

Package: ptetools (via r-universe)

July 20, 2026

Title Panel Treatment Effects Tools

Version 1.0.1

URL <https://github.com/bcallaway11/ptetools>

BugReports <https://github.com/bcallaway11/ptetools/issues>

Description Generic code for estimating treatment effects with panel data. The idea is to break into separate steps organizing the data, looping over groups and time periods, computing group-time average treatment effects, and aggregating group-time average treatment effects. Often, one is able to implement a new identification/estimation procedure by simply replacing the step on estimating group-time average treatment effects. See several different examples of this approach in the package documentation.

Depends R (>= 4.1.0)

License GPL-3

Encoding UTF-8

LazyData true

Imports BMisc (>= 1.4.8), Matrix, ggplot2, DRDID, data.table, dplyr, pbapply, splines2

Suggests testthat (>= 3.0.0), did

Config/testthat/edition 3

Config/roxygen2/version 8.0.0

NeedsCompilation no

Author Brantly Callaway [aut, cre]

Maintainer Brantly Callaway <brantly.callaway@uga.edu>

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

Date/Publication 2026-07-20 12:20:02 UTC

RemoteUrl <https://github.com/cran/ptetools>

RemoteRef HEAD

RemoteSha 97890b4914dc7debcc4e91446799747ec348ca06

Contents

aggte_obj	3
attgt_if	4
attgt_noif	5
attgt_pte_aggregations	5
autoplot.dose_obj	6
autoplot.ptemp_boot	6
autoplot.pteqtt	7
autoplot.pteresults	8
compute.ptemp	8
covid_attgt	9
covid_data	10
crit_val_checks	11
did_attgt	12
did_rcs_attgt	12
dose_obj	13
ggpte	14
ggpte_cont	15
group_time_att	15
gt_data_frame	16
keep_all_pretreatment_subset	17
keep_all_untreated_subset	17
mboot2	18
overall_weights	19
panel_empirical_bootstrap	19
plot.dose_obj	21
plot.ptemp_boot	22
plot.pteqtt	22
plot.pteresults	23
process_att_gt	23
process_dose_gt	24
pte	24
pte_aggte	28
pte_attgt	29
pte_default	30
pte_dose_results	32
ptemp_boot	33
pte_params	34
pteqtt	36
pteresults	36
qott_pte_aggregations	37
qtt_empirical_bootstrap	37
qtt_pte_aggregations	39
setup_pte	40
setup_pte_basic	42
two_by_two_rcs_subset	43
two_by_two_subset	44

aggte_obj

*Aggregated Treatment Effects Class***Description**

Objects of this class hold results on aggregated group-time average treatment effects. This is derived from the AGGTEobj class in the did package.

An object for holding aggregated treatment effect parameters.

Usage

```
aggte_obj(
  overall.att = NULL,
  overall.se = NULL,
  type = "simple",
  egt = NULL,
  att.egt = NULL,
  se.egt = NULL,
  crit.val.egt = NULL,
  inf.function = NULL,
  min_e = NULL,
  max_e = NULL,
  balance_e = NULL,
  DIDparams = NULL
)
```

Arguments

<code>overall.att</code>	The estimated overall ATT
<code>overall.se</code>	Standard error for overall ATT
<code>type</code>	The type of aggregation to be done. Default is "overall".
<code>egt</code>	Holds the length of exposure (for dynamic effects), the group (for selective treatment timing), or the time period (for calendar time effects)
<code>att.egt</code>	The ATT specific to egt
<code>se.egt</code>	The standard error specific to egt
<code>crit.val.egt</code>	A critical value for computing uniform confidence bands for dynamic effects, selective treatment timing, or time period effects.
<code>inf.function</code>	The influence function of the chosen aggregated parameters
<code>min_e</code>	The minimum event time computed in the event study results. This is useful when there are a huge number of pre-treatment periods.
<code>max_e</code>	The maximum event time computed in the event study results. This is useful when there are a huge number of post-treatment periods.

balance_e	Drops groups that do not have at least balance_e periods of post-treatment data. This keeps the composition of groups constant across different event times in an event study. Default is NULL, in which case this is ignored.
DIDparams	A DIDparams object

Value

an aggte_obj

attgt_if	<i>Class for (g,t)-Specific Results with Influence Function</i>
----------	---

Description

Class for holding group-time average treatment effects along with their influence function

Usage

```
attgt_if(attgt, inf_func, extra_gt_returns = NULL)
```

Arguments

attgt	group-time average treatment effect
inf_func	influence function
extra_gt_returns	A place to return anything extra from particular group-time average treatment effect calculations. For DID, this might be something like propensity score estimates, regressions of untreated potential outcomes on covariates. For ife, this could be something like the first step regression 2sls estimates. This argument is also potentially useful for debugging.

Value

attgt_if object

attgt_noif

Class for (g,t)-Specific Results without Influence Function

Description

Class for holding returns from group-time specific estimates in settings when an influence function is not returned

Usage

```
attgt_noif(attgt, extra_gt_returns = NULL)
```

Arguments

attgt group-time average treatment effect

extra_gt_returns

A place to return anything extra from particular group-time average treatment effect calculations. For DID, this might be something like propensity score estimates, regressions of untreated potential outcomes on covariates. For ife, this could be something like the first step regression 2sls estimates. This argument is also potentially useful for debugging.

Value

an attgt_noif object

attgt_pte_aggregations

Aggregate Group-Time Average Treatment Effects

Description

Aggregate group-time average treatment effects into overall, group, and dynamic effects. This function is only used for (i) computing standard errors using the empirical bootstrap, and (ii) combining distributions at the (g,t) level

Usage

```
attgt_pte_aggregations(attgt.list, ptep)
```

Arguments

attgt.list list of attgt results from compute_pte

ptep pte_params object

Value

pte_emp_boot object

autoplot.dose_obj *autoplot.dose_obj*

Description

Plot dose-specific results for a continuous treatment.

Usage

```
## S3 method for class 'dose_obj'
autoplot(object, type = "att", ...)
```

Arguments

object	a dose_obj object
type	whether to plot "att" (default) or "acrt"
...	unused

Value

a ggplot object

autoplot.pte_emp_boot *autoplot.pte_emp_boot*

Description

Event-study plot for a pte_emp_boot object returned by empirical-bootstrap estimators (e.g., `cic()`, `qdid()`, `mdid()`). Pre- and post-treatment periods are distinguished by color.

Usage

```
## S3 method for class 'pte_emp_boot'
autoplot(object, ...)
```

Arguments

object	a pte_emp_boot object
...	unused

Value

a ggplot object

autoplot.pte_qtt	<i>autoplot.pte_qtt</i>
------------------	-------------------------

Description

Plot a `pte_qtt` object.

For `type = "overall"`: QTT curve with quantile on the x-axis.

For `type = "dynamic"`: event-study plot with event time on the x-axis. Each selected quantile is a separate colored line. CIs are shown by default when a single quantile is plotted, and suppressed by default when multiple quantiles are plotted.

Usage

```
## S3 method for class 'pte_qtt'
autoplot(
  object,
  type = "overall",
  cband = TRUE,
  plot_probs = 0.5,
  plot_ci = NULL,
  ...
)
```

Arguments

<code>object</code>	a <code>pte_qtt</code> object
<code>type</code>	which aggregation to plot: "overall" (default) or "dynamic". "group" is a stub.
<code>cband</code>	logical; if TRUE (default), show uniform confidence band; if FALSE, show point-wise intervals. Applies when CIs are displayed.
<code>plot_probs</code>	numeric vector of quantile levels to show in the dynamic plot. Defaults to 0.5 (median). All values must be present in <code>object\$dynamic\$probs</code> .
<code>plot_ci</code>	logical or NULL. If NULL (default), CIs are shown when <code>length(plot_probs) == 1</code> and suppressed otherwise. Set TRUE to always show CIs, FALSE to never show them.
<code>...</code>	unused

Value

a ggplot object

autoplot.pte_results	<i>autoplot.pte_results</i>
----------------------	-----------------------------

Description

Event-study plot for a pte_results object. Pre- and post-treatment periods are distinguished by color.

