Package: cdid (via r-universe)

November 17, 2026

Type Package

Title The Chained Difference-in-Differences

Version 1.3.0.1

Maintainer David Benatia <david.benatia@hec.ca>

Description Estimates group-time average treatment effects in panels with missing outcomes by the chained difference-in-differences method of Bellégo, Benatia and Dortet-Bernadet (2025) <doi:10.1016/j.jeconom.2024.105783>. The package extends 'did' and retains the att_gt() and aggte() interfaces. Chained estimation combines observed changes within units by the generalized method of moments. It supports inverse probability weighting, outcome regression and doubly robust estimation. Missing outcomes can be handled by observation weighting, imputation of outcome changes, or an augmented combination of both. Users can choose comparison groups and parallel trends assumptions, fit linear cohort-by-date fixed effects, test trends before treatment, and inspect fitted regression models. A separate analysis estimates the effects of starting and stopping a binary treatment under no carryover. Ordinary mode uses the estimators of Callaway and Sant'Anna (2021) <doi:10.1016/j.jeconom.2020.12.001>.

License GPL-3

URL https://github.com/dvbn/cdid

BugReports https://github.com/dvbn/cdid/issues

Depends R (>= 4.1.0)

Imports did (>= 2.5.1), DRDID (>= 1.4.0), parallel, rlang (>= 1.1.0)

Suggests testthat (>= 3.0.0), codetools, digest, ggplot2, knitr, rmarkdown, generics, jsonlite, waldo, withr

VignetteBuilder knitr

Config/testthat/edition 3

Encoding UTF-8

NeedsCompilation yes

Author David Benatia [cre, aut], Christophe Bellégo [aut], Joel Cuerrier [aut], Vincent Dortet-Bernadet [aut]

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

Date/Publication 2026-11-17 08:30:36 UTC

RemoteUrl https://github.com/cran/cdid

RemoteRef HEAD

RemoteSha e902eff71bfea5880ebb6a605e49d7bfb64be8a4

Contents

cdid-package 2
aggte 3
att_gt 5
chain_report 11
conditional_did_pretest 13
nuisance_models 16
reexports 18

Index **19**

`cdid-package` *cdid: The Chained Difference-in-Differences*

Description

Estimate difference-in-differences effects with panels in which some outcomes are missing. Chained estimation combines changes measured within units across the dates when they are observed. Use `mode = "ordinary"` for **did**'s estimator or for repeated cross sections.

Details

Use `gname` to estimate effects by first-treatment cohort and date when treatment is absorbing. Use `dname` for effects at entry and exit when a binary treatment can change and past treatment has no carryover.

Choose the treatment adjustment with `est_method`. If the units observed at both dates represent their cohort with respect to mean changes only after conditioning on covariates, use `selection = "conditional_mtar"`. Choose observation weighting, imputation of actual changes or their augmented combination with `selection_est_method`.

- `att_gt()` fits the model from a long data frame and, for cohort models, an optional study-unit table.
- `aggte()` summarizes cohort effects. A switching fit already contains the switcher-weighted estimate in `fit$aggregate`.

- `chain_report()` shows which comparisons identify cohort effects and how they are weighted.
- `nuisance_models()` displays fitted equations and coefficient tables when requested during estimation.
- `conditional_did_pretest()` tests pre-treatment trends conditional on the model covariates.
- `tidy()`, `glance()` and `ggdid()` return tables, summaries and plots.

Open `vignette("cdid", package = "cdid")` for a hands-on introduction. The user guide covers preparing data, choosing a model and inspecting results. Use `citation("cdid")` for references.

References

Bellego, C., Benatia, D. and Dortet-Bernadet, V. (2025). The Chained Difference-in-Differences. *Journal of Econometrics* 248, 105783. doi:10.1016/j.jeconom.2024.105783

Callaway, B. and Sant'Anna, P. H. C. (2021). Difference-in-Differences with Multiple Time Periods. *Journal of Econometrics* 225(2), 200–230. doi:10.1016/j.jeconom.2020.12.001

de Chaisemartin, C., D'Haultfoeuille, X., Pasquier, F., Sow, D. and Vazquez-Bare, G. (2026). Difference-in-Differences Estimators for Treatments Continuously Distributed at Every Period. Working paper, arXiv:2201.06898v7 (revised 2026-07-22). Static binary specialization, used on the `dname` route.

