Package ‘did2s’

September 19, 2025

Title Two-Stage Difference-in-Differences Following Gardner (2021)

Version 1.2.0

Description Estimates Two-way Fixed Effects difference-in-differences/event-study models using the approach proposed by Gardner (2021) <doi:10.48550/arXiv.2207.05943>. To avoid the problems caused by OLS estimation of the Two-way Fixed Effects model, this function first estimates the fixed effects and covariates using untreated observations and then in a second stage, estimates the treatment effects.

License MIT + file LICENSE

Encoding UTF-8

LazyData true

RoxygenNote 7.3.2

Depends R (>= 3.5.0), fixest (>= 0.13.2)

Imports boot, broom, data.table, did, didimputation, dreamerr, ggplot2, HonestDiD, Matrix, rlang, staggered, stats

URL https://kylebutts.github.io/did2s/

Suggests haven, knitr, rmarkdown, testthat (>= 3.0.0)

VignetteBuilder knitr

Config/testthat/edition 3

NeedsCompilation no

Author Kyle Butts [aut, cre] (ORCID: <https://orcid.org/0000-0002-9048-8059>),
John Gardner [aut] (ORCID: <https://orcid.org/0000-0002-4028-6862>),
Grant McDermott [ctb] (ORCID: <https://orcid.org/0000-0001-7883-8573>),
Laurent Berge [ctb]

Maintainer Kyle Butts <buttskyle96@gmail.com>

Repository CRAN

Date/Publication 2025-09-19 07:10:10 UTC

Contents

castle 2
df_het 3
df_hom 3
did2s 4
event_study 8
gen_data 9
get_honestdid_obj_did2s 10
honest_did_did2s 11
robust_solve_XtX 13

Index **14**

`castle` *Data from Cheng and Hoekstra (2013)*

Description

State-wide panel data from 2000-2010 that has information on castle-doctrine, the so-called "stand-your-ground" laws that were implemented by 20 states.

Usage

```
castle
```

Format

A data frame with 550 rows and 5 variables:

sid state id, unit of observation

year time in panel data

l_homicide log of the number of homicides per capita

effyear year that castle doctrine is passed

post 0/1 variable for when castle doctrine is active

time_til time relative to castle doctrine being passed into law

df_het — *Simulated data with two treatment groups and heterogenous effects*

Description

Generated using the following call: `did2s::gen_data(panel = c(1990, 2020), g1 = 2000, g2 = 2010, g3 = 0, te1 = 2, te2 = 1, te3 = 0, te_m1 = 0.05, te_m2 = 0.15, te_m3 = 0)`

Usage

```
df_het
```

Format

A data frame with 31000 rows and 15 variables:

unit individual in panel data

year time in panel data

g the year that treatment starts

dep_var outcome variable

treat T/F variable for when treatment is on

rel_year year relative to treatment start. Inf = never treated.

rel_year_binned year relative to treatment start, but <=-6 and >=6 are binned.

unit_fe Unit FE

year_fe Year FE

error Random error component

te Static treatment effect = te

te_dynamic Dynamic treatmet effect = te_m

state State that unit is in

group String name for group

df_hom — *Simulated data with two treatment groups and homogenous effects*

Description

Generated using the following call: `did2s::gen_data(panel = c(1990, 2020), g1 = 2000, g2 = 2010, g3 = 0, te1 = 2, te2 = 2, te3 = 0, te_m1 = 0, te_m2 = 0, te_m3 = 0)`

Usage

```
df_hom
```

Format

A data frame with 31000 rows and 15 variables:

unit individual in panel data

year time in panel data

g the year that treatment starts

dep_var outcome variable

treat T/F variable for when treatment is on

rel_year year relative to treatment start. Inf = never treated.

rel_year_binned year relative to treatment start, but <=-6 and >=6 are binned.

unit_fe Unit FE

year_fe Year FE

error Random error component

te Static treatment effect = te

te_dynamic Dynamic treatmet effect = te_m

group String name for group

state State that unit is in

weight Weight from runif()