Usage

```
## S3 method for class 'pte_results'
autoplot(object, ...)
```

Arguments

object	a pte_results object
...	unused

Value

a ggplot object

compute.pte	<i>Heavy-Lifting for pte Function</i>
-------------	---------------------------------------

Description

Function that actually computes panel treatment effects. The difference relative to compute.pte is that this function loops over time periods first (instead of groups) and tries to estimate model for untreated potential outcomes jointly for all groups.

Usage

```
compute.pte(pte, subset_fun, attgt_fun, ...)
```

Arguments

pte	pte_params object
subset_fun	This is a function that should take in data, g (for group), tp (for time period), and ... and be able to return the appropriate data.frame that can be used by attgt_fun to produce ATT(g=g, t=tp). The data frame should be constructed using gt_data_frame in order to guarantee that it has the appropriate columns that identify which group an observation belongs to, etc.

attgt_fun	This is a function that should work in the case where there is a single group and the "right" number of time periods to recover an estimate of the ATT. For example, in the context of difference in differences, it would need to work for a single group, find the appropriate comparison group (untreated units), find the right time periods (pre- and post-treatment), and then recover an estimate of ATT for that group. It will be called over and over separately by groups and by time periods to compute $ATT(g,t)$'s. The function needs to work in a very specific way. It should take in the arguments: data, . . . data should be constructed using the function <code>gt_data_frame</code> which checks to make sure that data has the correct columns defined. . . . are additional arguments (such as formulas for covariates) that <code>attgt_fun</code> needs. From these arguments <code>attgt_fun</code> must return a list with element <code>ATT</code> containing the group-time average treatment effect for that group and that time period. If <code>attgt_fun</code> returns an influence function (which should be provided in a list element named <code>inf_func</code>), then the code will use the multiplier bootstrap to compute standard errors for group-time average treatment effects, an overall treatment effect parameter, and a dynamic treatment effect parameter (i.e., event study parameter). If <code>attgt_fun</code> does not return an influence function, then the same objects will be computed using the empirical bootstrap. This is usually (perhaps substantially) easier to code, but also will usually be (perhaps substantially) computationally slower.
. . .	extra arguments that can be passed to create the correct subsets of the data (depending on <code>subset_fun</code>), to estimate group time average treatment effects (depending on <code>attgt_fun</code>), or to aggregating treatment effects (particularly useful are <code>min_e</code>, <code>max_e</code>, and <code>balance_e</code> arguments to event study aggregations)

Value

a list containing the following elements:

- `attgt.list`: list of $ATT(g,t)$ estimates
- `inffunc`: influence function matrix
- `extra_gt_returns`: list of extra returns from gt-specific calculations

covid_attgt

Covid $ATT(g,t)$ Estimator

Description

Computes a group-time average treatment effect and influence function using an unconfoundedness-type identification strategy. This estimator is appropriate when parallel trends is implausible but a selection-on-observables assumption holds in levels (rather than differences) — e.g., during the early COVID-19 pandemic.

Originally from Callaway and Li (2021). Moved into `ptetools` from the `ppe` package.

Usage

```
covid_atttgt(gt_data, xformula, d_outcome = FALSE, d_covs_formula = ~-1, ...)
```

Arguments

<code>gt_data</code>	data that is "local" to a particular group-time average treatment effect, structured as a <code>gt_data_frame</code>
<code>xformula</code>	one-sided formula for covariates used in the propensity score and outcome regression models
<code>d_outcome</code>	logical; if TRUE, use first-differenced outcomes. Default is FALSE (levels).
<code>d_covs_formula</code>	one-sided formula for covariates to include as changes (differences). Default is <code>~-1</code> (no change covariates).
<code>...</code>	extra arguments; not used

Value

`atttgt_if` object

References

Callaway, B. and Li, T. (2021). Policy Evaluation during a Pandemic. <https://arxiv.org/abs/2105.06927>

<code>covid_data</code>	<i>State-level Covid-19 Data</i>
-------------------------	----------------------------------

Description

A panel dataset containing Covid-19 related data for 46 states. This data comes from Callaway and Li (2021). See the paper for additional descriptions.

Usage

```
covid_data
```

Format

A data frame with 1656 rows and 9 variables:

positive The cumulative number of cases per million individuals in a particular state by a particular time period.

time.period Time period

group The group that a state belongs to. It is based on the time period when they enacted the shelter-in-place order.

state State abbreviation

totalTestResults The total Covid-19 number of tests run per million individuals in a particular state by a particular time period.

state_id Numeric state identifier

region Census region for particular state

retail_and_recreation_percent_change_from_baseline The percentage change in retail and recreational travel from pre-Covid baseline. This is from Google's Mobility report (see paper for details).

current The current number of cases per million individuals in a particular state by a particular time period. This variable is constructed from positive (see paper for details).

Source

Callaway and Li (2021)

crit_val_checks	<i>Sanity Checks on Critical Values</i>
-----------------	---

Description

A function to perform sanity checks and possibly adjust a critical value to form a uniform confidence band

Usage

```
crit_val_checks(crit_val, alp = 0.05)
```

Arguments

crit_val	the critical value
alp	the significance level

Value

a (possibly adjusted) critical value

did_attgt	<i>Difference-in-differences for ATT(g,t)</i>
-----------	---

Description

Takes a `data.frame` and computes for a particular group `g` and time period `t` and computes an estimate of a group time average treatment effect and a corresponding influence function using a difference in differences approach.

The code relies on `gt_data` having certain variables defined. In particular, there should be an `id` column (individual identifier), `D` (treated group identifier), `period` (time period), `name` (equal to "pre" for pre-treatment periods and equal to "post" for post treatment periods), `Y` (outcome).

In our case, we call `two_by_two_subset` which sets up the data to have this format before the call to `did_attgt`.

Usage

```
did_attgt(gt_data, xformula = ~1, ...)
```

Arguments

<code>gt_data</code>	data that is "local" to a particular group-time average treatment effect
<code>xformula</code>	one-sided formula for covariates used in the propensity score and outcome regression models
<code>...</code>	extra function arguments; not used here

Value

`attgt_if`

did_rcs_attgt	<i>Repeated Cross Sections Difference-in-Differences for ATT(g,t)</i>
---------------	---

Description

Takes a local repeated cross sections data set and computes an estimate of a group-time average treatment effect and corresponding influence function using a repeated cross sections DID approach.

Usage

```
did_rcs_attgt(gt_data, xformula = ~1, est_method = "dr", ...)
```

Arguments

gt_data	data that is "local" to a particular group-time average treatment effect
xformula	one-sided formula for covariates used in the propensity score and outcome regression models
est_method	Which type of estimation method to use. Default is "dr" for doubly robust. The other option is "reg" for regression adjustment.
...	extra function arguments; not used here

Value

attgt_if

dose_obj

Class for Continuous Treatments

Description

Holds results from computing dose-specific treatment effects with a continuous treatment

Usage

```
dose_obj(
  dose,
  overall_att = NULL,
  overall_att_se = NULL,
  overall_att_inffunc = NULL,
  overall_acrt = NULL,
  overall_acrt_se = NULL,
  overall_acrt_inffunc = NULL,
  att.d = NULL,
  att.d_se = NULL,
  att.d_crit.val = NULL,
  att.d_inffunc = NULL,
  acrt.d = NULL,
  acrt.d_se = NULL,
  acrt.d_crit.val = NULL,
  acrt.d_inffunc = NULL,
  pte_params = NULL
)
```

Arguments

dose	vector containing the values of the dose used in estimation
overall_att	estimate of the overall ATT, the mean of ATT(D) given $D > 0$
overall_att_se	the standard error of the estimate of overall_att