See Also

`att_gt`, `aggte`, `chain_report`, `conditional_did_pretest`, `nuisance_models`, `reexports`, `vignette("cdid", package = "cdid")`

`aggte` *Aggregate group-time average treatment effects*

Description

Summarize fitted cohort-date effects into an overall effect, cohort averages, an event study or calendar-date averages. With a switching model, use the switcher-weighted estimate already stored in `fit$aggregate`.

Usage

```
aggte(
  MP,
  type = "group",
  balance_e = NULL,
  min_e = -Inf,
  max_e = Inf,
  na.rm = FALSE,
  bstrap = NULL,
  biters = NULL,
  cband = NULL,
```

```
  alp = NULL,
  clustervars = NULL
)
```

Arguments

MP	An MP or cdid_MP object returned by att_gt.
type	Use "simple" for an overall average, "group" for cohort averages, "dynamic" for event time, or "calendar" for calendar-time averages.
balance_e, min_e, max_e	Event-time selectors with did's meaning. Simple and group aggregation use only max_e. Dynamic aggregation uses event time t - g in original date units with did's window and cohort-balance rules. Calendar aggregation ignores all three with did's warning. These selectors change the reported target, not the fitted comparisons.
na.rm	FALSE (default): an aggregate that needs an unavailable cell is NA, and the missing cells are named. TRUE: the aggregate is computed over the available cells and labelled as a restricted target.
bstrap, biters, cband, alp, clustervars	Inference settings. For chained results, NULL inherits the value used by att_gt(). An explicit value overrides it with did's meaning. In particular bstrap = FALSE, cband = TRUE gives analytic standard errors with a multiplier band for group, dynamic and calendar bins, together with did's warning. Simple aggregation has no band. A cluster override that cannot be honoured is refused rather than silently changed.

Details

For chained results, aggregation reuses the influence functions and cohort shares stored by att_gt(). Cohorts are weighted by their population shares in the study population. For fits with reference-date covariates or weights that change within a unit, cohorts are weighted by their masses at the first stored date, and the uncertainty of those shares enters the standard errors. With fix_weights = "base_period" each cell keeps the weights of its own cohort's comparisons, so an aggregate is a declared mixture of cohort effects rather than the effect in one weighted population. See att_gt for the reference-date rules.

Multiplier draws use your session's random number stream. Calling set.seed() before aggte() reproduces them. There is no built-in seed. Draws reuse the worker count requested by the fit's pl and cores request and give the same results as a serial run.

A result fitted with compute_inffunc = FALSE has no influence functions. Refit with compute_inffunc = TRUE to obtain summaries.

Value

For ordinary results, the AGGTEobj returned by did::aggte(). Its call element records the delegated call.

For chained results, a cdid_AGGTEobj (inheriting AGGTEobj) with the overall estimate and its standard error (overall.att, overall.se), the bin estimates, full-frame influence functions that include the cohort-share uncertainty, the requested interval type, and the record of intended, used

and excluded cells. Inspect it with `summary()`, `tidy()` (columns `label`, `available`, `restricted`, `n.intended`, `n.used`, `excluded.ids`, `reason`), `glance()` and `ggdid()`. Plots exist for `"group"`, `"dynamic"` and `"calendar"` aggregates. `"simple"` has no plot.

See Also

`att_gt`, `conditional_did_pretest`, `tidy`, `glance`, `ggdid`, `vignette("cdid", package = "cdid")`

Examples

```
set.seed(1)
out <- cdid::att_gt(yname = "lemp", tname = "year", idname = "countyreal",
                    gname = "first.treat", data = did::mpdta,
                    mode = "ordinary", bstrap = FALSE, cband = FALSE)
cdid::aggte(out, type = "simple")
cdid::aggte(out, type = "group")
```

`att_gt` *Difference-in-differences estimates for panel data*

Description

Estimate average treatment effects by cohort and date or, for a binary treatment that can turn on and off, by switching date and direction. For panels, the default method combines changes measured within units across the dates at which they are observed. Ordinary mode uses **did**.

