# Package 'gapclosing'

August 28, 2021

**Title** Estimate Gaps Under an Intervention

**Version** 1.0.0

**Description**
Provides functions to estimate the disparities across categories (e.g. Black and white) that persists if a treatment variable (e.g. college) is equalized. Makes estimates by treatment modeling, outcome modeling, and doubly-robust augmented inverse probability weighting estimation, with standard errors calculated by a nonparametric bootstrap. Cross-fitting is supported. Survey weights are supported for point estimation but not for standard error estimation; those applying this package with complex survey samples should consult the data distributor to select an appropriate approach for standard error construction, which may involve calling the functions repeatedly for many sets of replicate weights provided by the data distributor.

**License** MIT + file LICENSE

**URL** https://ilundberg.github.io/gapclosing/

**BugReports** https://github.com/ilundberg/gapclosing/issues

**Encoding** UTF-8

**Roxygen** list(markdown = TRUE)

**RoxygenNote** 7.1.1

**Imports** stats, utils, mgcv, ranger, glmnet, Rdpack, magrittr, dplyr, forcats, ggplot2, gridExtra, tidyr, foreach, tidyselect, parallel, doParallel

**Suggests** testthat (>= 3.0.0), knitr, rmarkdown, spelling

**VignetteBuilder** knitr

**NeedsCompilation** no

**Author** Ian Lundberg [aut, cre] (<https://orcid.org/0000-0002-1909-2270>)

**Maintainer** Ian Lundberg <ianlundberg@ucla.edu>

**Config/testthat/edition** 3

**Language** en-US

## R topics documented:

---

`as.data.frame.gapclosing` *Coerce to a Data Frame*

---

## Description

This function converts a `gapclosing` object into a data frame. The gapclosing class contains results within a named list, thus simplifying things for manual user interaction with the results. In some programming settings (e.g. a bootstrap), it is easier to work with a rectangular data frame of results. This function produces that data frame.

## Usage

```
## S3 method for class 'gapclosing'
as.data.frame(x, ...)
```

## Arguments

| | |
|---|---|
| `x` | Object of class `gapclosing`, produced by a call to `gapclosing()`. |
| `...` | Additional arguments to be passed to or from methods. |

## Value

A data frame containing estimates.

## References

Lundberg I (2021). "The gap-closing estimand: A causal approach to study interventions that close disparities across social categories." Sociological Methods and Research. Available at https://osf.io/gx4y3/.

## Examples

```
# Simulate example data
simulated_data <- generate_simulated_data(n = 100)

# Fit by outcome modeling
estimate <- gapclosing(
  data = simulated_data,
  outcome_formula = formula(outcome ~ treatment * category + confounder),
  treatment_name = "treatment",
  category_name = "category",
  counterfactual_assignments = 1
```

```
)
summary(estimate)

# Convert to a data frame
estimate.df <- as.data.frame(estimate)
```

---

`cross_fit_estimator` *Cross-fitting gap closing estimator*

---

## Description

This is an internal function typically called from other functions rather than by the user. It creates cross-validation folds and repeatedly calls split_sample_estimator to conduct cross-fitting.

## Usage

```
cross_fit_estimator(
  data,
  counterfactual_assignments,
  outcome_formula,
  treatment_formula,
  category_name,
  outcome_name,
  treatment_name,
  treatment_algorithm = "glm",
  outcome_algorithm = "lm",
  weight_name = NULL,
  n_folds = 2,
  folds_name = NULL
)
```

## Arguments

| | |
|---|---|
| `data` | Data frame containing the observed data |
| `counterfactual_assignments` | Numeric scalar or vector of length nrow(data), each element of which is on the [0,1] interval. If a scalar, the counterfactual probability by which all units are assigned to treatment condition 1. If a vector, each element i corresponds to the counterfactual probability by which each unit i is assigned to treatment condition 1. |
| `outcome_formula` | Model formula the outcome. Covariates should include those needed for causal identification of the treatment effect (e.g. as defended in your Directed Acyclic Graph). If `outcome_algorithm` = "ranger", then the outcome model will be fit separately on the treatment and control groups. Otherwise, the user must specify all interactions in the formula. |
| `treatment_formula` | Treatment formula, in the style formula(treatment ~ covariates). Covariates should include those needed for causal identification of the treatment effect (e.g. as defended in your Directed Acyclic Graph). |

`category_name` Character name of the variable indicating the categories over which the gap is defined. Must be the name of a column in `data`.

