nicer. A minimum of two allocations is required.   |
| data                      | the analysis data frame. This must include all the variables defined in the formula.   |
| model_method              | the method used to estimate or set the propensity model parameters. Must be one of 'glm', 'glmer', 'HLfit', or 'oracle'. Defaults to 'glmer'. For a fixed effects only model use 'glm', and to include random effects use 'glmer' or 'HLfit'. logit_integrand only supports a single random effect for the grouping variable, so if more random effects are included in the model, different propensity_integrand and loglikelihood_integrand functions should be defined. When the propensity parameters are known (as in simulations) or if estimating parameters by other methods, use the 'oracle' option. See model_options for details on how to pass the oracle parameters. |
| model_options             | a list of options passed to the function in model_method. Defaults to list(family = binomial(link = 'logit')). When model_method = 'oracle', the list must have two elements (1) fixed_effects and (2) random_effects. If the model did not include random effects, set random_effects = NULL.   |
| causal_estimation_method  | currently only supports 'ipw'.   |
| causal_estimation_options | A list. Current options are: (1) variance_estimation is either 'naive' or 'robust'. See details. Defaults to 'robust'.   |
| conf.level                | level for confidence intervals. Defaults to 0.95.  |
| rescale.factor            | a scalar multiplication factor by which to rescale outcomes and effects. Defaults to 1.  |
| integrate_allocation      | Indicator of whether the integrand function uses the allocation parameter. Defaults to TRUE.   |
| runSilent                 | if FALSE, status of computations are printed to console. Defaults to TRUE.   |
| ...                       | Used to pass additional arguments to internal functions such as numDeriv::grad() or integrate(). Additionally, arguments can be passed to the propensity_integrand and loglikelihood_integrand functions.  |

## Details

The following formula includes a random effect for the group: `outcome | exposure ~ propensity covariates + (1|group) | group`. In this instance, the group variable appears twice. If the study design includes a "participation" variable, this is easily added to the formula: `outcome | exposure | participation ~ propensity covariates | group`.

logit\_integrand has two options that can be passed via the ... argument:

- `randomization`: a scalar. This is the  $r$  in the formula just above. It defaults to 1 in the case that a participation vector is not included. The vaccine study example demonstrates use of this argument.
- `integrate_allocation`: TRUE/FALSE. When group sizes grow large (over 1000), the product term of `logit_integrand` tends quickly to 0. When set to TRUE, the IP weights tend less quickly to 0. Defaults to FALSE.

If the true propensity model is known (e.g. in simulations) use `variance_estimation = 'naive'`; otherwise, use the default `variance_estimation = 'robust'`. Refer to the web appendix of Heydrich-Perez et al. (2014) ([doi:10.1111/biom.12184](https://doi.org/10.1111/biom.12184)) for complete details.

## Value

Returns a list of overall and group-level IPW point estimates, overall and group-level IPW point estimates (using the weight derivatives), derivatives of the loglikelihood, the computed weight matrix, the computed weight derivative array, and a summary.

## References

Saul, B. and Hudgens, M. G. (2017). A Recipe for inference: Start with Causal Inference. Add Interference. Mix Well with R. *Journal of Statistical Software*, 82(2), 1-21. [doi:10.18637/jss.v082.i02](https://doi.org/10.18637/jss.v082.i02)

Perez-Heydrich, C., Hudgens, M. G., Halloran, M. E., Clemens, J. D., Ali, M., & Emch, M. E. (2014). Assessing effects of cholera vaccination in the presence of interference. *Biometrics*, 70(3), 731-741.

Tchetgen Tchetgen, E. J., & VanderWeele, T. J. (2012). On causal inference in the presence of interference. *Statistical Methods in Medical Research*, 21(1), 55-75.

---

|                 |   |
|-----------------|---|
| logit_integrand | <i>Default integrand for the group-level propensity score</i> |
|-----------------|---|

---

## Description

Computes the following function:

$$\prod_{j=1}^n (rh_j(b))^{A_j} (1 - rh_j(b))^{1-A_j} f_b(b; \theta_b)$$

where  $r$  is the randomization scheme.  $X$  is the covariate(s) vectors.  $fixef$  is the vector of fixed effects.  $b$  is the random (group-level) effect.  $ranef$  is the random effect variance.