Usage

```
att_gt(
  yname,
  tname,
  idname = NULL,
  gname,
  xformla = NULL,
  data,
  panel = TRUE,
  allow_unbalanced_panel = NULL,
  control_group = c("nevertreated", "notyettreated"),
  anticipation = 0,
  weightsname = NULL,
  fix_weights = NULL,
  alp = 0.05,
  bstrap = TRUE,
  cband = TRUE,
  biters = 1000,
  clustervars = NULL,
  est_method = "dr",
  base_period = "varying",
```

```
  faster_mode = TRUE,
  print_details = FALSE,
  pl = FALSE,
  cores = 1,
  compute_inffunc = TRUE,
  ...,
  mode = NULL,
  frame = NULL,
  selection = "none",
  selection_model = "pair_logit",
  selection_pooling = "separate",
  selection_xformla = NULL,
  selection_universe = "F",
  selection_rho = NULL,
  selection_justification = NULL,
  selection_est_method = "ipw",
  imputation_pooling = "separate",
  gmm_weighting = c("optimal", "identity", "within"),
  gmm_covariance = "tikhonov",
  pt_assumption = c("post", "all", "ctrl"),
  pt_start = NULL,
  cross_pair = FALSE,
  control = NULL,
  outcome_pooling = "separate",
  dname = NULL,
  imputation_scale = "per_period",
  switchers = "both",
  switch_times = NULL,
  missing_reference = "exclude"
)
```

Arguments

yname	Name of the numeric outcome column. Use NA for missing outcomes.
tname	Name of the numeric date column, on the same scale as the treatment dates.
idname	Name of the unit identifier column. Required in chained mode.
gname	Name of the first-treatment-date column. Use zero for never-treated units. Supply either gname or dname.
xformla	One-sided covariate formula, such as ~ age + income. The default NULL uses an intercept only.
data	A data frame in long format. A panel may have at most one row per unit and date.
panel	Whether the data follow the same units over time. Use FALSE for repeated cross sections.
allow_unbalanced_panel	In ordinary mode, NULL uses **did**'s default of balancing the panel. Chained mode uses incomplete panels with NULL or TRUE. To require a balanced sample, use mode = "ordinary" and allow_unbalanced_panel = FALSE.

`control_group` For cohort models, use never-treated units (`"nevertreated"`) or units not yet treated at the comparison date (`"notyettreated"`).

`anticipation` Number of periods before adoption in which treatment may affect outcomes. The reference date is the last date before that window.

`weightsname` Name of a column of positive sampling weights for cohort effects.

`fix_weights` For weights that vary over time, use `"first_period"` or `"base_period"` in chained mode. These use the first study date or the cohort's reference date, respectively. The default is suitable for constant weights.

`alp` Significance level for confidence intervals and bands.

`bstrap` Whether to use the multiplier bootstrap for inference.

`cband` Whether to report simultaneous confidence bands when `bstrap = TRUE`.

`biters` Number of bootstrap draws.

`clustervars` Names of cluster variables. Chained mode accepts at most one cluster variable in addition to the unit identifier. Clustering affects inference.

`est_method` Treatment adjustment. Use `"dr"` for doubly robust estimation, `"reg"` for outcome regression, or `"ipw"` for inverse probability weighting. Chained cohort estimation also accepts a custom IPW function, described below.

`base_period` Use `"varying"` for adjacent-date comparisons before treatment or `"universal"` for comparisons to the cohort reference date. Both use the same reference date for post-treatment effects. Leave at its default for switching models.

`faster_mode` Passed to **did** in ordinary mode.

`print_details` Whether to print information about the progress of the fit.

`pl`, `cores` Use `pl = TRUE` and set `cores` to run bootstrap calculations on multiple workers. In chained mode, also set `control = list(workers = cores)`.

`compute_inffunc` Set to `FALSE` for point estimates only. Use `TRUE` to obtain standard errors, pretrend tests and `aggte()` summaries.

`...` Additional arguments to a custom estimation function.

`mode` Use `"chained"` for within-unit comparisons or `"ordinary"` to call `did::att_gt()`. By default, panels use chained mode and repeated cross sections use ordinary mode.

`frame` For cohort models, an optional study-unit table with one row per unit, matched to `data` by `idname`. Include units with one or no observed outcomes if they belong to the study population. Without this table, the units in `data` define the population.

`selection` Use `"none"` when units observed at both dates represent their cohort's mean change. Use `"conditional_mtar"` when this representativeness holds conditional on covariates. In cohort models, `"corollary1"` remains an alias for the latter with direct pair logits.