`outcome_name` Character name of the outcome variable. Only required when there is no outcome_formula; otherwise extracted automatically. Must be a name of a column in `data`.

`treatment_name` Character name of the treatment variable. Only required when there is no treatment_formula; otherwise extracted automatically. Must be a name of a column in `data`.

`treatment_algorithm`
Character name of the algorithm for the treatment model. One of "glm", "ridge", "gam", or "ranger". Defaults to "glm", which is a logit model. Option "ridge" is ridge regression. Option "gam" is a generalized additive model fit (see package `mgcv`). Option "ranger" is a random forest (see package `ranger`). If "ranger", this function avoids propensity scores equal to 0 or 1 by bottom- and top-coding predicted values at .001 and .999.

`outcome_algorithm`
Character name of the algorithm for the outcome model. One of "lm", "ridge", "gam", or "ranger". Defaults to "lm", which is an OLS model. Option "ridge" is ridge regression. Option "gam" is a generalized additive model fit (see package `mgcv`). Option "ranger" is a random forest (see package `ranger`).

`weight_name` Character name of a sampling weight variable, if any, which captures the inverse probability of inclusion in the sample. The default assumes a simple random sample (all weights equal).

`n_folds` Only used if `method` = "cross_fit" and if `folds` is not provided. Integer scalar containing number of cross-validation folds. The function will assign observations to folds systematically: sort the data by the variable named `category_name`, then by the treatment variable, then at random. On this sorted dataset, folds are assigned systematically by repeated `1:n_folds`. To be used if the user does not provide `folds`. Defaults to 2.

`folds_name` Only used if `method` = "cross_fit". Character string indicating a column of `data` containing fold identifiers. This may be preferable to `n_folds` if the researcher has a reason to assign the folds in these data by some other process, perhaps due to particulars of how these data were generated. If null (the default), folds are assigned as stated in `n_folds`.

## Value

A list with four elements.

`counterfactual_means` A tibble with a counterfactual mean estimate for each category `counterfactual_means`
A tibble with a counterfactual disparity estimate for each pair of categories `treatment_model`
Object containing the fitted treatment model `outcome_model` Object containing the fitted outcome model

## References

Lundberg I (2021). "The gap-closing estimand: A causal approach to study interventions that close disparities across social categories." Sociological Methods and Research. Available at https://osf.io/gx4y3/.

`df_to_gapclosing_list` *Convert Back to Canonical List Output*

## Description

If the user has used `as.data.frame(x)` to convert a `gapclosing` object to a data frame of estimates, this function will invert back to the original list format. This function does not fully reinstate the original gapclosing object because some elements are lost when `as.data.frame()` is called. This function is most useful as a check on `as.data.frame()` and as a helper in settings like bootstrapping where a data frame is easier to work with but we want to return to the original format before returning an object to the user.

## Usage

```
df_to_gapclosing_list(x)
```

## Arguments

| | |
|---|---|
| `x` | A data frame produced by `as.data.frame(x)` applied to an object `x` of class `gapclosing`. |

## Value

A list containing a subset of the elements in a `gapclosing` object.

## References

Lundberg I (2021). "The gap-closing estimand: A causal approach to study interventions that close disparities across social categories." Sociological Methods and Research. Available at https://osf.io/gx4y3/.