<code>overall_att_inffunc</code>	the influence function for estimating <code>overall_att</code>
<code>overall_acrt</code>	estimate of the overall ACRT, the mean of <code>ACRT(D D)</code> given $D > 0$
<code>overall_acrt_se</code>	the standard error for the estimate of <code>overall_acrt</code>
<code>overall_acrt_inffunc</code>	the influence function for estimating <code>overall_acrt</code>
<code>att.d</code>	estimates of <code>ATT(d)</code> for each value of dose
<code>att.d_se</code>	standard error of <code>ATT(d)</code> for each value of dose
<code>att.d_crit.val</code>	critical value to produce pointwise or uniform confidence interval for <code>ATT(d)</code>
<code>att.d_inffunc</code>	matrix containing the influence function from estimating <code>ATT(d)</code>
<code>acrt.d</code>	estimates of <code>ACRT(d)</code> for each value of dose
<code>acrt.d_se</code>	standard error of <code>ACRT(d)</code> for each value of dose
<code>acrt.d_crit.val</code>	critical value to produce pointwise or uniform confidence interval for <code>ACRT(d)</code>
<code>acrt.d_inffunc</code>	matrix containing the influence function from estimating <code>ACRT(d)</code>
<code>pte_params</code>	a <code>pte_params</code> object containing other parameters passed to the function

Value

a `dose_obj` object

`ggpte`

ggpte

Description

Deprecated. Use `autoplot()` on the `pte_results` object instead.

Usage

```
ggpte(pte_results)
```

Arguments

`pte_results` a `pte_results` object

Value

a `ggplot` object

ggpte_cont	<i>ggpte_cont</i>
------------	-------------------

Description

Deprecated. Use autoplot() on the dose_obj instead.

Usage

```
ggpte_cont(dose_obj, type = "att")
```

Arguments

dose_obj	a dose_obj object
type	whether to plot "att" (default) or "acrt"

Value

a ggplot object

group_time_att	<i>Class for Estimates across Groups and Time</i>
----------------	---

Description

Class that holds causal effect parameter estimates across timing groups and time periods

Usage

```
group_time_att(
  group,
  time.period,
  att,
  V_analytical,
  se,
  crit_val,
  inf_func,
  n,
  W,
  Wpval,
  cband,
  alp,
  ptep,
  extra_gt_returns
)
```

Arguments

group	numeric vector of groups for ATT(g,t)
time.period	numeric vector of time periods for ATT(g,t)
att	numeric vector containing the value of ATT(g,t) for corresponding group and time period
V_analytical	analytical asymptotic variance matrix for ATT(g,t)'s
se	numeric vector of standard errors
crit_val	critical value (usually a critical value for conducting uniform inference)
inf_func	matrix of influence function
n	number of unique individuals
W	Wald statistic for ATT(g,t) version of pre-test of parallel trends assumption
Wpval	p-value for Wald pre-test of ATT(g,t) version of parallel trends assumption
cband	logical indicating whether or not to report a confidence band
alp	significance level
ptep	pte_params object
extra_gt_returns	list containing extra returns at the group-time level

Value

object of class group_time_att

gt_data_frame	<i>Convert Data to Usable Format</i>
---------------	--------------------------------------

Description

Checks and converts data to satisfy criteria to be used in internal ptetools functions. In particular, the function takes in a data.frame, checks if it has the right columns to be used to calculate a group-time average treatment effect, and sets the class of the data.frame to include gt_data_frame

Usage

```
gt_data_frame(data)
```

Arguments

data	data that will be checked to see if has right format for computing group-time average treatment effects
------	---

Value

gt_data_frame object

keep_all_pretreatment_subset

Keep All Pre-Treatment Subset

Description

A function that takes an original data set and keeps all data for all groups that are not-yet-treated by period `tp` as well as for group `g`.

In particular, this keeps more data than functions like `two_by_two_subset` that use a fixed base period.

A main use case for this function is the interactive fixed effects approach proposed in Callaway and Tsyawo (2023).

Usage

```
keep_all_pretreatment_subset(data, g, tp, ...)
```

Arguments

<code>data</code>	the full dataset
<code>g</code>	the current group
<code>tp</code>	the current time period
<code>...</code>	additional arguments

Value

list that contains the following elements:

- `gt_data`: a `gt_data_frame` object that contains the correct subset of data
- `n1`: the number of observations in this subset
- `disidx`: a vector of the correct ids for this subset

keep_all_untreated_subset

Keep All Untreated Subset

Description

A function that takes an original data set and keeps all pre-treatment data for all groups. For group `g`, it also includes data for the current period.

Also, note that if `tp` is still a pre-treatment period for group `g`, then periods after `tp` will also be dropped for group `g`. This is a design choice and is useful especially for estimating placebo group-time average treatment effects in pre-treatment periods.

A main use case for this function is to compute $ATT(g,t)$'s using a global estimation strategy such as imputation in Gardner (2022).

Usage

```
keep_all_untreated_subset(data, g, tp, ...)
```

Arguments

data	the full dataset
g	the current group
tp	the current time period
...	extra arguments to get the subset correct

Value

list that contains the following elements:

- `gt_data`: a `gt_data_frame` object that contains the correct subset of data
- `n1`: the number of observations in this subset
- `disidx`: a vector of the correct ids for this subset

mboot2

Multiplier Bootstrap

Description

Function for using multiplier bootstrap to conduct inference

Usage

```
mboot2(inffunc, biters = 1000, alp = 0.05)
```

Arguments

inffunc	influence function matrix
biters	number of bootstrap iterations; default is 100
alp	significance level; default is 0.05

Value

list with the following elements:

- `boot_se`: bootstrap standard errors
- `crit_val`: critical value for uniform confidence bands

overall_weights	<i>Weights for Overall Aggregation</i>
-----------------	--

Description

A function that returns weights on (g,t)'s to deliver overall (averaged across groups and time periods) treatment effect parameters

Usage

```
overall_weights(attgt, balance_e = NULL, min_e = -Inf, max_e = Inf, ...)
```

Arguments

attgt	A group_time_att object to be aggregated
balance_e	Drops groups that do not have at least balance_e periods of post-treatment data. This keeps the composition of groups constant across different event times in an event study. Default is NULL, in which case this is ignored.
min_e	The minimum event time computed in the event study results. This is useful when there are a huge number of pre-treatment periods.
max_e	The maximum event time computed in the event study results. This is useful when there are a huge number of post-treatment periods.
...	extra arguments

Value

a data.frame containing columns:

- group: the group
- time.period: the time period
- overall_weight: the weight

panel_empirical_bootstrap	<i>Panel Empirical Bootstrap</i>
---------------------------	----------------------------------

Description

Computes empirical bootstrap pointwise standard errors

Usage

```
panel_empirical_bootstrap(
  attgt.list,
  ptep,
  setup_pte_fun,
  subset_fun,
  attgt_fun,
  extra_gt_returns,
  aggte_fun = NULL,
  ...
)
```

Arguments

attgt.list	list of attgt results from compute_pte
ptep	pte_params object
setup_pte_fun	This is a function that should take in data, yname (the name of the outcome variable in data), gname (the name of the group variable), idname (the name of the id variable), and possibly other arguments such as the significance level alp, the number of bootstrap iterations biters, and how many clusters for parallel computing in the bootstrap cl. The key thing that needs to be figured out in this function is which groups and time periods ATT(g,t) should be computed in. The function should return a pte_params object which contains all of the parameters passed into the function as well as glist and tlist which should be ordered lists of groups and time periods for ATT(g,t) to be computed. This function provides also provides a good place for error handling related to the types of data that can be handled. The pte package contains the function setup_pte that is a lightweight function that basically just takes the data, omits the never-treated group from glist but includes all other groups and drops the first time period. This works in cases where ATT would be identified in the 2x2 case (i.e., where there are two time periods, no units are treated in the first period and the identification strategy "works" with access to a treated and untreated group and untreated potential outcomes for both groups in the first period) — for example, this approach works if DID is the identification strategy.
subset_fun	This is a function that should take in data, g (for group), tp (for time period), and ... and be able to return the appropriate data.frame that can be used by attgt_fun to produce ATT(g=g, t=tp). The data frame should be constructed using gt_data_frame in order to guarantee that it has the appropriate columns that identify which group an observation belongs to, etc.
attgt_fun	This is a function that should work in the case where there is a single group and the "right" number of time periods to recover an estimate of the ATT. For example, in the contest of difference in differences, it would need to work for a single group, find the appropriate comparison group (untreated units), find the right time periods (pre- and post-treatment), and then recover an estimate of ATT for that group. It will be called over and over separately by groups and by time periods to compute ATT(g,t)'s.