`selection_model` Observation-probability model. `"pair_logit"` fits the probability of observing both dates directly. `"independent_period_logit"` multiplies two date-specific probabilities and assumes conditional independence of observation at the two dates. This argument applies to observation weighting and augmentation.

selection_pooling
: How observation models share slopes. For cohort models use "separate", "within_cohort" or "across_cohorts". For switching models use "separate", "within_group" or "pooled". Within-group sharing keeps switch date, direction and switcher or stayer status fixed. Each pair or date has its own intercept when the formula includes one.

selection_xformla
: One-sided formula for observation probabilities. The default uses xformla. Imputation uses xformla.

selection_universe
: Sample for the observation model. "F" uses all study units. Direct pair weighting also permits "B" (at least one outcome) or "A" (at least two). The target remains the full study population.

selection_rho
: For "A" or "B", an optional column name containing the known, normalized density ratio of covariates in the fitting sample to those in the study population. Supply it in the unit table for cohort models or in data for switching models. The ratio is defined within cohort or within the corresponding switcher and stayer groups. Values must be positive. NULL uses a ratio of one.

selection_justification
: Required explanation when fitting on "A" or "B". Explain why the fitting sample is representative within the cohort or switcher and stayer group, or how the known density ratios were obtained.

selection_est_method
: With selection = "conditional_mtar", choose "ipw" for observation weighting, "imputation" to predict actual changes, or "dr" to combine these approaches. The last two require a built-in treatment method, the full study population ("F"), and complete required covariates for that population. Observation weighting and augmentation need an estimable probability model and overlap.

imputation_pooling
: How regressions of actual changes share slopes. Uses the same choices as selection_pooling, independently of it.

gmm_weighting
: How comparisons are combined. "optimal" (also "two_step") estimates weights using the covariance of the comparisons. "identity" uses equal weights. "within" uses the covariance of within-unit changes implied by uncorrelated level shocks and requires a built-in cohort method.

gmm_covariance
: Use "tikhonov" to regularize singular or nearly singular comparison covariance matrices when needed.

pt_assumption
: Parallel trends assumption for cohorts. "post" starts at each cohort's reference date. "ctrl" adds earlier common trends among eligible controls. "all" also includes the target cohorts in those earlier restrictions. This is separate from base_period.

pt_start
: Common calendar date from which to apply the extra restrictions. Required for "all", optional for "ctrl" (defaulting to the first study date), and unused with "post".

cross_pair
: With a built-in cohort method, set to TRUE to use target and control changes measured over different date pairs when the assumed trends link those changes.

`control` Named list of advanced settings. Use `list(keep_nuisance = TRUE)` to retain fitted model coefficients and standard errors, or `list(preset = "cohort_time_fe")` for the linear cohort-by-date fixed-effects specification described below. Other options include `numeric_workspace_bytes` (default one GiB for numerical arrays), `workers` (default one), `chunk_rows`, `counted_work_limit` (default `Inf`), `condition_trigger` (default `1e-10`) and `ridge_relative` (default `1e-4`).

`outcome_pooling` Use `"separate"` for comparison-specific outcome slopes or `"pooled"` for shared slopes with local intercepts when the formula includes an intercept. For cohort models, pooled slopes scale with the number of calendar steps in a comparison. For switching models, a named character vector or list can assign transition cells to groups that share slopes. Applies to `est_method = "reg"` and `"dr"`.

`dname` Name of the period-specific treatment indicator, coded 0 or 1. With this argument, effects are estimated at entry or exit under a static treatment model with no carryover. Supply a known treatment status for every unit and study date, including dates with missing outcomes.

`imputation_scale` For change imputation in a switching model, `"per_period"` scales covariate slopes by the number of calendar steps in the change. `"per_change"` leaves slopes unscaled.

`switchers` In a switching model, use `"entry"` for changes from 0 to 1, `"exit"` for changes from 1 to 0, or `"both"`.

`switch_times` In a switching model, dates at which to estimate effects. The default uses dates with switchers in the requested directions. A requested date and direction with no switchers is reported as unavailable.

`missing_reference` For chained cohort models with time-varying covariates, how to handle a missing value required at the reference date. `"exclude"` (the default) warns and limits the affected cohort's comparisons to units with the required reference covariates, including those in the observation formula when it is used. `"error"` stops and identifies the missing inputs. Ordinary mode does not use this argument.

Details

Each observed change uses two outcomes from the same unit. Units with fewer than two outcomes can still belong to the study population. Keep them in `data` or supply their information in `frame`. Cohort and population weights then describe the study units you specified. Chaining can connect different sets of units that supply changes over overlapping dates, subject to the comparison-group support and parallel trends assumptions.

For time-varying covariates, post-treatment cohort comparisons use covariates at the cohort's reference date. Comparisons reported before treatment use covariates from the earlier comparison date. Keep the rows providing these values even if their outcomes are missing. When sampling weights vary, supply the required weights at the chosen reference date. For `aggte()`, cohort shares use first-date sampling weights.