## Examples

```
# Simulate example data
simulated_data <- generate_simulated_data(n = 100)

# Fit by outcome modeling
estimate <- gapclosing(
  data = simulated_data,
  outcome_formula = formula(outcome ~ treatment * category + confounder),
  treatment_name = "treatment",
  category_name = "category",
  counterfactual_assignments = 1
)
summary(estimate)

# Convert to a data frame
estimate.df <- as.data.frame(estimate)
# Convert back to a list
estimate.df <- df_to_gapclosing_list(estimate.df)
```

## disparityplot *Plot a disparity*

### Description

Plots the factual and counterfactual mean outcomes in two categories. The returned object is a ggplot2 object which can be further customized using the syntax of ggplot2.

### Usage

```
disparityplot(
  x,
  category_A,
  category_B,
  custom_ylab = "Mean Outcome",
  custom_xlab = "Category"
)
```

### Arguments

| | |
|---|---|
| x | An object of class gapclosing, which results from a call to the function gapclosing |
| category_A | The first category to be plotted. A value of the category_name variable in x. |
| category_B | The second category to be plotted. Must be a value of x$category |
| custom_ylab | Custom y-axis label. Defaults to "Mean Outcome" |
| custom_xlab | Custom x-axis label. Defaults to "Category" |

### Value

A ggplot2 object

### References

Lundberg I (2021). "The gap-closing estimand: A causal approach to study interventions that close disparities across social categories." Sociological Methods and Research. Available at https://osf.io/gx4y3/.

## fit_ridge *Ridge regression estimator*

### Description

Not typically called by the user directly; called indirectly via other functions. Uses glmnet to fit a ridge regression with penalty chosen by cross-validation. Returns fitted values for the data in to_predict.

### Usage

```
fit_ridge(data, model_formula, to_predict)
```

### Arguments

| | |
|---|---|
| `data` | Data frame containing the observed data |
| `model_formula` | A model formula object for the ridge regression to be fitted |
| `to_predict` | Data frame containing observations for which predictions are to be made. If NULL, defaults to the same as `data`. |

### Value

A data frame containing predicted values for observations in `to_predict`.

### References

Lundberg I (2021). "The gap-closing estimand: A causal approach to study interventions that close disparities across social categories." Sociological Methods and Research. Available at https://osf.io/gx4y3/.

Friedman J, Hastie T, Tibshirani R (2010). "Regularization Paths for Generalized Linear Models via Coordinate Descent." Journal of Statistical Software, 33(1), 1–22. https://www.jstatsoft.org/v33/i01/.

---

## `gapclosing` *Gap closing estimator*

---

### Description

A function to estimate gap-closing estimands: means and disparities across categories of units that would persist under some counterfactual assignment of a treatment. To use this function, the user provides a data frame `data`, a rule `counterfactual_assignments` for counterfactually assigning treatment, a treatment and/or an outcome model for learning statistically about the counterfactuals, and the `category_name` of the variable in `data` over which categories are defined. The returned object summarizes factual and counterfactual means and disparities. Supported estimation algorithms include generalized linear models, ridge regression, generalized additive models, and random forests. Standard errors are supported by bootstrapping.

### Usage

```
gapclosing(
  data,
  counterfactual_assignments,
  outcome_formula = NULL,
  treatment_formula = NULL,
  category_name,
  outcome_name = NULL,
  treatment_name = NULL,
  treatment_algorithm = "glm",
  outcome_algorithm = "lm",
  sample_split = "single_sample",
  se = FALSE,
  bootstrap_samples = 1000,
  bootstrap_method = "simple",
  parallel_cores = NULL,
  weight_name = NULL,
  n_folds = 2,
  folds_name = NULL
)
```