The function needs to work in a very specific way. It should take in the arguments: `data, ...` `data` should be constructed using the function `gt_data_frame` which checks to make sure that `data` has the correct columns defined. `...` are additional arguments (such as formulas for covariates) that `attgt_fun` needs. From these arguments `attgt_fun` must return a list with element `ATT` containing the group-time average treatment effect for that group and that time period. If `attgt_fun` returns an influence function (which should be provided in a list element named `inf_func`), then the code will use the multiplier bootstrap to compute standard errors for group-time average treatment effects, an overall treatment effect parameter, and a dynamic treatment effect parameter (i.e., event study parameter). If `attgt_fun` does not return an influence function, then the same objects will be computed using the empirical bootstrap. This is usually (perhaps substantially) easier to code, but also will usually be (perhaps substantially) computationally slower.

<code>extra_gt_returns</code>	A place to return anything extra from particular group-time average treatment effect calculations. For DID, this might be something like propensity score estimates, regressions of untreated potential outcomes on covariates. For ife, this could be something like the first step regression 2sls estimates. This argument is also potentially useful for debugging.
<code>aggte_fun</code>	An optional function for aggregating group-time treatment effects. When <code>NULL</code> (the default), the function is selected automatically based on <code>gt_type</code> .
<code>...</code>	extra arguments that can be passed to create the correct subsets of the data (depending on <code>subset_fun</code>), to estimate group time average treatment effects (depending on <code>attgt_fun</code>), or to aggregating treatment effects (particularly useful are <code>min_e</code> , <code>max_e</code> , and <code>balance_e</code> arguments to event study aggregations)

Value

`pte_emp_boot` object

<code>plot.dose_obj</code>	<i>plot.dose_obj</i>
----------------------------	----------------------

Description

Convenience wrapper around `autoplot.dose_obj`.

Usage

```
## S3 method for class 'dose_obj'
plot(x, ...)
```

Arguments

<code>x</code>	a <code>dose_obj</code> object
<code>...</code>	passed to <code>autoplot.dose_obj</code>

Value

invisibly returns the ggplot object

plot.pte_emp_boot	<i>plot.pte_emp_boot</i>
-------------------	--------------------------

Description

Convenience wrapper around [autoplot.pte_emp_boot](#).

Usage

```
## S3 method for class 'pte_emp_boot'
plot(x, ...)
```

Arguments

x	a pte_emp_boot object
...	passed to autoplot.pte_emp_boot

Value

invisibly returns the ggplot object

plot.pte_qtt	<i>plot.pte_qtt</i>
--------------	---------------------

Description

Convenience wrapper around [autoplot.pte_qtt](#).

Usage

```
## S3 method for class 'pte_qtt'
plot(x, type = "overall", cband = TRUE, plot_probs = 0.5, plot_ci = NULL, ...)
```

Arguments

x	a pte_qtt object
type	which aggregation to plot. See autoplot.pte_qtt .
cband	logical; if TRUE (default), show uniform confidence band.
plot_probs	numeric vector of quantile levels to show. See autoplot.pte_qtt .
plot_ci	logical or NULL. See autoplot.pte_qtt .
...	passed to autoplot.pte_qtt

Value

invisibly returns the ggplot object

plot.pte_results	<i>plot.pte_results</i>
------------------	-------------------------

Description

Convenience wrapper around [autoplot.pte_results](#).

Usage

```
## S3 method for class 'pte_results'
plot(x, ...)
```

Arguments

x	a pte_results object
...	passed to autoplot.pte_results

Value

invisibly returns the ggplot object

process_att_gt	<i>Process ATT(g,t) Results</i>
----------------	---------------------------------

Description

Process ATT(g,t) results when influence function is available

Usage

```
process_att_gt(att_gt_results, ptep)
```

Arguments

att_gt_results	ATT(g,t)'s
ptep	pte_params object

Value

group_time_att object

process_dose_gt	<i>Process Results with a Continuous Treatment</i>
-----------------	--

Description

After computing results for each group and time period, `process_dose_gt` combines/averages them into overall effects and/or dose specific effects. This is generic code that can be used from different ways of estimating causal effects across different timing groups and periods in a previous step.

Usage

```
process_dose_gt(gt_results, ptep, ...)
```

Arguments

<code>gt_results</code>	list of group-time specific results
<code>ptep</code>	<code>pte_params</code> object
<code>...</code>	extra arguments

Value

a `dose_obj` object

pte	<i>Panel Treatment Effects</i>
-----	--------------------------------

Description

Tools for estimating treatment effects with panel data.
Main function for computing panel treatment effects

Usage

```
pte(
  yname,
  gname,
  tname,
  idname = NULL,
  data,
  setup_pte_fun,
  subset_fun,
  attgt_fun,
  aggte_fun = NULL,
  panel = TRUE,
```



```

cband = TRUE,
alp = 0.05,
boot_type = "multiplier",
weightsname = NULL,
gt_type = "att",
ret_quantile = NULL,
global_fun = FALSE,
time_period_fun = FALSE,
group_fun = FALSE,
process_dtt_gt_fun = process_dtt_gt,
process_dose_gt_fun = process_dose_gt,
probs = NULL,
biters = 100,
cl = 1,
call = NULL,
...
)

```

Arguments

yname	Name of outcome in data
gname	Name of group in data
tname	Name of time period in data
idname	Name of id in data
data	balanced panel or repeated cross sections data
setup_pte_fun	This is a function that should take in data, yname (the name of the outcome variable in data), gname (the name of the group variable), idname (the name of the id variable), and possibly other arguments such as the significance level alp, the number of bootstrap iterations biters, and how many clusters for parallel computing in the bootstrap cl. The key thing that needs to be figured out in this function is which groups and time periods ATT(g,t) should be computed in. The function should return a pte_params object which contains all of the parameters passed into the function as well as glist and tlist which should be ordered lists of groups and time periods for ATT(g,t) to be computed. This function provides also provides a good place for error handling related to the types of data that can be handled. The pte package contains the function setup_pte that is a lightweight function that basically just takes the data, omits the never-treated group from glist but includes all other groups and drops the first time period. This works in cases where ATT would be identified in the 2x2 case (i.e., where there are two time periods, no units are treated in the first period and the identification strategy "works" with access to a treated and untreated group and untreated potential outcomes for both groups in the first period) — for example, this approach works if DID is the identification strategy.
subset_fun	This is a function that should take in data, g (for group), tp (for time period), and ... and be able to return the appropriate data.frame that can be used by attgt_fun to produce ATT(g=g,t=tp). The data frame should be constructed

