Package 'ptetools'

July 23, 2025

Title Panel Treatment Effects Tools

Version 1.0.0

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.

License GPL-3

Encoding UTF-8

RoxygenNote 7.3.2

Imports BMisc (>= 1.4.7), Matrix, ggplot2, DRDID, tidyr, dplyr, pbapply, splines2

Suggests testthat (>= 3.0.0), did

Config/testthat/edition 3

NeedsCompilation no

Author Brantly Callaway [aut, cre]

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

Repository CRAN

Date/Publication 2025-02-13 10:00:01 UTC

Contents

aggte_obj 2
attgt_if 3
attgt_noif 4
attgt_pte_aggregations 5
compute.pte 5

crit_val_checks . . . 6
did_attgt . . . 7
dose_obj . . . 8
ggpte . . . 9
ggpte_cont . . . 9
group_time_att . . . 10
gt_data_frame . . . 11
keep_all_pretreatment_subset . . . 12
keep_all_untreated_subset . . . 12
mboot2 . . . 13
overall_weights . . . 14
panel_empirical_bootstrap . . . 14
process_att_gt . . . 16
process_dose_gt . . . 17
pte . . . 17
pte_aggte . . . 21
pte_attgt . . . 22
pte_default . . . 23
pte_dose_results . . . 25
pte_emp_boot . . . 25
pte_params . . . 26
pte_results . . . 28
qott_pte_aggregations . . . 29
qtt_pte_aggregations . . . 30
setup_pte . . . 30
setup_pte_basic . . . 32
two_by_two_subset . . . 33

Index **35**

`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

overall.att	The estimated overall ATT
overall.se	Standard error for overall ATT
type	The type of aggregation to be done. Default is "overall".
egt	Holds the length of exposure (for dynamic effects), the group (for selective treatment timing), or the time period (for calendar time effects)
att.egt	The ATT specific to egt
se.egt	The standard error specific to egt
crit.val.egt	A critical value for computing uniform confidence bands for dynamic effects, selective treatment timing, or time period effects.
inf.function	The influence function of the chosen aggregated parameters
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.
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 *Class for (g,t)-Specific Results with Influence Function*

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

compute.pte *Heavy-Lifting for pte Function*

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(ptep, subset_fun, attgt_fun, ...)
```

Arguments

ptep	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 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.

`...` 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

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 calculationsons

---

`crit_val_checks` *Sanity Checks on Critical Values*

Description

A function to perform sanity checks and possibly adjust a 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 *Difference-in-differences for ATT(g,t)*

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

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
...	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
overall_att_inffunc	the influence function for estimating overall_att
overall_acrt	estimate of the overall ACRT, the mean of ACRT(D|D) given D > 0
overall_acrt_se	the standard error for the estimate of overall_acrt
overall_acrt_inffunc	the influence function for estimating overall_acrt
att.d	estimates of ATT(d) for each value of dose
att.d_se	standard error of ATT(d) for each value of dose
att.d_crit.val	critical value to produce pointwise or uniform confidence interval for ATT(d)
att.d_inffunc	matrix containing the influence function from estimating ATT(d)

acrt.d	estimates of ACRT(d) for each value of dose
acrt.d_se	standard error of ACRT(d) for each value of dose
acrt.d_crit.val	critical value to produce pointwise or uniform confidence interval for ACRT(d)
acrt.d_inffunc	matrix containing the influence function from estimating ACRT(d)
pte_params	a pte_params object containing other parameters passed to the function

Value

a dose_obj object

---

ggpte *ptetools Generic Plotting Function*

Description

The main plotting function in the ptetools package. It plots event studies. This function is generic enough that most packages that otherwise use the ptetools package can call it directly to plot an event study.

Usage