When a required reference covariate is missing, the default restricts the affected cohort's comparisons to units with the required covariates at its reference date. A unit can still contribute to another

cohort whose reference covariates it has. Supply independently known values on the appropriate unit-date row, using NA for a missing outcome. Covariates that are constant within a unit are read as usual. If treated units are excluded, the cohort effect describes the remaining treated units, and aggte() uses those units to calculate the cohort's first-date weight. Influence functions retain a row for every original unit. Effects without sufficient support and pretrend comparisons with missing required inputs are reported as unavailable. See fit$chained$reference_eligibility for the affected cohorts, dates and counts. Missing sampling weights at a required reference date still cause an error. If the observation model is fitted on a subset, its transport assumptions and any supplied density ratios apply within each cohort's eligible sample.

The treatment method and the choice of missing-outcome adjustment are separate. Outcome regression for treatment predicts the untreated change. Change imputation predicts the actual change within its group. With augmentation, the package also uses observation probabilities to reweight the residuals of that imputation. With treatment "dr" and observation "dr", either model in each pair can be misspecified if the other is correct and the remaining identifying conditions hold. Extreme probability weights are bounded. Check overlap and clipping before interpreting a fit with large weights, since clipping can affect the target and its robustness properties.

A switching model compares entrants with units staying untreated and exiters with units staying treated. It assumes that past treatment has no effect beyond current treatment status. Exit effects are reported in the treated-minus-untreated direction. This route supports all three built-in treatment methods and missing-outcome adjustments. The cohort-specific trend and cross-pair options apply to gname models.

The linear fixed-effects specification uses control = list(preset = "cohort_time_fe"), consecutive positive integer dates, unit weights, zero anticipation, base_period = "universal" and a never-treated cohort. Use pt_assumption = "post" for never-treated controls, or pt_assumption = "all" with pt_start at the first date for not-yet-treated controls. Any covariates must be time-invariant and available for all study units. Regression uses within-unit least squares. IPW and DR use the common conditional untreated path evaluated in each cohort. Leave outcome_pooling and cross_pair at their defaults.

A custom chained cohort estimator follows the panel IPW interface f(y1, y0, D, covariates, i.weights, inffunc = TRUE, ...). Return a list with a scalar ATT and att.inf.func in the supplied row order. For fitted observation weights, also return log_weight_gradient, the derivative of the estimate with respect to each log weight, including refits inside the function. See the treatment adjustment guide for the callback requirements.

Value

Ordinary mode returns the MP object from **did**. Chained cohort estimation returns a cdid_MP object with cohort-date estimates, uncertainty and comparison support. The joint pretrend test is reported when the full set of requested comparisons and its covariance are usable.

Switching models return a cdid_static_att_gt object with effects by date and direction and a switcher-weighted estimate in $aggregate. Use summary(), tidy() or ggdid() to inspect either type of result. Use aggte() to aggregate cohort effects.

References

Bellego, C., Benatia, D. and Dortet-Bernadet, V. (2025). The Chained Difference-in-Differences. *Journal of Econometrics* 248, 105783. doi:10.1016/j.jeconom.2024.105783

Callaway, B. and Sant'Anna, P. H. C. (2021). Difference-in-Differences with Multiple Time Periods. *Journal of Econometrics* 225(2), 200–230. doi:10.1016/j.jeconom.2020.12.001

de Chaisemartin, C., D'Haultfoeuille, X., Pasquier, F., Sow, D. and Vazquez-Bare, G. (2026). Difference-in-Differences Estimators for Treatments Continuously Distributed at Every Period. Working paper, arXiv:2201.06898v7 for the static binary treatment model.

See Also

`aggte()`, `chain_report()`, `nuisance_models()`, `conditional_did_pretest()`, `vignette("cdid", package = "cdid")`, User guide

Examples

```
data("mpdta", package = "did")
fit <- att_gt(
  yname = "lemp", tname = "year", idname = "countyreal",
  gname = "first.treat", data = mpdta, bstrap = FALSE, cband = FALSE
)
summary(fit)
aggte(fit, type = "dynamic")
```

---

chain_report *Inspect the comparisons used in a chained fit*

Description

See the date pairs and control groups that identify a cohort-date effect and the weights used to combine those comparisons. Supply a chained cohort fit from `att_gt()`.