## Usage

```
logit_integrand(b, X, A, parameters, allocation = A, randomization = 1)
```

**Arguments**

|                            |   |
|----------------------------|---|
| <code>b</code>             | vector argument of values necessary for <code>integrate</code> .  |
| <code>X</code>             | <code>n</code> by <code>length</code> (fixed effects) matrix of covariates.   |
| <code>A</code>             | vector of binary treatments   |
| <code>parameters</code>    | vector of fixed effect (and random effect if applicable). Random effect should be last element in vector.                             |
| <code>allocation</code>    | The allocation strategy. Defaults to <code>A</code> so that is essentially ignored if allocation is not set to a value within (0, 1). |
| <code>randomization</code> | Randomization probability. Defaults to 1.   |

**Value**

value of the integrand

---

|                             |                       |
|-----------------------------|-----------------------|
| <code>log_likelihood</code> | <i>Log Likelihood</i> |
|-----------------------------|-----------------------|

---

**Description**

Used by `score_matrix` to compute the log likelihood.

**Usage**

```
log_likelihood(parameters, integrand, ...)
```

**Arguments**

|                         |  |
|-------------------------|--|
| <code>parameters</code> | vector of parameters passed to integrand           |
| <code>integrand</code>  | Defaults to <code>logit_integrand</code>           |
| <code>...</code>        | additional arguments passed to integrand function. |

**Value**

value of log likelihood



---

|                |  |
|----------------|--|
| overall_effect | <i>Retrieve Overall Effect Estimates</i> |
|----------------|--|

---

**Description**

Retrieves the population average overall causal effect:  $\hat{Y}(\alpha1) - \hat{Y}(\alpha2)$

**Usage**

```
overall_effect(object, allocation1, allocation2 = NULL)
```

```
oe(object, allocation1, allocation2 = NULL)
```

**Arguments**

|             |   |
|-------------|---|
| object      | an object of class <code>interference</code>                |
| allocation1 | the allocation scheme for which to estimate overall effects |
| allocation2 | the allocation scheme for which to estimate overall effects |

**Value**

a data.frame with a single row with requested values

---

|                    |   |
|--------------------|---|
| print.interference | <i>Prints a summary of an interference object</i> |
|--------------------|---|

---

**Description**

Prints a summary of an interference object

**Usage**

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

**Arguments**

|     |                                |
|-----|--------------------------------|
| x   | object of class 'interference' |
| ... | ignored                        |

---

|            |  |
|------------|--|
| score_calc | <i>Compute scores for a single group</i> |
|------------|--|

---

**Description**

Used by [score\\_matrix](#) to log likelihood derivatives for a single group.

**Usage**

```
score_calc(parameters, integrand, hide.errors = TRUE, ...)
```

**Arguments**

|             |   |
|-------------|---|
| parameters  | vector of parameters passed to integrand  |
| integrand   | function to used for the integrand. Defaults to <a href="#">logit_integrand</a> . |
| hide.errors | Hide errors printed from <a href="#">grad</a> . Defaults to true.                 |
| ...         | additional arguments pass to the integrand function.                              |

**Value**

length(theta) vector of scores

---

|              |   |
|--------------|---|
| score_matrix | <i>Calculate matrix of log Likelihood derivatives</i> |
|--------------|---|

---

**Description**

Calculate matrix of log Likelihood derivatives

**Usage**

```
score_matrix(integrand, X, A, G, parameters, runSilent = TRUE, ...)
```

**Arguments**

|            |  |
|------------|--|
| integrand  | function passed to <a href="#">log_likelihood</a> . Defaults to <a href="#">logit_integrand</a>  |
| X          | covariate matrix   |
| A          | vector of treatment assignments  |
| G          | vector of group assignments  |
| parameters | vector of parameters passed to integrand   |
| runSilent  | If FALSE, prints errors to console. Defaults to TRUE.  |
| ...        | additional arguments passed to integrand or <a href="#">grad</a> . For example, one can change the method argument in <a href="#">grad</a> . |