## Arguments

| | |
|---|---|
| data | Data frame containing the observed data |
| counterfactual_assignments | Numeric scalar or vector of length nrow(data), each element of which is on the [0,1] interval. If a scalar, the counterfactual probability by which all units are assigned to treatment condition 1. If a vector, each element i corresponds to the counterfactual probability by which each unit i is assigned to treatment condition 1. |
| outcome_formula | Outcome formula , in the style `outcome ~ treatment*covariate`. Covariates should include those needed for causal identification of the treatment effect (e.g. as defended in your Directed Acyclic Graph). If `outcome_algorithm = "ranger"`, then the outcome model will be fit separately on the treatment and control groups. Otherwise, the user must specify all interactions in the formula. |
| treatment_formula | Treatment formula, in the style `treatment ~ covariate`. Covariates should include those needed for causal identification of the treatment effect (e.g. as defended in your Directed Acyclic Graph). |
| category_name | Character name of the variable indicating the categories over which the gap is defined. Must be the name of a column in `data`. |
| outcome_name | Character name of the outcome variable. Only required when there is no outcome_formula; otherwise extracted automatically. Must be a name of a column in `data`. |
| treatment_name | Character name of the treatment variable. Only required when there is no treatment_formula; otherwise extracted automatically. Must be a name of a column in `data`. |
| treatment_algorithm | Character name of the algorithm for the treatment model. One of "glm", "ridge", "gam", or "ranger". Defaults to "glm", which is a logit model. Option "ridge" is ridge regression. Option "gam" is a generalized additive model fit (see package `mgcv`). Option "ranger" is a random forest (see package `ranger`). If "ranger", this function avoids propensity scores equal to 0 or 1 by bottom- and top-coding predicted values at .001 and .999. |
| outcome_algorithm | Character name of the algorithm for the outcome model. One of "lm", "ridge", "gam", or "ranger". Defaults to "lm", which is an OLS model. Option "ridge" is ridge regression. Option "gam" is a generalized additive model fit (see package `mgcv`). Option "ranger" is a random forest (see package `ranger`). |
| sample_split | Character for the type of sample splitting to be conducted. One of "single_sample" or "cross_fit". Defaults to "single_sample", in which case `data` is used for both learning the nuisance functions and aggregating to an estimate. Option "cross_fit" uses cross-fitting to repeatedly use part of the sample to learn the nuisance function and another part to estimate the estimand, averaged over repetitions with these roles swapped. |
| se | Logical indicating whether standard errors should be calculated. Default is FALSE. Standard errors assume a simple random sample by default; to stratify by (category x treatment), see the `bootstrap_method` argument. Because many datasets are not simple random samples, users should carefully consider whether a simple random sample bootstrap will accurately capture uncertainty. |

`bootstrap_samples`
: Only used if `se = TRUE`. Number of bootstrap samples. Default is 1000.

`bootstrap_method`
: Only used if `se = TRUE`. A character string stating how to conduct bootstrap samples. If "simple", then samples are drawn with replacement from the full data. If "stratified", then the bootstrap is carried out within subpopulations defined by category and treatment. The latter may be useful if the sample contains only a small number of observations in these cells and the user wants to ensure that every (category x treatment) cell appears in every bootstrap sample. With "stratified", inference assumes that in repeated samples from the true population the proportion in each (category x treatment) cell would not change; this may or may not correspond to the true sampling process. Users should be cautious.

`parallel_cores`
: Integer number of cores for parallel processing of the bootstrap. Defaults to sequential processing.

`weight_name`
: Character name of a sampling weight variable, if any, which captures the inverse probability of inclusion in the sample. The default assumes a simple random sample (all weights equal).

`n_folds`
: Only used if `method = "cross_fit"` and if `folds` is not provided. Integer scalar containing number of cross-validation folds. The function will assign observations to folds systematically: sort the data by the variable named `category_name`, then by the treatment variable, then at random. On this sorted dataset, folds are assigned systematically by repeated `1:n_folds`. To be used if the user does not provide `folds`. Defaults to 2.

`folds_name`
: Only used if `method = "cross_fit"`. Character string indicating a column of `data` containing fold identifiers. This may be preferable to `n_folds` if the researcher has a reason to assign the folds in these data by some other process, perhaps due to particulars of how these data were generated. If null (the default), folds are assigned as stated in `n_folds`.

## Value

An object of S3 class `gapclosing`, which supports `summary()`, `print()`, and `plot()` functions. The returned object can be coerced to a data frame of estimates with `as.data.frame()`.
The object returned by a call to `gapclosing` contains several elements.