Usage

```
chain_report(
  object,
  cohort = NULL,
  time = NULL,
  include_pre = FALSE,
  tol = 1e-10,
  group = NULL,
  t = NULL
)

## S3 method for class 'cdid_chain_report'
print(
  x,
  digits = 4,
  show_gmm_weights = FALSE,
  max_cells = 30L,
  ...
```

```
)

## S3 method for class 'cdid_chain_report'
as.data.frame(
  x,
  row.names = NULL,
  optional = FALSE,
  what = c("summary", "witness", "gmm_weights"),
  ...
)
```

Arguments

object	A cdid_MP result produced by att_gt in chained mode.
cohort, group	Optional numeric vector of treatment cohorts to keep. Supply at most one of cohort or group.
time, t	Optional numeric vector of calendar periods to keep. Supply at most one of time or t.
include_pre	Logical scalar (FALSE by default). When FALSE and no period filter is given, only post-treatment cells are included.
tol	Nonnegative numeric threshold (1e-10 by default). GMM moment weights at or below it in absolute value are omitted from the active weight table.
x	A cdid_chain_report object.
digits	Number of decimal places for printed estimates and weights.
show_gmm_weights	Logical. If TRUE, the table of nonzero GMM moment weights is printed beneath each cell.
max_cells	Maximum number of cells printed before truncating.
row.names, optional	As in as.data.frame().
what	Which table to extract: "summary" (one row per cell), "witness" (the identification path of each cell: its date pairs and control pools in order), or "gmm_weights" (all nonzero GMM weights, whose contribution column sums to the reported estimate).
...	Additional arguments (unused).

Details

A cell identified in one step compares the cohort's reference period with the target date directly. A cell that needs chaining passes through intermediate dates, each step using the units observed at both dates of that step. The "witness" table shows one such path. The "gmm_weights" table shows how the fit actually combined all retained moments, which generally involves more comparisons than the single path. Weights reflect the gmm_weighting choice recorded in the fit.

Value

`chain_report()` returns an object of class `"cdid_chain_report"` with elements `summary` (one row per cell), `chains` (a named list holding the `witness` and `gmm_weights` data frames of each cell) and `meta`. `as.data.frame()` returns the requested table as a `data.frame`.

See Also

`att_gt`, `nuisance_models`

Examples

```
set.seed(1)
rows <- list()
uid <- 0L
for (g in 2005:2008) {
  for (i in 1:20) {
    uid <- uid + 1L
    a_i <- rnorm(1)
    for (tv in max(2002L, g - 3L):2008L) {
      rows[[length(rows) + 1L]] <- data.frame(
        id = uid, period = tv, G = g, y = a_i + (tv >= g) * 2 + rnorm(1, sd = 0.2)
      )
    }
  }
}
fit <- att_gt(
  yname = "y", tname = "period", idname = "id", gname = "G",
  data = do.call(rbind, rows),
  control_group = "notyettreated", base_period = "universal",
  cross_pair = TRUE, pt_assumption = "ctrl", gmm_weighting = "within",
  bstrap = FALSE, cband = FALSE
)
cr <- chain_report(fit, cohort = 2005, time = 2007)
print(cr, show_gmm_weights = TRUE)
as.data.frame(cr, what = "gmm_weights")
```

---

conditional_did_pretest

Test for parallel trends before treatment

Description

Test whether pre-treatment differences in trends have zero mean conditional on the supplied covariates. Panels use the chained test by default. Use `mode = "ordinary"` or repeated cross sections to use the test from **did**. The default treatment adjustment is `"ipw"`.