	using <code>gt_data_frame</code> in order to guarantee that it has the appropriate columns that identify which group an observation belongs to, etc.
<code>attgt_fun</code>	This is a function that should work in the case where there is a single group and the "right" number of time periods to recover an estimate of the ATT. For example, in the context of difference in differences, it would need to work for a single group, find the appropriate comparison group (untreated units), find the right time periods (pre- and post-treatment), and then recover an estimate of ATT for that group. It will be called over and over separately by groups and by time periods to compute $ATT(g,t)$'s. The function needs to work in a very specific way. It should take in the arguments: <code>data</code>, . . . <code>data</code> should be constructed using the function <code>gt_data_frame</code> which checks to make sure that <code>data</code> has the correct columns defined. . . . are additional arguments (such as formulas for covariates) that <code>attgt_fun</code> needs. From these arguments <code>attgt_fun</code> must return a list with element <code>ATT</code> containing the group-time average treatment effect for that group and that time period. If <code>attgt_fun</code> returns an influence function (which should be provided in a list element named <code>inf_func</code>), then the code will use the multiplier bootstrap to compute standard errors for group-time average treatment effects, an overall treatment effect parameter, and a dynamic treatment effect parameter (i.e., event study parameter). If <code>attgt_fun</code> does not return an influence function, then the same objects will be computed using the empirical bootstrap. This is usually (perhaps substantially) easier to code, but also will usually be (perhaps substantially) computationally slower.
<code>aggte_fun</code>	An optional function for aggregating group-time treatment effects.
<code>panel</code>	Whether the data are panel data. The default is <code>TRUE</code> . Set to <code>FALSE</code> for repeated cross sections.
<code>cband</code>	whether or not to report a uniform (instead of pointwise) confidence band (default is <code>TRUE</code>)
<code>alp</code>	significance level; default is 0.05
<code>boot_type</code>	should be one of "multiplier" (the default) or "empirical". The multiplier bootstrap is generally much faster, but <code>attgt_fun</code> needs to provide an expression for the influence function (which could be challenging to figure out). If no influence function is provided, then the <code>pte</code> package will use the empirical bootstrap no matter what the value of this parameter.
<code>weightsname</code>	The name of the column that contains sampling weights. The default is <code>NULL</code> , in which case no sampling weights are used.
<code>gt_type</code>	which type of group-time effects are computed. The default is "att". Different estimation strategies can implement their own choices for <code>gt_type</code>
<code>ret_quantile</code>	For functions that compute quantile treatment effects, this is a specific quantile at which to report results, e.g., <code>ret_quantile = 0.5</code> will return that the <code>qte</code> at the median.
<code>global_fun</code>	Logical indicating whether or not untreated potential outcomes can be estimated in one shot, i.e., for all groups and time periods. Main use case would be for one-shot imputation estimators. Not supported yet.

<code>time_period_fun</code>	Logical indicating whether or not untreated potential outcomes can be estimated for all groups in the same time period. Not supported yet.
<code>group_fun</code>	Logical indicating whether or not untreated potential outcomes can be estimated for all time periods for a single group. Not supported yet. These functions aim at reducing or eliminating running the same code multiple times.
<code>process_dtt_gt_fun</code>	An optional function to customize results when the <code>gt</code> -specific function returns the distribution of treated and untreated potential outcomes. The default is <code>process_dtt_gt</code> , which is a function provided by the package. See that function for an example of what this function should return. This is unused except in cases where the results involve distributions.
<code>process_dose_gt_fun</code>	An optional function to customize results when the <code>gt</code> -specific function returns treatment effects that depend on dose (i.e., amount of the treatment). The default is <code>process_dose_gt</code> , which is a function provided by the package. See that function for an example of what this function should return. This is unused except in cases where the results involve doses.
<code>probs</code>	For <code>gt_type = "qtt"</code> , a numeric vector of quantile levels at which to evaluate the QTT curve. Defaults to <code>seq(0.05, 0.95, 0.05)</code> .
<code>biters</code>	number of bootstrap iterations; default is 100
<code>cl</code>	number of clusters to be used when bootstrapping; default is 1
<code>call</code>	keeps track of through the <code>call</code> from external functions/packages
<code>...</code>	extra arguments that can be passed to create the correct subsets of the data (depending on <code>subset_fun</code>), to estimate group time average treatment effects (depending on <code>attgt_fun</code>), or to aggregating treatment effects (particularly useful are <code>min_e</code> , <code>max_e</code> , and <code>balance_e</code> arguments to event study aggregations)

Value

`pte_results` object

Author(s)

Maintainer: Brantly Callaway <brantly.callaway@uga.edu>

Authors:

- Brantly Callaway <brantly.callaway@uga.edu>

See Also

Useful links:

- <https://github.com/bcallaway11/ptetools>
- Report bugs at <https://github.com/bcallaway11/ptetools/issues>

Examples

```
# example using minimum wage data
# and difference-in-differences identification strategy
library(did)
data(mpdta)
did_res <- pte(
  yname = "lemp",
  gname = "first.treat",
  tname = "year",
  idname = "countyreal",
  data = mpdta,
  setup_pte_fun = setup_pte,
  subset_fun = two_by_two_subset,
  attgt_fun = did_attgt,
  xformula = ~lpop
)

summary(did_res)
ggplot2::autoplot(did_res)
```

pte_aggte

Aggregates (g,t)-Specific Results

Description

This is a slight edit of the aggte function from the did package. Currently, it only provides aggregations for "overall" treatment effects and event studies. It also will provide the weights directly which is currently used for constructing aggregations based on distributions. The other difference is that, pte_aggte provides inference results where the only randomness is coming from the outcomes (not from the group assignment nor from the covariates).

Usage

```
pte_aggte(
  attgt,
  type = "overall",
  balance_e = NULL,
  min_e = -Inf,
  max_e = Inf,
  ...
)
```

Arguments

attgt	A group_time_att object to be aggregated
type	The type of aggregation to be done. Default is "overall".

balance_e	Drops groups that do not have at least balance_e periods of post-treatment data. This keeps the composition of groups constant across different event times in an event study. Default is NULL, in which case this is ignored.
min_e	The minimum event time computed in the event study results. This is useful when there are a huge number of pre-treatment periods.
max_e	The maximum event time computed in the event study results. This is useful when there are a huge number of post-treatment periods.
...	extra arguments

Value

an aggte_obj

pte_attgt	<i>General ATT(g,t)</i>
-----------	-------------------------

Description

pte_attgt takes a "local" data.frame and computes an estimate of a group time average treatment effect and a corresponding influence function. This function generalizes a number of existing methods and underlies the pte_default function.

The code relies on gt_data having certain variables defined. In particular, there should be an id column (individual identifier), G (group identifier), period (time period), name (equal to "pre" for pre-treatment periods and equal to "post" for post treatment periods), Y (outcome).

In our case, we call two_by_two_subset which sets up the data to have this format before the call to pte_attgt

Usage

```
pte_attgt(
  gt_data,
  xformula,
  d_outcome = FALSE,
  d_covs_formula = ~-1,
  lagged_outcome_cov = FALSE,
  est_method = "dr",
  ...
)
```

Arguments

gt_data	data that is "local" to a particular group-time average treatment effect
xformula	one-sided formula for covariates used in the propensity score and outcome regression models
d_outcome	Whether or not to take the first difference of the outcome. The default is FALSE. To use difference-in-differences, set this to be TRUE.

d_covs_formula	A formula for time varying covariates to enter the first estimation step models. The default is not to include any, and, hence, to only include pre-treatment covariates.
lagged_outcome_cov	Whether to include the lagged outcome as a covariate. Default is FALSE.
est_method	Which type of estimation method to use. Default is "dr" for doubly robust. The other option is "reg" for regression adjustment.
...	extra function arguments; not used here

Value

attgt_if

pte_default	<i>Default, General Function for Computing Treatment Effects with Panel Data</i>
-------------	--

Description

This is a generic/example wrapper for a call to the pte function.