- `factual_means` A tibble containing the factual mean outcome in each category
- `factual_disparities` A tibble containing the disparities in factual mean outcomes across categories
- `counterfactual_means` A tibble containing the counterfactual mean outcome (post-intervention mean) in each category
- `counterfactual_disparities` A tibble containing the counterfactual disparities (gap-closing estimands) across categories
- `change_means` A tibble containing the additive and proportional change from factual to counterfactual values for mean outcomes

- `change_disparities` A tibble containing the additive and proportional change from factual to counterfactual values for disparities in mean outcomes (e.g. proportion of the factual gap which is closed by the intervention)

- `all_estimators` A list containing estimates by treatment modeling, outcome modeling, and doubly-robust estimation. If any of these are not applicable, estimates are NA.

- `primary_estimator_name` The name of the primary estimator (treatment_modeling, outcome_modeling, or doubly_robust). The estimates reported in the first 6 slots of the returned object come from this estimator.

- `treatment_model` The fitted treatment model (or models on each fold in the case of cross-fitting). Note that this model object is a point estimate with standard errors derived from the algorithm used to fit it; any standard errors within `treatment_model` do not come from bootstrapping by the package.

- `outcome_model` The fitted outcome model (or models on each fold in the case of cross-fitting). Note that this model object is a point estimate with standard errors derived from the algorithm used to fit it; any standard errors within `treatment_model` do not come from bootstrapping by the package.

- `call` The call that produced this `gapclosing` object

- `arguments` A list of all arguments from the call to `gapclosing`

## References

Lundberg I (2021). "The gap-closing estimand: A causal approach to study interventions that close disparities across social categories." Sociological Methods and Research. Available at https://osf.io/gx4y3/.

Friedman J, Hastie T, Tibshirani R (2010). "Regularization Paths for Generalized Linear Models via Coordinate Descent." Journal of Statistical Software, 33(1), 1–22. https://www.jstatsoft.org/v33/i01/.

Wood S (2017). Generalized Additive Models: An Introduction with R, 2 edition. Chapman and Hall/CRC.

Wright MN, Ziegler A (2017). "ranger: A Fast Implementation of Random Forests for High Dimensional Data in C++ and R." Journal of Statistical Software, 77(1), 1–17. doi: 10.18637/jss.v077.i01.

## Examples

```
# Simulate example data
simulated_data <- generate_simulated_data(n = 100)

# Fit by outcome modeling
# You can add standard errors with se = T
estimate <- gapclosing(
  data = simulated_data,
  outcome_formula = outcome ~ treatment * category + confounder,
  treatment_name = "treatment",
  category_name = "category",
  counterfactual_assignments = 1
)
summary(estimate)
```

```
# Fit by treatment modeling
# You can add standard errors with se = T
estimate <- gapclosing(
  data = simulated_data,
  treatment_formula = treatment ~ category + confounder,
  outcome_name = "outcome",
  category_name = "category",
  counterfactual_assignments = 1
)
summary(estimate)

# Fit by doubly-robust estimation
# You can add standard errors with se = T
estimate <- gapclosing(
  data = simulated_data,
  outcome_formula = outcome ~ treatment * category + confounder,
  treatment_formula = treatment ~ category + confounder,
  category_name = "category",
  counterfactual_assignments = 1
)
summary(estimate)

# Fit by doubly-robust cross-fitting estimation with random forests
# You can add standard errors with se = T
estimate <- gapclosing(
  data = simulated_data,
  outcome_formula = outcome ~ category + confounder,
  treatment_formula = treatment ~ category + confounder,
  category_name = "category",
  counterfactual_assignments = 1,
  outcome_algorithm = "ranger",
  treatment_algorithm = "ranger",
  sample_split = "cross_fit"
)
summary(estimate)
```

---

`generate_simulated_data` *Generate simulated data*

---

## Description

Generates simulated data to illustrate the gapclosing function

## Usage

```
generate_simulated_data(n = 1000)
```

## Arguments

| | |
|---|---|
| `n` | Number of observations to be generated |

## Value

A data frame with n rows and 4 columns containing simulated data containing `category` over which disparities are defined, a `confounder` that affects treatment assignment, a binary `treatment`, and a continuous `outcome`.