`did2s` *Calculate two-stage difference-in-differences following Gardner (2021)*

Description

Calculate two-stage difference-in-differences following Gardner (2021)

Usage

```
did2s(
  data,
  yname,
  first_stage,
  second_stage,
  treatment,
  cluster_var,
  weights = NULL,
  bootstrap = FALSE,
  n_bootstraps = 250,
  return_bootstrap = FALSE,
  verbose = FALSE
)
```

Arguments

- `data` — The dataframe containing all the variables
- `yname` — Outcome variable
- `first_stage` — Fixed effects and other covariates you want to residualize with in first stage. Formula following `fixest::feols`. Fixed effects specified after "`|`".
- `second_stage` — Second stage, these should be the treatment indicator(s) (e.g. treatment variable or event-study leads/lags). Formula following `fixest::feols`. Use `i()` for factor variables, see `fixest::i`.
- `treatment` — A variable that = 1 if treated, = 0 otherwise. The first stage will be estimated for `treatment == 0`. The second stage will be estimated for the *full sample*.
- `cluster_var` — What variable to cluster standard errors. This can be IDs or a higher aggregate level (state for example)
- `weights` — Optional. Variable name for regression weights.
- `bootstrap` — Optional. Should standard errors be calculated using bootstrap? Default is `FALSE`.
- `n_bootstraps` — Optional. How many bootstraps to run. Default is `250`.
- `return_bootstrap` — Optional. Logical. Will return each bootstrap second-stage estimate to allow for manual use, e.g. percentile standard errors and empirical confidence intervals.
- `verbose` — Optional. Logical. Should information about the two-stage procedure be printed back to the user? Default is `TRUE`.

Value

`fixest` object with adjusted standard errors (either by formula or by bootstrap). All the methods from `fixest` package will work, including `fixest::esttable` and `fixest::coefplot`

Examples

Load example dataset which has two treatment groups and homogeneous treatment effects

```
# Load Example Dataset
data("df_hom")
```

Static TWFE:

You can run a static TWFE fixed effect model for a simple treatment indicator

```
static <- did2s(df_hom,
    yname = "dep_var", treatment = "treat", cluster_var = "state",
    first_stage = ~ 0 | unit + year,
    second_stage = ~ i(treat, ref=FALSE))

fixest::esttable(static)
#>                            static
#> Dependent Var.:           dep_var
#>
```

```
#> treat = TRUE    2.005*** (0.0202)
#> _______________ _________________
#> S.E.: Clustered         by: state
#> Observations               46,500
#> R2                        0.47520
#> Adj. R2                   0.47520
#> ---
#> Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Event Study:

Or you can use relative-treatment indicators to estimate an event study estimate

```
es <- did2s(df_hom,
    yname = "dep_var", treatment = "treat", cluster_var = "state",
    first_stage = ~ 0 | unit + year,
    second_stage = ~ i(rel_year, ref=c(-1, Inf)))

fixest::esttable(es)
#>                                  es
#> Dependent Var.:             dep_var
#>
#> rel_year = -20    0.0043 (0.0322)
#> rel_year = -19    0.0222 (0.0296)
#> rel_year = -18   -0.0358 (0.0308)
#> rel_year = -17    0.0043 (0.0337)
#> rel_year = -16   -0.0186 (0.0353)
#> rel_year = -15   -0.0045 (0.0346)
#> rel_year = -14   -0.0393 (0.0384)
#> rel_year = -13    0.0453 (0.0323)
#> rel_year = -12    0.0324 (0.0309)
#> rel_year = -11   -0.0245 (0.0349)
#> rel_year = -10   -0.0017 (0.0241)
#> rel_year = -9     0.0155 (0.0242)
#> rel_year = -8    -0.0073 (0.0210)
#> rel_year = -7   -0.0513* (0.0202)
#> rel_year = -6     0.0269 (0.0237)
#> rel_year = -5     0.0136 (0.0237)
#> rel_year = -4    0.0381. (0.0223)
#> rel_year = -3    -0.0228 (0.0284)
#> rel_year = -2     0.0041 (0.0228)
#> rel_year = 0    1.971*** (0.0470)
#> rel_year = 1    2.050*** (0.0466)
#> rel_year = 2    2.033*** (0.0441)
#> rel_year = 3    1.966*** (0.0400)
#> rel_year = 4    1.965*** (0.0430)
#> rel_year = 5    2.030*** (0.0456)
#> rel_year = 6    2.040*** (0.0447)
#> rel_year = 7    1.995*** (0.0370)
#> rel_year = 8    2.019*** (0.0485)
```