**Value**

N X length(params) matrix of scores

---

|              |  |
|--------------|--|
| total_effect | <i>Retrieve Total Effect estimates</i> |
|--------------|--|

---

**Description**

Retrieves the population average total causal effect for specified allocations:  $\hat{Y}(0, \alpha1) - \hat{Y}(1, \alpha2)$

**Usage**

```
total_effect(object, allocation1, allocation2 = NULL, trt.lv11 = 0)
```

```
te(object, allocation1, allocation2 = NULL, trt.lv11 = 0)
```

**Arguments**

|             |   |
|-------------|---|
| object      | an object of class interference   |
| allocation1 | the allocation scheme for which to estimate total effects   |
| allocation2 | the allocation scheme for which to estimate total effects If NULL, then returns all indirect effects compared to allocation1. |
| trt.lv11    | Defaults to 0.  |

**Value**

a data.frame with requested values

---

|            |                                  |
|------------|----------------------------------|
| vaccinesim | <i>Vaccine Study Sample Data</i> |
|------------|----------------------------------|

---

**Description**

A sample dataset based on the simulations of a cholera vaccine trial in Heydrich-Perez et al. (2014) ([doi:10.1111/biom.12184](https://doi.org/10.1111/biom.12184)) except with 3000 individuals in 250 groups rather than 10000 in 500.

**Format**

a dataset with 6 variables and 3000 rows

**Y** the outcome (0 - no cholera; 1 - cholera)

**X1** an individual's age (in decades)

**X2** an individual's distance from river

**A** an indicator of vaccination (0 - no vaccine; 1 - vaccine)

**B** an indicator of participation (0 - did not participant in vaccine trial, 1 - did participate)

**group** group membership

## References

Perez-Heydrich, C., Hudgens, M. G., Halloran, M. E., Clemens, J. D., Ali, M., & Emch, M. E. (2014). Assessing effects of cholera vaccination in the presence of interference. *Biometrics*, 70(3), 731-741.

---

voters

*Voting Contagion Experiment Data*

---

## Description

A dataset of a voting contagion experiment. See Nickerson (2008) for more details. The variables used in the package vignette are documented here.

## Format

a dataset with 21 variables and 7722 rows

**family** household ID

**denver** 1 = subject in Denver, 0 = Minneapolis

**treatment** 1 = voting encouragement, 2 = recycling message, 3 = not contacted

**reached** 1 = subject answered door, 0 = not

**hsecontact** 1 = household contacted by canvassers, 0 = not

**voted02p** 1 = voted in '02 primary, 0 = not

**party** party affiliation

**age** age

**gender** gender

## References

Nickerson, D. W. (2008). Is voting contagious? Evidence from two field experiments. *American Political Science Review*, 102(01), 49-57. doi:[10.1017/S0003055408080039](https://doi.org/10.1017/S0003055408080039)

---

|           |                           |
|-----------|---------------------------|
| wght_calc | <i>Compute IPW weight</i> |
|-----------|---------------------------|

---

### Description

Calculates the IPW for a single group. Used by `wght_matrix` to create a matrix of weights for each group and allocation scheme.

### Usage

```
wght_calc(parameters, integrand, allocation, integrate_allocation = TRUE, ...)
```

### Arguments

|                                   |  |
|-----------------------------------|--|
| <code>parameters</code>           | vector of parameter values   |
| <code>integrand</code>            | function to pass to the argument 'f' of <code>integrate</code> .                             |
| <code>allocation</code>           | the allocation ratio for which to compute the weight   |
| <code>integrate_allocation</code> | Indicator of whether the integrand function uses the allocation parameter. Defaults to TRUE. |
| <code>...</code>                  | other arguments passed to <code>integrand</code> .   |

### Details

If `allocation` is an argument in the integrand function and `integrate_allocation == TRUE`, then the weight is calculated as:

$$\frac{1}{Pr(A|X)}$$

Otherwise, the weight is computed by:

$$\frac{\prod_{j=1}^n \alpha_j^A (1 - \alpha)^{(1 - A_j)}}{Pr(A|X)}$$

### Value

scalar result of the integral

---

wght\_deriv\_array      *Create an array of group weight derivatives*

---

### Description

Uses [wght\\_deriv\\_calc](#) to compute the weight derivatives for each group per coverage level