Usage

```
conditional_did_pretest(
  yname,
  tname,
  idname = NULL,
  gname,
  xformla = NULL,
  data,
  panel = TRUE,
  allow_unbalanced_panel = NULL,
  control_group = c("nevertreated", "notyettreated"),
  weightsname = NULL,
  alp = 0.05,
  bstrap = TRUE,
  cband = TRUE,
  biters = 1000,
  clustervars = NULL,
  est_method = "ipw",
  print_details = FALSE,
  pl = FALSE,
  cores = 1,
  anticipation = 0,
  base_period = "varying",
  fix_weights = NULL,
  mode = NULL,
  frame = NULL,
  selection = "none",
  selection_model = "pair_logit",
  selection_pooling = "separate",
  selection_xformla = NULL,
  selection_universe = "F",
  selection_rho = NULL,
  selection_justification = NULL,
  selection_est_method = "ipw",
  imputation_pooling = "separate",
  control = NULL,
  outcome_pooling = "separate"
)

## S3 method for class 'cdid_MP.TEST'
print(x, ...)

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

Arguments

`yname`, `tname`, `idname`, `gname`, `xformla`, `data`, `panel`, `control_group`, `weightsname`, `clustervars`, `print_details`
: As in `did::conditional_did_pretest()` and `att_gt()`. In chained mode the test coordinates are the columns of `model.matrix(xformla)`. Variables that appear only in `selection_xformla` are not test coordinates.

`allow_unbalanced_panel`
: `NULL` or `TRUE` allows incomplete panels in chained mode. Use `mode = "ordinary"` to balance the panel. In ordinary mode, `NULL` uses **did**'s default of `FALSE`.

`alp`
: Significance level of the reported critical value.

`bstrap`, `cband`
: Validated only. As in did, the conditional test always uses multiplier draws and reports no standard errors.

`biters`
: Number of multiplier draws.

`est_method`
: Treatment adjustment of the tested comparisons: `"ipw"` (default), `"reg"` or `"dr"` in chained mode. A custom function returns a result with status `"not_applicable"`, because no conditional process is defined for arbitrary functions.

`pl`, `cores`
: Multiplier draws may use `cores` worker processes when `pl = TRUE` and `control$workers` declares them. The calling session draws the same signs in the same order for every worker count. On Windows, parallel evaluation can change the last digits of the computed statistics through floating-point rounding.

`anticipation`, `base_period`, `fix_weights`, `mode`, `frame`, `selection`, `selection_model`, `selection_pooling`, `selection_xformla`, `selection_universe`, `selection_rho`, `selection_justification`, `selection_est_method`, `imputation_pooling`, `outcome_pooling`, `control`
: The chained settings of `att_gt()`, with the same meanings. They select the fitted route whose pre-treatment comparisons are tested. The `selection` arguments describe the adjustment for missing outcomes and are separate from the treatment adjustment `est_method`. GMM weighting, cross-pair chaining and the `pt_assumption` settings are not test arguments.

`x`, `object`
: A `cdid_MP.TEST` result.

`...`
: Unused.

Details

The null is that every requested pre-treatment comparison has zero mean conditional on the covariate design from `xformla`. The comparison dates follow `base_period`. Reference zeros are excluded. The test uses outcome changes for IPW and outcome residuals for regression or doubly robust adjustment. When a missing-outcome adjustment is requested, the comparison uses the corresponding adjusted changes.

The test evaluates the comparisons over the covariate values of all supplied study units, including units with one or no observed outcomes. Changing covariates are read at the reference date of the comparison. If a required covariate is missing or a prediction is unsupported, the test reports why it is unavailable.

The Cramer-von Mises statistic combines these comparisons. Its p-value and critical value come from multiplier draws over units or the requested clusters. Use `set.seed()` to reproduce the draws.

Clustered testing requires many independent clusters. The unconditional Wald test is reported separately by `att_gt()`.

Value

In chained mode, an object of class `c("cdid_MP.TEST", "MP.TEST")` with the did fields `CvM`, `CvMb`, `CvMcval`, `CvMpval`, `clustervars` and `xformla`. The Kolmogorov-Smirnov fields are `NULL`. The `chained` element records the null, status and reason, base period, tested cells with their scalar values, route, grid rule and columns, statistic and multiplier conventions, clusters, random-number record, execution and resource accounting. Unavailable tests have status `"unavailable"` (or `"not_applicable"` for custom functions) and `NA` statistics. In ordinary mode, did's `MP.TEST` object.

See Also

`att_gt`, `aggte`, `reexports`, `vignette("cdid", package = "cdid")`

Examples

```
## Not run:
pre <- cdid::conditional_did_pretest(yname = "y", tname = "period", idname = "id",
  gname = "G", xformla = ~x, data = panel_data, control_group = "notyettreated",
  biters = 199)
summary(pre)
pre$chained$cells

## End(Not run)
```

---

nuisance_models *Inspect fitted treatment and missing-outcome models*

Description

Show the fitted equations, coefficient estimates and standard errors for the models used by `att_gt()`. Set `control = list(keep_nuisance = TRUE)` when fitting to retain the model information. The report uses the fitted models and does not fit them again.

Usage