## References

Lundberg I (2021). "The gap-closing estimand: A causal approach to study interventions that close disparities across social categories." Sociological Methods and Research. Available at https://osf.io/gx4y3/.

---

`pairwise_diff` *Pairwise difference calculator*

---

## Description

A function to estimate the pairwise differences of estimates made for each category.

## Usage

```
pairwise_diff(category_means_data, category_name)
```

## Arguments

| | |
|---|---|
| `category_means_data` | Data frame containing two columns: the category-specific mean estimate (a column named `estimate`) and the category name (named as specified in `category_name`) |
| `category_name` | The name of the column containing the category identifier. |

## Value

A data frame with pairwise differences of `estimate` over pairs of categories.

## References

Lundberg I (2021). "The gap-closing estimand: A causal approach to study interventions that close disparities across social categories." Sociological Methods and Research. Available at https://osf.io/gx4y3/.

Friedman J, Hastie T, Tibshirani R (2010). "Regularization Paths for Generalized Linear Models via Coordinate Descent." Journal of Statistical Software, 33(1), 1–22. https://www.jstatsoft.org/v33/i01/.

Wood S (2017). Generalized Additive Models: An Introduction with R, 2 edition. Chapman and Hall/CRC.

Wright MN, Ziegler A (2017). "ranger: A Fast Implementation of Random Forests for High Dimensional Data in C++ and R." Journal of Statistical Software, 77(1), 1–17. doi: 10.18637/jss.v077.i01.

## Examples

```
sim_data <- data.frame(example_category = c("A","B","C"),
                       estimate = c(1,2,3))
pairwise_diff(sim_data, category_name = "example_category")
```

## plot.gapclosing *Plot function for gapclosing objects*

### Description

Produces summary plots for a gapclosing object.

### Usage

```
## S3 method for class 'gapclosing'
plot(x, return_plots = F, arranged = F, ...)
```

### Arguments

| | |
|---|---|
| x | An object of class gapclosing, which results from a call to the function gapclosing |
| return_plots | Logical, defaults to FALSE. If TRUE, returns a list of the 4 plots without printing. Defaults to FALSE, in which case the console will interactively ask the user to hit "return" to proceed through printouts of the four plots, with no plots returned. |
| arranged | Logical, defaults to FALSE. If TRUE, returns a list of the 4 plots arranged in a 2x2 table. Useful to visualize all four in one screen. |
| ... | Other arguments to plot commands |

### Value

If return_plots = TRUE, returns a list of ggplot2 objects. If return_plots = FALSE (the default), then nothing is returned and output is printed.

### References

Lundberg I (2021). "The gap-closing estimand: A causal approach to study interventions that close disparities across social categories." Sociological Methods and Research. Available at https://osf.io/gx4y3/.

## point_estimator *Point estimator for gap-closing estimands*

### Description

This is an internal function typically called from other functions rather than by the user. It uses a learning sample to learn the nuisance functions (treatment and outcome model) and then an auxiliary estimation sample to use those functions in estimation of the gap-closing estimand. For single-sample estimation, both the learning and estimation samples are the same. For cross-fitting, this function is called repeatedly with the roles of each dataset swapped.