This function provides access to difference-in-differences and unconfoundedness based identification/estimation strategies given (i) panel data and (ii) staggered treatment adoption

Usage

```
pte_default(
  yname,
  gname,
  tname,
  idname = NULL,
  data,
  panel = TRUE,
  xformula = ~1,
  d_outcome = FALSE,
  d_covs_formula = ~-1,
  lagged_outcome_cov = FALSE,
  est_method = "dr",
  anticipation = 0,
  base_period = "varying",
  control_group = "notyettreated",
  weightsname = NULL,
  cband = TRUE,
  alp = 0.05,
  boot_type = "multiplier",
  biters = 100,
  cl = 1,
  ...
)
```

Arguments

yname	Name of outcome in data
gname	Name of group in data
tname	Name of time period in data
idname	Name of id in data
data	balanced panel or repeated cross sections data
panel	Whether the data are panel data. The default is TRUE. Set to FALSE for repeated cross sections.
xformula	one-sided formula for covariates used in the propensity score and outcome regression models
d_outcome	Whether or not to take the first difference of the outcome. The default is FALSE. To use difference-in-differences, set this to be TRUE.
d_covs_formula	A formula for time varying covariates to enter the first estimation step models. The default is not to include any, and, hence, to only include pre-treatment covariates.
lagged_outcome_cov	Whether to include the lagged outcome as a covariate. Default is FALSE.
est_method	Which type of estimation method to use. Default is "dr" for doubly robust. The other option is "reg" for regression adjustment.
anticipation	how many periods before the treatment actually takes place that it can have an effect on outcomes
base_period	The type of base period to use. This only affects the numeric value of results in pre-treatment periods. Results in post-treatment periods are not affected by this choice. The default is "varying", where the base period will "back up" to the immediately preceding period in pre-treatment periods. The other option is "universal" where the base period is fixed in pre-treatment periods to be the period right before the treatment starts. "Universal" is commonly used in difference-in-differences applications, but can be unnatural for other identification strategies.
control_group	Which group is used as the comparison group. The default choice is "notyet-treated", but different estimation strategies can implement their own choices for the control group
weightsname	The name of the column that contains sampling weights. The default is NULL, in which case no sampling weights are used.
cband	whether or not to report a uniform (instead of pointwise) confidence band (default is TRUE)
alp	significance level; default is 0.05
boot_type	should be one of "multiplier" (the default) or "empirical". The multiplier bootstrap is generally much faster, but at tgt_fun needs to provide an expression for the influence function (which could be challenging to figure out). If no influence function is provided, then the pte package will use the empirical bootstrap no matter what the value of this parameter.
biters	number of bootstrap iterations; default is 100
cl	number of clusters to be used when bootstrapping; default is 1
...	additional arguments passed to pte, such as min_e, max_e, and balance_e for controlling the event study range.

Value

pte_results object

Examples

```
# example using minimum wage data
# and a lagged outcome unconfoundedness strategy
library(did)
data(mpdta)
lou_res <- pte_default(
  yname = "lemp",
  gname = "first.treat",
  tname = "year",
  idname = "countyreal",
  data = mpdta,
  xformula = ~lpop,
  d_outcome = FALSE,
  d_covs_formula = ~lpop,
  lagged_outcome_cov = TRUE
)

summary(lou_res)
ggplot2::autoplot(lou_res)
```

pte_dose_results	<i>Class for Continuous Treatment Results</i>
------------------	---

Description

Class for holding results with a continuous treatment

Usage

```
pte_dose_results(att_gt, dose, att_d = NULL, acrt_d = NULL, ptep)
```

Arguments

att_gt	attgt results
dose	vector of doses
att_d	ATT(d) for each value of dose
acrt_d	ACRT(d) for each value of dose
ptep	a pte_params object

Value

a pte_dose_results object

pte_emp_boot

*Class for Empirical Bootstrap Results***Description**

Class for holding ptetools empirical bootstrap results

Usage

```
pte_emp_boot(
  attgt_results,
  overall_results,
  group_results,
  dyn_results,
  overall_weights = NULL,
  dyn_weights = NULL,
  group_weights = NULL,
  extra_gt_returns = NULL,
  ptep = NULL
)
```

Arguments

attgt_results	data.frame holding attgt results
overall_results	data.frame holding overall results
group_results	data.frame holding group results
dyn_results	data.frame holding dynamic results
overall_weights	vector containing weights on underlying ATT(g,t) for overall treatment effect parameter
dyn_weights	list containing weights on underlying ATT(g,t) for each value of e corresponding to the dynamic treatment effect parameters.
group_weights	list containing weights on underlying ATT(g,t) corresponding to deliver averaged group-specific treatment effects
extra_gt_returns	A place to return anything extra from particular group-time average treatment effect calculations. For DID, this might be something like propensity score estimates, regressions of untreated potential outcomes on covariates. For ife, this could be something like the first step regression 2sls estimates. This argument is also potentially useful for debugging.
ptep	pte_params object, stored for reference in the result.

Value

a pte_emp_boot object

pte_params

*PTE Parameters Class***Description**

Class that contains pte parameters

Usage

```
pte_params(
  yname,
  gname,
  tname,
  idname = NULL,
  data,
  panel = TRUE,
  glist,
  tlist,
  cband,
  alp,
  boot_type,
  anticipation = NULL,
  base_period = NULL,
  weightsname = NULL,
  control_group = "notyettreated",
  gt_type = "att",
  ret_quantile = 0.5,
  probs = NULL,
  global_fun = FALSE,
  time_period_fun = FALSE,
  group_fun = FALSE,
  biters,
  cl,
  call = NULL
)
```

Arguments

yname	Name of outcome in data
gname	Name of group in data
tname	Name of time period in data
idname	Name of id in data
data	balanced panel or repeated cross sections data
panel	Whether the data are panel data. The default is TRUE. Set to FALSE for repeated cross sections.

glist	list of groups to create group-time average treatment effects for
tlist	list of time periods to create group-time average treatment effects for
cband	whether or not to report a uniform (instead of pointwise) confidence band (default is TRUE)
alp	significance level; default is 0.05
boot_type	which type of bootstrap to use
anticipation	how many periods before the treatment actually takes place that it can have an effect on outcomes
base_period	The type of base period to use. This only affects the numeric value of results in pre-treatment periods. Results in post-treatment periods are not affected by this choice. The default is "varying", where the base period will "back up" to the immediately preceding period in pre-treatment periods. The other option is "universal" where the base period is fixed in pre-treatment periods to be the period right before the treatment starts. "Universal" is commonly used in difference-in-differences applications, but can be unnatural for other identification strategies.
weightsname	The name of the column that contains sampling weights. The default is NULL, in which case no sampling weights are used.
control_group	Which group is used as the comparison group. The default choice is "notyet-treated", but different estimation strategies can implement their own choices for the control group
gt_type	which type of group-time effects are computed. The default is "att". Different estimation strategies can implement their own choices for gt_type
ret_quantile	For functions that compute quantile treatment effects, this is a specific quantile at which to report results, e.g., <code>ret_quantile = 0.5</code> will return that the qte at the median.
probs	For <code>gt_type = "qtt"</code> , a numeric vector of quantile levels at which to evaluate the QTT curve (e.g., <code>seq(0.05, 0.95, 0.05)</code>). Defaults to <code>seq(0.05, 0.95, 0.05)</code> when NULL.
global_fun	Logical indicating whether or not untreated potential outcomes can be estimated in one shot, i.e., for all groups and time periods. Main use case would be for one-shot imputation estimators. Not supported yet.
time_period_fun	Logical indicating whether or not untreated potential outcomes can be estimated for all groups in the same time period. Not supported yet.
group_fun	Logical indicating whether or not untreated potential outcomes can be estimated for all time periods for a single group. Not supported yet. These functions aim at reducing or eliminating running the same code multiple times.
biters	number of bootstrap iterations; default is 100
cl	number of clusters to be used when bootstrapping; default is 1
call	keeps track of through the call from external functions/packages

Value

pte_params object

pte_qtt	<i>Class for QTT Curve Results</i>
---------	------------------------------------

Description

Holds the full quantile treatment effect (QTT) curve at the overall, group-specific, and dynamic (event-study) aggregation levels. Each aggregation contains estimates at all quantile levels in probs together with bootstrap standard errors and pointwise confidence intervals.