```
#> rel_year = 9    1.955*** (0.0468)
#> rel_year = 10   1.950*** (0.0455)
#> rel_year = 11   2.117*** (0.0664)
#> rel_year = 12   2.132*** (0.0741)
#> rel_year = 13   2.019*** (0.0640)
#> rel_year = 14   2.013*** (0.0522)
#> rel_year = 15   1.961*** (0.0605)
#> rel_year = 16   1.916*** (0.0584)
#> rel_year = 17   1.938*** (0.0607)
#> rel_year = 18   2.070*** (0.0666)
#> rel_year = 19   2.066*** (0.0609)
#> rel_year = 20   1.964*** (0.0612)
#> _______________ _________________
#> S.E.: Clustered         by: state
#> Observations               46,500
#> R2                        0.47577
#> Adj. R2                   0.47533
#> ---
#> Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

# plot rel_year coefficients and standard errors
fixest::coefplot(es, keep = "rel_year::(.*)")
```

Example from Cheng and Hoekstra (2013):

Here's an example using data from Cheng and Hoekstra (2013)

```
# Castle Data
castle <- haven::read_dta("https://github.com/scunning1975/mixtape/raw/master/castle.dta")

did2s(
data = castle,
yname = "l_homicide",
first_stage = ~ 0 | sid + year,
second_stage = ~ i(post, ref=0),
treatment = "post",
cluster_var = "state", weights = "popwt"
)
#> OLS estimation, Dep. Var.: l_homicide
#> Observations: 550
#> Weights: weights_vector
#> Standard-errors: Corrected Clustered (state)
#>        Estimate Std. Error t value Pr(>|t|)
#> post::1 0.075142    0.03538 2.12387 0.034127 *
#> ---
#> Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
#> RMSE: 0.109374   Adj. R2: 0.052465
```

event_study — *Estimate event-study coefficients using TWFE and 5 proposed improvements.*

Description

Uses the estimation procedures recommended from Borusyak, Jaravel, Spiess (2021); Callaway and Sant'Anna (2020); Gardner (2021); Roth and Sant'Anna (2021); Sun and Abraham (2020)

Usage

```
event_study(
  data,
  yname,
  idname,
  gname,
  tname,
  xformla = NULL,
  weights = NULL,
  estimator = c("all", "TWFE", "did2s", "did", "impute", "sunab", "staggered"),
  verbose = TRUE
)

plot_event_study(out, separate = TRUE, horizon = NULL)
```

Arguments

data	The dataframe containing all the variables
yname	Variable name for outcome variable
idname	Variable name for unique unit id
gname	Variable name for unit-specific date of initial treatment (never-treated should be zero or NA)
tname	Variable name for calendar period
xformla	A formula for the covariates to include in the model. It should be of the form ~ X1 + X2. Default is NULL.
weights	Variable name for estimation weights. This is used in estimating Y(0) and also augments treatment effect weights. Implementation of Roth and Sant'Anna (2021) currently does not allow for weights.
estimator	Estimator you would like to use. Use "all" to estimate all. Otherwise see table to know advantages and requirements for each of these.
verbose	Optional. Logical. Should information about the two-stage procedure be printed back to the user? Default is TRUE.
out	Output from event_study()
separate	Logical. Should the estimators be on separate plots? Default is TRUE.
horizon	Numeric. Vector of length 2. First element is min and second element is max of event_time to plot

Value

event_study returns a data.frame of point estimates for each estimator

plot_event_study returns a ggplot object that can be fully customized

Examples