## Usage

```
point_estimator(
  data_learn,
  data_estimate,
  counterfactual_assignments,
  outcome_formula,
  treatment_formula,
  category_name,
  outcome_name,
  treatment_name,
  treatment_algorithm = "glm",
  outcome_algorithm = "lm",
  weight_name = NULL
)
```

## Arguments

| | |
|---|---|
| data_learn | Data frame in which treatment and outcome models will be learned |
| data_estimate | Data frame in which the learned models will be converted to an estimate of the gap-closing estimand |
| counterfactual_assignments | Numeric scalar or vector of length nrow(data), each element of which is on the [0,1] interval. If a scalar, the counterfactual probability by which all units are assigned to treatment condition 1. If a vector, each element i corresponds to the counterfactual probability by which each unit i is assigned to treatment condition 1. |
| outcome_formula | Model formula the outcome. Covariates should include those needed for causal identification of the treatment effect (e.g. as defended in your Directed Acyclic Graph). If outcome_algorithm = "ranger", then the outcome model will be fit separately on the treatment and control groups. Otherwise, the user must specify all interactions in the formula. |
| treatment_formula | Treatment formula, in the style formula(treatment ~ covariates). Covariates should include those needed for causal identification of the treatment effect (e.g. as defended in your Directed Acyclic Graph). |
| category_name | Character name of the variable indicating the categories over which the gap is defined. Must be the name of a column in data. |
| outcome_name | Character name of the outcome variable. Only required when there is no outcome_formula; otherwise extracted automatically. Must be a name of a column in data. |
| treatment_name | Character name of the treatment variable. Only required when there is no treatment_formula; otherwise extracted automatically. Must be a name of a column in data. |
| treatment_algorithm | Character name of the algorithm for the treatment model. One of "glm", "ridge", "gam", or "ranger". Defaults to "glm", which is a logit model. Option "ridge" is ridge regression. Option "gam" is a generalized additive model fit (see package mgcv). Option "ranger" is a random forest (see package ranger). If "ranger", this function avoids propensity scores equal to 0 or 1 by bottom- and top-coding predicted values at .001 and .999. |

outcome_algorithm
: Character name of the algorithm for the outcome model. One of "lm", "ridge", "gam", or "ranger". Defaults to "lm", which is an OLS model. Option "ridge" is ridge regression. Option "gam" is a generalized additive model fit (see package mgcv). Option "ranger" is a random forest (see package ranger).

weight_name
: Character name of a sampling weight variable, if any, which captures the inverse probability of inclusion in the sample. The default assumes a simple random sample (all weights equal).

## Value

@return A list with four elements.

counterfactual_means A tibble with a counterfactual mean estimate for each category counterfactual_means A tibble with a counterfactual disparity estimate for each pair of categories treatment_model Object containing the fitted treatment model outcome_model Object containing the fitted outcome model

## References

Lundberg I (2021). "The gap-closing estimand: A causal approach to study interventions that close disparities across social categories." Sociological Methods and Research. Available at https://osf.io/gx4y3/.

---

# print.gapclosing *Print function for gapclosing objects*

---

## Description

Prints the same output as generated by summary

## Usage

```
## S3 method for class 'gapclosing'
print(
  x,
  ...,
  digits = 2,
  quote = FALSE,
  right = FALSE,
  row.names = FALSE,
  max = NULL
)
```

## Arguments

| | |
|---|---|
| x | An object of class gapclosing, which results from a call to the function gapclosing |
| ... | Other arguments to print commands |
| digits | Argument passed to print.data.frame |
| quote | Argument passed to print.data.frame |
| right | Argument passed to print.data.frame |
| row.names | Argument passed to print.data.frame |
| max | Argument passed to print.data.frame |

## Value

Prints a summary of the estimates.

## References

Lundberg I (2021). "The gap-closing estimand: A causal approach to study interventions that close disparities across social categories." Sociological Methods and Research. Available at https://osf.io/gx4y3/.

---

`summary.gapclosing` *Summary function for gapclosing objects*

---

## Description

Summarizes the S3 class object returned by the gapclosing function

## Usage

```
## S3 method for class 'gapclosing'
summary(object, ...)
```

## Arguments

| | |
|---|---|
| `object` | An object of class `gapclosing`, which results from a call to the function `gapclosing` |
| `...` | Other arguments to `summary` commands |

## Value

Prints a summary of the estimates.

## References

Lundberg I (2021). "The gap-closing estimand: A causal approach to study interventions that close disparities across social categories." Sociological Methods and Research. Available at https://osf.io/gx4y3/.

# Index