Usage

```
pte_qtt(overall, dynamic, group, F0_overall = NULL, F1_overall = NULL, ptep)
```

Arguments

overall	data.frame with columns probs, qtt, se, lower_pw, upper_pw, lower_ub, upper_ub
dynamic	data.frame with columns e, probs, qtt, se, lower_pw, upper_pw, lower_ub, upper_ub
group	data.frame with columns group, probs, qtt, se, lower_pw, upper_pw, lower_ub, upper_ub
F0_overall	mixture CDF of untreated potential outcomes
F1_overall	mixture CDF of treated potential outcomes
ptep	pte_params object

Value

a pte_qtt object

pte_results	<i>Class for PTE Results</i>
-------------	------------------------------

Description

Class for holding overall results with a staggered treatment, including an overall ATT and an event study

Usage

```
pte_results(att_gt, overall_att, event_study, ptep)
```

Arguments

att_gt	attgt results
overall_att	overall_att results
event_study	event_study results
ptep	pte_params object

Value

a pte_results object

qott_pte_aggregations *Aggregate Group-Time Quantile of the Treatment Effect*

Description

Aggregate group-time distribution of the treatment effect into overall, group, and dynamic effects.

Usage

```
qott_pte_aggregations(attgt.list, ptep, extra_gt_returns)
```

Arguments

attgt.list list of attgt results from compute_pte

ptep pte_params object

extra_gt_returns

A place to return anything extra from particular group-time average treatment effect calculations. For DID, this might be something like propensity score estimates, regressions of untreated potential outcomes on covariates. For ife, this could be something like the first step regression 2sls estimates. This argument is also potentially useful for debugging.

Value

pte_emp_boot object

qtt_empirical_bootstrap
 Empirical Bootstrap for QTT Curves

Description

Runs the empirical bootstrap for the full QTT curve case (gt_type = "qtt"). Called automatically by panel_empirical_bootstrap when gt_type == "qtt".

Usage

```

qtt_empirical_bootstrap(
  attgt.list,
  ptep,
  setup_pte_fun,
  subset_fun,
  attgt_fun,
  extra_gt_returns,
  aggte_fun = NULL,
  ...
)

```

Arguments

attgt.list	list of attgt results from compute_pte
ptep	pte_params object
setup_pte_fun	This is a function that should take in data, yname (the name of the outcome variable in data), gname (the name of the group variable), idname (the name of the id variable), and possibly other arguments such as the significance level alp, the number of bootstrap iterations biters, and how many clusters for parallel computing in the bootstrap cl. The key thing that needs to be figured out in this function is which groups and time periods ATT(g,t) should be computed in. The function should return a pte_params object which contains all of the parameters passed into the function as well as glist and tlist which should be ordered lists of groups and time periods for ATT(g,t) to be computed. This function also provides a good place for error handling related to the types of data that can be handled. The pte package contains the function setup_pte that is a lightweight function that basically just takes the data, omits the never-treated group from glist but includes all other groups and drops the first time period. This works in cases where ATT would be identified in the 2x2 case (i.e., where there are two time periods, no units are treated in the first period and the identification strategy "works" with access to a treated and untreated group and untreated potential outcomes for both groups in the first period) — for example, this approach works if DID is the identification strategy.
subset_fun	This is a function that should take in data, g (for group), tp (for time period), and ... and be able to return the appropriate data.frame that can be used by attgt_fun to produce ATT(g=g, tp=tp). The data frame should be constructed using gt_data_frame in order to guarantee that it has the appropriate columns that identify which group an observation belongs to, etc.
attgt_fun	This is a function that should work in the case where there is a single group and the "right" number of time periods to recover an estimate of the ATT. For example, in the context of difference in differences, it would need to work for a single group, find the appropriate comparison group (untreated units), find the right time periods (pre- and post-treatment), and then recover an estimate of ATT for that group. It will be called over and over separately by groups and by time periods to compute ATT(g,t)'s.

The function needs to work in a very specific way. It should take in the arguments: `data`, ... `data` should be constructed using the function `gt_data_frame` which checks to make sure that `data` has the correct columns defined. ... are additional arguments (such as formulas for covariates) that `attgt_fun` needs. From these arguments `attgt_fun` must return a list with element `ATT` containing the group-time average treatment effect for that group and that time period.

If `attgt_fun` returns an influence function (which should be provided in a list element named `inf_func`), then the code will use the multiplier bootstrap to compute standard errors for group-time average treatment effects, an overall treatment effect parameter, and a dynamic treatment effect parameter (i.e., event study parameter). If `attgt_fun` does not return an influence function, then the same objects will be computed using the empirical bootstrap. This is usually (perhaps substantially) easier to code, but also will usually be (perhaps substantially) computationally slower.

`extra_gt_returns`

A place to return anything extra from particular group-time average treatment effect calculations. For DID, this might be something like propensity score estimates, regressions of untreated potential outcomes on covariates. For ife, this could be something like the first step regression 2sls estimates. This argument is also potentially useful for debugging.

`aggte_fun`

An optional function for aggregating group-time treatment effects. When `NULL` (the default), the function is selected automatically based on `gt_type`.

...

extra arguments that can be passed to create the correct subsets of the data (depending on `subset_fun`), to estimate group time average treatment effects (depending on `attgt_fun`), or to aggregating treatment effects (particularly useful are `min_e`, `max_e`, and `balance_e` arguments to event study aggregations)

Value

`pte_qtt` object

`qtt_pte_aggregations` *Aggregate Group-Time Quantile Treatment Effects*

Description

Aggregate group-time F0/F1 distributions into QTT curves at the overall, group, and dynamic level. CDFs are mixed first using `BMisc::combine_ecdfs` and then inverted at all quantile levels in `probs`, avoiding the bias from averaging scalar QTTs.

Usage

```
qtt_pte_aggregations(attgt.list, ptep, extra_gt_returns)
```

Arguments

attgt.list	list of attgt results from compute_pte
ptep	pte_params object
extra_gt_returns	A place to return anything extra from particular group-time average treatment effect calculations. For DID, this might be something like propensity score estimates, regressions of untreated potential outcomes on covariates. For ife, this could be something like the first step regression 2sls estimates. This argument is also potentially useful for debugging.

Value

named list with elements overall_results, dyn_results, group_results, F0_overall, F1_overall

setup_pte	<i>Generic Setup Function</i>
-----------	-------------------------------

Description

This is a function for how to setup the data to be used in the ptetools package.

The setup_pte function builds on setup_pte_basic and attempts to provide a general purpose function (with error handling) to arrange the data in a way that can be processed by subset_fun and attgt_fun in the next steps.

Usage

```
setup_pte(
  yname,
  gname,
  tname,
  idname = NULL,
  data,
  panel = TRUE,
  required_pre_periods = 1,
  anticipation = 0,
  base_period = "varying",
  cband = TRUE,
  alp = 0.05,
  boot_type = "multiplier",
  weightsname = NULL,
  gt_type = "att",
  ret_quantile = 0.5,
  probs = NULL,
  biters = 100,
  cl = 1,
  call = NULL,
```



```
    ...
)
```