```
nuisance_models(fit, component = NULL, block = NULL)
```

Arguments

`fit`	A result of `att_gt()`.
`component`	`NULL` (a compact index of all components) or one of `"selection"`, `"outcome"`, `"treatment"`, `"imputation"`.
`block`	`NULL`, or one fitted block of the component: its stable identifier (`"S1"`, `"S2"`, ..., in block order), its integer position or its label (for example `"within_cohort[g=2004]"`). Supplying `block` alone selects `component = "selection"`.

Details

Call `nuisance_models(fit)` to see which models are available. Select a component, then a block when several equations were fitted. For example, a model with separate slopes can have one block for each date pair and cohort.

`"selection"` Logit for observing a date or both outcomes in a date pair. With date-specific observation probabilities, the pair probability is the product of the two fitted date probabilities.

`"imputation"` Regression of observed actual changes. The estimates are used to predict missing changes within the relevant cohort or switching group.

`"treatment"` Logit for treatment status within the units eligible for the comparison.

`"outcome"` Regression for the counterfactual outcome change. In the linear cohort-by-date model, the report gives coefficients relative to the stated reference levels.

The estimation settings determine which models are fitted. A doubly robust treatment fit with augmented observation correction can report all four models. Some specifications, including custom estimation functions and some models with changing covariates or weights, give a reason when a report is unavailable.

Coefficient standard errors use unit contributions from the fitted model, combined within clusters when clustering was requested. The table can have missing standard errors for a point-only fit, insufficient cluster support, or insufficient numerical workspace. See the reported reason.

Value

With `component = NULL`, an index with model availability and fitted blocks. With a component, an object with fitted-block and coefficient tables. The coefficient table includes `term`, `estimate`, `std_error`, `z_value` and `p_value`. Selecting a block returns its equation, fitting sample, weights and coefficient table. Use `print()` to display the result. A component without a report includes its status and reason.

See Also

`att_gt`, `chain_report`

Examples

```
## Not run:
fit <- att_gt(yname = "y", tname = "t", idname = "id", gname = "g",
              xformla = ~ x + region, data = panel, weightsname = "w",
              est_method = "dr", selection = "conditional_mtar",
              selection_model = "independent_period_logit",
              selection_pooling = "within_cohort",
              control = list(keep_nuisance = TRUE))
nuisance_models(fit)                       # index and template equation
nuisance_models(fit, "selection")          # every block's table
nuisance_models(fit, block = "S1")         # one block
nuisance_models(fit, "outcome")            # status for this specification

# Augmented correction with treatment DR: every fitted role.
fit_aug <- att_gt(yname = "y", tname = "t", idname = "id", gname = "g",
```

```
                  xformla = ~ x + region, data = panel, weightsname = "w",
                  est_method = "dr", selection = "conditional_mtar",
                  selection_est_method = "dr", outcome_pooling = "pooled",
                  control = list(keep_nuisance = TRUE))
nuisance_models(fit_aug, "selection", block = "S1")
nuisance_models(fit_aug, "outcome")              # pooled fit O1
nuisance_models(fit_aug, "treatment", block = "T1")
nuisance_models(fit_aug, "imputation")

## End(Not run)
```

`reexports` *Functions re-exported from did*

Description

Display results from `att_gt()` and `aggte()`. These functions use the same interface as in **did**.

Usage

```
ggdid(object, ...)
tidy(x, ...)
glance(x, ...)
```

Arguments

`object`, `x`	A result returned by `att_gt()` or `aggte()`.
`...`	Arguments passed to the method for the supplied result.

Details

`tidy(x)` Returns a table of effects. Columns include `estimate`, `std.error`, `conf.low` and `conf.high`. For chained fits, `available` and the support or aggregation columns explain which effects the data support.

`glance(x)` Returns a one-row summary of the fit, with information such as the estimation method, the available effects and the type of inference.

`ggdid(object, ...)` Plots the effects from a cohort or switching fit or a cohort, event-time or calendar-time summary. Use `tidy(x)` to view a simple aggregate as a table.

Use `summary(x)` for a printed table and a description of the fit.

See Also

`att_gt`, `aggte`, `conditional_did_pretest`, `vignette("cdid", package = "cdid")`

Index

aggte, *2*, 3, *3*, *7*, *9–11*, *16*, *18*
as.data.frame.cdid_chain_report
        *(*chain_report*)*, 11
att_gt, *2–4*, 5, *5*, *11–13*, *16–18*

cdid *(*cdid-package*)*, 2
cdid-package, 2
chain_report, *3*, 11, *11*, *17*
conditional_did_pretest, *3*, *5*, *11*, 13, *18*

ggdid, *3*, *5*, *10*
ggdid *(*reexports*)*, 18
glance, *3*, *5*
glance *(*reexports*)*, 18

nuisance_models, *3*, *11*, *13*, 16

print.cdid_chain_report *(*chain_report*)*,
        11
print.cdid_MP.TEST
        *(*conditional_did_pretest*)*, 13

reexports, *3*, *16*, 18

summary.cdid_MP.TEST
        *(*conditional_did_pretest*)*, 13

tidy, *3*, *5*, *10*
tidy *(*reexports*)*, 18