```
out = event_study(
  data = did2s::df_het, yname = "dep_var", idname = "unit",
  tname = "year", gname = "g", estimator = "all"
)
plot_event_study(out)
```

---

gen_data *Generate TWFE data*

Description

Generate TWFE data

Usage

```
gen_data(
  g1 = 2000,
  g2 = 2010,
  g3 = 0,
  panel = c(1990, 2020),
  te1 = 2,
  te2 = 2,
  te3 = 2,
  te_m1 = 0,
  te_m2 = 0,
  te_m3 = 0,
  n = 1500
)
```

Arguments

g1	treatment date for group 1. For no treatment, set g = 0.
g2	treatment date for group 2. For no treatment, set g = 0.
g3	treatment date for group 3. For no treatment, set g = 0.
panel	numeric vector of size 2, start and end years for panel
te1	treatment effect for group 1. Will ignore for that group if g = 0.
te2	treatment effect for group 1. Will ignore for that group if g = 0.
te3	treatment effect for group 1. Will ignore for that group if g = 0.

te_m1	treatment effect slope per year
te_m2	treatment effect slope per year
te_m3	treatment effect slope per year
n	number of individuals in sample

Value

Dataframe of generated data

Examples

```
# Homogeneous treatment effect
df_hom <- gen_data(panel = c(1990, 2020),
    g1 = 2000, g2 = 2010, g3 = 0,
    te1 = 2, te2 = 2, te3 = 0,
    te_m1 = 0, te_m2 = 0, te_m3 = 0)
# Heterogeneous treatment effect
df_het <- gen_data(panel = c(1990, 2020),
    g1 = 2000, g2 = 2010, g3 = 0,
    te1 = 2, te2 = 1, te3 = 0,
    te_m1 = 0.05, te_m2 = 0.15, te_m3 = 0)
```

---

get_honestdid_obj_did2s *get_honestdid_obj_did2s*

Description

a helper function that takes a fixest feols object (likely from `did2s`) that plugs into `honest_did`. Note this function assumes the event study coefficients are using `i()` syntax, e.g. `i(rel_year)`. This should also work for a TWFE event-study model estimated by `feols`.

Usage

```
get_honestdid_obj_did2s(est, coef_name = "rel_year")
```

Arguments

est	A `fixest` object, likely from `did2s`.
coef_name	Character. The name of the event-study relative-year variable name, from `i(rel_year)`.

Value

A list containing the vector of event-study coefficients `beta`, the variance-covariance matrix of `beta`, `V`, and a vector of relative years, `event_time`.

honest_did_did2s   *honest_did_did2s*

Description

a function to compute a sensitivity analysis using the approach of Rambachan and Roth (2021) when the event study is estimated using the did2s package. Note that you should first use the helper function get_honestdid_obj_did2s to create the object, obj, that you will then pass into this function with honest_did(obj)

Usage