Arguments

yname	Name of outcome in data
gname	Name of group in data
tname	Name of time period in data
idname	Name of id in data
data	balanced panel or repeated cross sections data
panel	Whether the data are panel data. The default is TRUE. Set to FALSE for repeated cross sections.
required_pre_periods	The number of required pre-treatment periods to implement the estimation strategy. Default is 1.
anticipation	how many periods before the treatment actually takes place that it can have an effect on outcomes
base_period	The type of base period to use. This only affects the numeric value of results in pre-treatment periods. Results in post-treatment periods are not affected by this choice. The default is "varying", where the base period will "back up" to the immediately preceding period in pre-treatment periods. The other option is "universal" where the base period is fixed in pre-treatment periods to be the period right before the treatment starts. "Universal" is commonly used in difference-in-differences applications, but can be unnatural for other identification strategies.
cband	whether or not to report a uniform (instead of pointwise) confidence band (default is TRUE)
alp	significance level; default is 0.05
boot_type	which type of bootstrap to use
weightsname	The name of the column that contains sampling weights. The default is NULL, in which case no sampling weights are used.
gt_type	which type of group-time effects are computed. The default is "att". Different estimation strategies can implement their own choices for gt_type
ret_quantile	For functions that compute quantile treatment effects, this is a specific quantile at which to report results, e.g., <code>ret_quantile = 0.5</code> will return that the qte at the median.
probs	For <code>gt_type = "qtt"</code> , a numeric vector of quantile levels at which to evaluate the QTT curve (e.g., <code>seq(0.05, 0.95, 0.05)</code>). Defaults to <code>seq(0.05, 0.95, 0.05)</code> when NULL.
biters	number of bootstrap iterations; default is 100
cl	number of clusters to be used when bootstrapping; default is 1
call	keeps track of through the call from external functions/packages
...	additional arguments

Value

pte_params object

setup_pte_basic

Basic Setup Function

Description

This is a lightweight (example) function for how to setup the data to be used in the ptetools package.

setup_pte_basic takes in information about the structure of data and returns a pte_params object. The key piece of information that is computed by this function is the list of groups and list of time periods where ATT(g,t) should be computed. In particular, this function omits the never-treated group but includes all other groups and drops the first time period. This setup is basically geared towards the 2x2 case — i.e., where ATT could be identified with two periods, a treated and untreated group, and the first period being pre-treatment for both groups. This is the relevant case for DID, but is also relevant for other cases as well. However, for example, if more pre-treatment periods were needed, then this function should be replaced by something else.

For code that is written with the idea of being easy-to-use by other researchers, this is a good place to do some error handling / checking that the data is in the correct format, etc.

Usage

```
setup_pte_basic(
  yname,
  gname,
  tname,
  idname = NULL,
  data,
  panel = TRUE,
  cband = TRUE,
  alp = 0.05,
  boot_type = "multiplier",
  gt_type = "att",
  ret_quantile = 0.5,
  probs = NULL,
  biters = 100,
  cl = 1,
  call = NULL,
  ...
)
```

Arguments

yname	Name of outcome in data
gname	Name of group in data

tname	Name of time period in data
idname	Name of id in data
data	balanced panel or repeated cross sections data
panel	Whether the data are panel data. The default is TRUE. Set to FALSE for repeated cross sections.
cband	whether or not to report a uniform (instead of pointwise) confidence band (default is TRUE)
alp	significance level; default is 0.05
boot_type	which type of bootstrap to use
gt_type	which type of group-time effects are computed. The default is "att". Different estimation strategies can implement their own choices for gt_type
ret_quantile	For functions that compute quantile treatment effects, this is a specific quantile at which to report results, e.g., ret_quantile = 0.5 will return that the qte at the median.
probs	For gt_type = "qtt", a numeric vector of quantile levels at which to evaluate the QTT curve (e.g., seq(0.05, 0.95, 0.05)). Defaults to seq(0.05, 0.95, 0.05) when NULL.
biters	number of bootstrap iterations; default is 100
cl	number of clusters to be used when bootstrapping; default is 1
call	keeps track of through the call from external functions/packages
...	additional arguments

Value

pte_params object

two_by_two_rcs_subset *Two Period Two Group Repeated Cross Sections Subset*

Description

A function for computing a 2x2 subset for repeated cross sections data. This is analogous to two_by_two_subset, but indexes observations by rows rather than by panel ids.

Usage

```
two_by_two_rcs_subset(
  data,
  g,
  tp,
  control_group = "notyettreated",
  anticipation = 0,
  base_period = "varying",
  ...
)
```

Arguments

<code>data</code>	the full dataset
<code>g</code>	the current group
<code>tp</code>	the current time period
<code>control_group</code>	whether to use "notyettreated" (default) or "nevertreated"
<code>anticipation</code>	the number of periods of anticipation (i.e., number of periods before the treatment happens where the treatment can "already" affect the outcome)
<code>base_period</code>	The type of base period to use. This only affects the numeric value of results in pre-treatment periods. Results in post-treatment periods are not affected by this choice. The default is "varying", where the base period will "back up" to the immediately preceding period in pre-treatment periods. The other option is "universal" where the base period is fixed in pre-treatment periods to be the period right before the treatment starts. "Universal" is commonly used in difference-in-differences applications, but can be unnatural for other identification strategies.
<code>...</code>	extra arguments to get the subset correct

Value

list that contains the following elements:

- `gt_data`: a `gt_data_frame` object that contains the correct subset of data
- `n1`: the number of observations in this subset
- `disidx`: a vector of the correct rows for this subset

<code>two_by_two_subset</code>	<i>Two Period Two Group Subset</i>
--------------------------------	------------------------------------

Description

A function for computing a 2x2 subset of original data. This is the subset with post treatment periods separately for the treated group and comparison group and pre-treatment periods in the period immediately before the treated group became treated.

Usage

```
two_by_two_subset(
  data,
  g,
  tp,
  control_group = "notyettreated",
  anticipation = 0,
  base_period = "varying",
  ...
)
```

Arguments

data	the full dataset
g	the current group
tp	the current time period
control_group	whether to use "notyettreated" (default) or "nevertreated"
anticipation	the number of periods of anticipation (i.e., number of periods before the treatment happens where the treatment can "already" affect the outcome)
base_period	The type of base period to use. This only affects the numeric value of results in pre-treatment periods. Results in post-treatment periods are not affected by this choice. The default is "varying", where the base period will "back up" to the immediately preceding period in pre-treatment periods. The other option is "universal" where the base period is fixed in pre-treatment periods to be the period right before the treatment starts. "Universal" is commonly used in difference-in-differences applications, but can be unnatural for other identification strategies.
...	extra arguments to get the subset correct

Value

list that contains the following elements:

- gt_data: a gt_data_frame object that contains the correct subset of data
- n1: the number of observations in this subset
- disidx: a vector of the correct ids for this subset

Index

- * **datasets**
 - covid_data, [10](#)
- aggte_obj, [3](#)
- attgt_if, [4](#)
- attgt_noif, [5](#)
- attgt_pte_aggregations, [5](#)
- autoplot.dose_obj, [6](#), [21](#)
- autoplot.ptemp_boot, [6](#), [22](#)
- autoplot.pteqtt, [7](#), [22](#)
- autoplot.pteresults, [8](#), [23](#)
- compute.pteq, [8](#)
- covid_attgt, [9](#)
- covid_data, [10](#)
- crit_val_checks, [11](#)
- did_attgt, [12](#)
- did_rcs_attgt, [12](#)
- dose_obj, [13](#)
- ggpte, [14](#)
- ggpte_cont, [15](#)
- group_time_att, [15](#)
- gt_data_frame, [16](#)
- keep_all_pretreatment_subset, [17](#)
- keep_all_untreated_subset, [17](#)
- mboot2, [18](#)
- overall_weights, [19](#)
- panel_empirical_bootstrap, [19](#)
- plot.dose_obj, [21](#)
- plot.ptemp_boot, [22](#)
- plot.pteqtt, [22](#)
- plot.pteresults, [23](#)
- process_att_gt, [23](#)
- process_dose_gt, [24](#)
- pte, [24](#)
- pte_aggte, [28](#)
- pte_attgt, [29](#)
- pte_default, [30](#)
- pte_dose_results, [32](#)
- pte_emp_boot, [33](#)
- pte_params, [34](#)
- pteqtt, [36](#)
- pteresults, [36](#)
- ptetools (pte), [24](#)
- ptetools-package (pte), [24](#)
- qott_pte_aggregations, [37](#)
- qtt_empirical_bootstrap, [37](#)
- qtt_pte_aggregations, [39](#)
- setup_pteq, [40](#)
- setup_pteq_basic, [42](#)
- two_by_two_rcs_subset, [43](#)
- two_by_two_subset, [44](#)