```
ggpte(pte_results)
```

Arguments

pte_results	A pte_results object

Value

A ggplot object

---

ggpte_cont *Generic Plots with a Continuous Treatment*

Description

Plots dose-specific results in applications with a continuous treatment

Usage

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

Arguments

dose_obj	a dose_obj that holds results with a continuous treatment
type	whether to plot ATT(d) or ACRT(d), defaults to att for plotting ATT(d). For ACRT(d), use "acrt"

Value

A ggplot object

---

group_time_att *Class for Estimates across Groups and Time*

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 *Convert Data to Usable Format*

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

data	the full dataset
g	the current group
tp	the current time period
...	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 *Weights for Overall Aggregation*

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 *Panel Empirical Bootstrap*

Description

Computes empirical bootstrap pointwise standard errors

Usage

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

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.

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.

...

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_emp_boot object

---

process_att_gt *Process ATT(g,t) Results*

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 *Process Results with a Continuous Treatment*

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

gt_results	list of group-time specific results
ptep	pte_params object
...	extra arguments

Value

a dose_obj object

pte *Panel Treatment Effects*

Description

Tools for estimating treatment effects with panel data.

Main function for computing panel treatment effects

Usage

```
pte(
  yname,
  gname,
  tname,
  idname,
  data,
  setup_pte_fun,
  subset_fun,
  attgt_fun,
  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,
  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 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 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.

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 `attgt_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.

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., `ret_quantile = 0.5` will return that the qte at the median.

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.

process_dtt_gt_fun
: An optional function to customize results when the gt-specific function returns the distribution of treated and untreated potential outcomes. The default is process_dtt_gt, which is a function provided by the package. See that function for an example of what this function should return. This is unused is unused except in cases where the results involve distributions.

process_dose_gt_fun
: An optional function to customize results when the gt-specific function returns treatment effects that depend on dose (i.e., amount of the treatment). The default is process_dose_gt, 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.

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

...
: 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_results object

Author(s)

Maintainer: 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,
  xformla = ~lpop
)

summary(did_res)
ggpte(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 *General ATT(g,t)*

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 *Default, General Function for Computing Treatment Effects with Panel Data*

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,
  data,
  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 data
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 `attgt_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

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)
ggpte(lou_res)
```

`pte_dose_results` *Class for Continuous Treatment Results*

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
)
```

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.

Value

a pte_emp_boot object

---

pte_params *PTE Parameters Class*

Description

Class that contains pte parameters

Usage

```
pte_params(
  yname,
  gname,
  tname,
  idname,
  data,
  glist,
  tlist,
  cband,
  alp,
  boot_type,
  anticipation = NULL,
  base_period = NULL,
  weightsname = NULL,
  control_group = "notyettreated",
  gt_type = "att",
  ret_quantile = 0.5,
  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 data
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., `ret_quantile = 0.5` will return that the qte at the median.
`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_results` *Class for PTE Results*

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_pte_aggregations *Aggregate Group-Time Quantile Treatment Effects*

Description

Aggregate group-time distributions into qtt versions of overall, group, and dynamic effects.

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

pte_emp_boot object

setup_pte *Generic Setup Function*

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,
  data,
  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,
  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 data
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., ret_quantile = 0.5 will return that the qte at the median.
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,
  data,
  cband = TRUE,
  alp = 0.05,
  boot_type = "multiplier",
  gt_type = "att",
  ret_quantile = 0.5,
  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 data
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.
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_subset *Two Period Two Group Subset*

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

aggte_obj, 2
attgt_if, 3
attgt_noif, 4
attgt_pte_aggregations, 5

compute.pte, 5
crit_val_checks, 6

did_attgt, 7
dose_obj, 8

ggpte, 9
ggpte_cont, 9
group_time_att, 10
gt_data_frame, 11

keep_all_pretreatment_subset, 12
keep_all_untreated_subset, 12

mboot2, 13

overall_weights, 14

panel_empirical_bootstrap, 14
process_att_gt, 16
process_dose_gt, 17
pte, 17
pte_aggte, 21
pte_attgt, 22
pte_default, 23
pte_dose_results, 25
pte_emp_boot, 25
pte_params, 26
pte_results, 28
ptetools *(pte)*, 17
ptetools-package *(pte)*, 17

qott_pte_aggregations, 29
qtt_pte_aggregations, 30

setup_pte, 30
setup_pte_basic, 32

two_by_two_subset, 33