```
honest_did_did2s(
  es,
  e = 0,
  type = c("smoothness", "relative_magnitude"),
  method = NULL,
  bound = "deviation from parallel trends",
  Mvec = NULL,
  Mbarvec = NULL,
  monotonicityDirection = NULL,
  biasDirection = NULL,
  alpha = 0.05,
  parallel = FALSE,
  gridPoints = 10^3,
  grid.ub = NA,
  grid.lb = NA,
  ...
)
```

Arguments

es	an object of class honestdid_obj_did2s from the function get_honestdid_obj_did2s
e	event time to compute the sensitivity analysis for. The default value is e=0 corresponding to the "on impact" effect of participating in the treatment.
type	Options are "smoothness" (which conducts a sensitivity analysis allowing for violations of linear trends in pre-treatment periods) or "relative_magnitude" (which conducts a sensitivity analysis based on the relative magnitudes of deviations from parallel trends in pre-treatment periods).
method	String that specifies the choice of method for constructing robust confidence intervals. This must be one of "FLCI", "Conditional", "C-F" (conditional FLCI hybrid), or "C-LF" (conditional least-favorable hybrid). Default equals NULL and the function automatically sets method based on the recommendations in Rambachan & Roth (2021) depending on the choice of Delta. If Delta = DeltaSD, default selects the FLCI. If Delta = DeltaSDB or DeltaSDM, default delects the conditional FLCI hybrid.

bound
: String that specifies the base choice of Delta (to which additional sign and shape restrictions will be incorporated if specified by the user). This must be either "deviation from parallel trends" or "deviation from linear trend". If bound equals "deviation from parallel trends", then the function will select $\Delta^{RM}(Mbar)$ as the base choice of Δ. If bound equals "deviation from linear trends", then the function will select Δ^{SDRM} as the base choice of Δ. By default, this is set to "deviation from parallel trends". See Section 2.3.1 and 2.3.2 of Rambachan & Roth (2021) for a discussion of these choices of Δ.

Mvec
: Vector of M values for which the user wishes to construct robust confidence intervals. If NULL, the function constructs a grid of length 10 that starts at M = 0 and ends at M equal to the upper bound constructed from the pre-periods using the function DeltaSD_upperBound_Mpre if number of pre-periods > 1 or the standard deviation of the first pre-period coefficient if number of pre-periods = 1. Default equals null.

Mbarvec
: Vector of Mbar values for which the user wishes to construct robust confidence intervals. If NULL, the function constructs a grid of length 10 that starts at Mbar = 0 and ends at Mbar = 2. Default equals null.

monotonicityDirection
: This must be specified if the user wishes to add an additional monotonicity restriction to $\Delta^{SD}(M)$. If "increasing", underlying trend specified to be increasing, $\delta_t \geq \delta_{t-1}$. If "decreasing", underlying trend specified to be decreasing $\delta_t \leq \delta_{t-1}$. Default equals NULL

biasDirection
: This must be specified if the user wishes to add an additional bias restriction to $\Delta^{SD}(M)$. If "positive", bias is restricted to be positive, $\delta \geq 0$. If "negative", bias is restricted to be negative, $\delta \leq 0$. Default equals NULL.

alpha
: Desired size of the robust confidence sets. Default equals 0.05 (corresponding to 95% confidence interval)

parallel
: Logical to indicate whether the user would like to construct the robust confidence intervals in parallel. This uses the Foreach package and doParallel package. Default equals FALSE.

gridPoints
: Number of grid points used for the underlying test inversion. Default equals 1000. User may wish to change the number of grid points for computational reasons.

grid.ub
: Upper bound of grid used for underlying test inversion. Default sets grid.ub to be equal to twenty times the standard deviation of the estimated target parameter, l_vec * betahat. User may wish to change the upper bound of the grid to suit their application.

grid.lb
: Lower bound of grid used for underlying test inversion. Default sets grid.lb to be equal to negative twenty times the standard deviation of the estimated target parameter, l_vec * betahat. User may wish to change the lower bound of the grid to suit their application.

...
: Ignored.

robust_solve_XtX *Robust solve for X'X beta = X'Y using QR decomposition*

Description

This function computes the least squares solution beta = (X'X)^(-1) X'Y in a numerically stable way using QR decomposition, handling rank-deficient matrices gracefully.

Usage

```
robust_solve_XtX(X, Y)
```

Arguments

X	Design matrix (sparse or dense)
Y	Response matrix/vector (can be X'Y if already computed)

Value

The least squares solution beta (may contain 0 for rank-deficient columns)

Index

∗ **datasets**
    castle, 2
    df_het, 3
    df_hom, 3

castle, 2

df_het, 3
df_hom, 3
did2s, 4

event_study, 8
event_study(), *8*

fixest::coefplot, *5*
fixest::esttable, *5*
fixest::feols, *5*
fixest::i, *5*

gen_data, 9
get_honestdid_obj_did2s, 10

honest_did_did2s, 11

plot_event_study *(event_study)*, 8

robust_solve_XtX, 13