### Usage

```
wght_deriv_array(
  parameters,
  integrand,
  allocations,
  X,
  A,
  G,
  runSilent = TRUE,
  integrate_allocation = TRUE,
  ...
)
```

### Arguments

|                      |  |
|----------------------|--|
| parameters           | vector of parameters passed to integrand   |
| integrand            | function to pass to the argument 'f' of <a href="#">integrate</a> .                          |
| allocations          | coverage levels in [0, 1]. Can be vector.  |
| X                    | covariate matrix   |
| A                    | vector of treatment assignments  |
| G                    | vector of group assignments  |
| runSilent            | if FALSE, errors are printed to console. Defaults to TRUE.                                   |
| integrate_allocation | Indicator of whether the integrand function uses the allocation parameter. Defaults to TRUE. |
| ...                  | other arguments passed to integrand.   |

### Value

a length(unique(group)) X length(params) X length(alphas) array of group weight derivatives

---

|                 |  |
|-----------------|--|
| wght_deriv_calc | <i>Compute the derivative(s) of a weight</i> |
|-----------------|--|

---

**Description**

Takes the derivative of the [wght\\_calc](#) function with respect to each parameter in params.

**Usage**

```
wght_deriv_calc(
  integrand,
  parameters,
  allocation,
  integrate_allocation = TRUE,
  ...
)
```

**Arguments**

|                      |  |
|----------------------|--|
| integrand            | function to pass to the argument 'f' of <a href="#">integrate</a> .                          |
| parameters           | vector of parameter values   |
| allocation           | the allocation ratio for which to compute the weight   |
| integrate_allocation | Indicator of whether the integrand function uses the allocation parameter. Defaults to TRUE. |
| ...                  | other arguments passed to integrand.   |

**Value**

vector of derivatives with respect to element of params

---

|             |   |
|-------------|---|
| wght_matrix | <i>Creates a number of groups by number of allocation schemes matrix of group weights. Allocation schemes are selected by the user.</i> |
|-------------|---|

---

**Description**

Groups should be numbered 1, ..., N

**Usage**

```
wght_matrix(  
  integrand,  
  allocations,  
  X,  
  A,  
  G,  
  parameters,  
  runSilent = TRUE,  
  integrate_allocation = TRUE,  
  ...  
)
```

**Arguments**

|                      |  |
|----------------------|--|
| integrand            | function to pass to the argument 'f' of <a href="#">integrate</a> .                          |
| allocations          | coverage levels in [0, 1]. Can be vector.  |
| X                    | covariate matrix   |
| A                    | vector of treatment assignments  |
| G                    | vector of group assignments  |
| parameters           | vector of parameters passed to integrand   |
| runSilent            | if FALSE, errors are printed to console. Defaults to TRUE.                                   |
| integrate_allocation | Indicator of whether the integrand function uses the allocation parameter. Defaults to TRUE. |
| ...                  | other arguments passed to integrand.   |

**Value**

a  $\text{length}(\text{unique}(\text{group})) \times \text{length}(\text{alphas})$  matrix of group weights



# Index

## \* datasets

vaccinesim, 11

voters, 12

diagnose\_weights, 2

direct\_effect, 3

get\_args, 3

grad, 10

hist, 2

ie (indirect\_effect), 4

indirect\_effect, 4

interference, 4

interference-package (interference), 4

integrate, 8, 13–16

interference, 5

log\_likelihood, 8, 10

logit\_integrand, 7, 10

oe (overall\_effect), 9

overall\_effect, 9

print.interference, 9

score\_calc, 10

score\_matrix, 8, 10, 10

te (total\_effect), 11

total\_effect, 11

vaccinesim, 11

voters, 12

wght\_calc, 13, 15

wght\_deriv\_array, 14

wght\_deriv\_calc, 14, 15

wght\_matrix, 13, 15