

Package: NonlinearDiD (via r-universe)

September 20, 2026

Type Package

Title Staggered Difference-in-Differences with Nonlinear Outcomes

Version 0.2.0

Description Supports staggered difference-in-differences designs with nonlinear outcomes for both panel and repeated cross-section data. Implements estimators for staggered treatment adoption with binary, count, and other nonlinear outcomes, extending Callaway and Sant'Anna (2021) [<doi:10.1016/j.jeconom.2020.12.001>](https://doi.org/10.1016/j.jeconom.2020.12.001) to settings with nonlinear outcome models such as logit, probit, and Poisson. For panel data, units are followed over time and 'idname' identifies repeated observations. For repeated cross-section data, observations are independent within each time period; 'idname' is optional and may identify survey records or households, but the estimator does not require the same units to appear across periods. Repeated cross-section estimation includes pooled quasi-maximum likelihood approaches motivated by Wooldridge (2023) [<doi:10.1093/ectj/utad016>](https://doi.org/10.1093/ectj/utad016), with optional weighting and clustered inference. Methods also draw on Roth and Sant'Anna (2023) [<doi:10.3982/ECTA19402>](https://doi.org/10.3982/ECTA19402) and Sant'Anna and Zhao (2020) [<doi:10.1016/j.jeconom.2020.06.003>](https://doi.org/10.1016/j.jeconom.2020.06.003).

License MIT + file LICENSE

Encoding UTF-8

RoxygenNote 7.3.3

Depends R (>= 4.0.0)

Imports stats, utils, MASS, sandwich, lmtest, ggplot2

Suggests did, dplyr, knitr, rmarkdown, testthat (>= 3.0.0), covr

Config/testthat/edition 3

URL <https://github.com/causalfragility-lab/NonlinearDiD>

BugReports <https://github.com/causalfragility-lab/NonlinearDiD/issues>

NeedsCompilation no

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

Date/Publication 2026-05-21 11:01:54 UTC

RemoteUrl <https://github.com/subirhait/nonlineardid>

RemoteRef HEAD

RemoteSha b12f759a5ad360d345368c9824632ecb5d3dedf6

Contents

NonlinearDiD-package	2
binary_did_dr	4
binary_did_logit	6
binary_did_probit	7
count_did_poisson	8
nonlinear_aggte	9
nonlinear_attgt	10
nonlinear_bounds	13
nonlinear_pretest	14
nonlineardid_methods	15
odds_ratio_did	16
plot.nonlinear_aggte	17
plot.nonlinear_attgt	17
sim_binary_panel	18
sim_binary_rcs	19
sim_count_panel	20
Index	22

NonlinearDiD-package *NonlinearDiD: Staggered DiD with Nonlinear Outcomes*

Description

NonlinearDiD supports staggered difference-in-differences designs with nonlinear outcomes for both *panel* and *repeated cross-section* data.

For panel data, units are followed over time and `idname` identifies repeated observations. For repeated cross-section data, observations are independent within each time period; `idname` is optional and may identify survey records or households, but the estimator does not require the same units to appear across periods.

The package extends the Callaway and Sant’Anna (2021) framework to nonlinear outcome models, including binary (logit/probit), count (Poisson/NegBin), and odds-ratio estimands.

The Core Problem

The canonical CS2021 framework assumes parallel trends on the mean scale of a continuous outcome. For binary and count outcomes, this assumption is not scale-invariant: parallel trends in $P(Y=1)$ does NOT imply parallel trends in log-odds, pre-trend tests depend on which scale is used, and treatment effect estimates conflate true effects with Jensen’s inequality.

Main Functions

- `nonlinear_attgt()` – Estimate ATT(g,t) under nonlinear outcome models; supports panel and repeated cross-sections, with optional sampling weights (`weightsname`) and clustered inference (`cluster_var`).
- `nonlinear_aggte()` – Aggregate: event-study, group, calendar, overall.
- `nonlinear_pretest()` – Pre-treatment parallel trends test.
- `binary_did_logit()` – 2x2 DiD with logit outcome.
- `binary_did_probit()` – 2x2 DiD with probit outcome.
- `binary_did_dr()` – Doubly-robust binary DiD.
- `count_did_poisson()` – Poisson QMLE DiD for count outcomes.
- `odds_ratio_did()` – Odds-ratio DiD (scale-free).
- `nonlinear_bounds()` – Nonparametric Manski / PT bounds.
- `sim_binary_panel()` – Simulate binary staggered panel data.
- `sim_count_panel()` – Simulate count staggered panel data.
- `sim_binary_rcs()` – Simulate binary repeated cross-section data.

Quick Start: Panel

```
library(NonlinearDiD)
dat <- sim_binary_panel(n = 500, nperiods = 8, seed = 42)
res <- nonlinear_attgt(dat, yname = "y", tname = "period",
                      idname = "id", gname = "g",
                      outcome_model = "logit")
agg <- nonlinear_aggte(res, type = "dynamic")
plot(agg)
nonlinear_pretest(res)
```

Quick Start: Repeated Cross-Section

```
library(NonlinearDiD)
rcs <- sim_binary_rcs(n_per_period = 500, nperiods = 8, seed = 7)
res <- nonlinear_attgt(rcs, yname = "y", tname = "period",
                      gname = "g", outcome_model = "logit",
                      data_type = "repeated_cross_section",
                      estimand = "ape",
                      control_group = "notyetreated")
plot(nonlinear_aggte(res, type = "dynamic"))
```

Survey-Weighted Repeated Cross-Section Example

```
# Example: CPS-FSS-style data with survey weights and state clustering
# res <- nonlinear_attgt(
#   data      = my_survey_data,
#   yname     = "food_insecure",
```

```
# tname      = "year",
# gname      = "policy_end_year",
# idname     = "household_id",
# data_type  = "repeated_cross_section",
# outcome_model = "logit",
# estimand   = "ape",
# weightsname = "survey_weight",
# cluster_var = "state",
# control_group = "notyetreated"
# )
```

Author(s)

Maintainer: Subir Hait <haitsubi@msu.edu> ([ORCID](#))

References

- Callaway, B., & Sant'Anna, P. H. C. (2021). Difference-in-differences with multiple time periods. *Journal of Econometrics*, 225(2), 200-230.
- Roth, J., & Sant'Anna, P. H. C. (2023). When is parallel trends sensitive to functional form? *Econometrica*, 91(2), 737-747.
- Wooldridge, J. M. (2023). Simple approaches to nonlinear difference-in-differences with panel data. *The Econometrics Journal*, 26(3).
- Sant'Anna, P. H. C., & Zhao, J. (2020). Doubly robust difference-in-differences estimators. *Journal of Econometrics*, 219(1), 101-122.

See Also

Useful links:

- <https://github.com/causalfragility-lab/NonlinearDiD>
- Report bugs at <https://github.com/causalfragility-lab/NonlinearDiD/issues>

binary_did_dr

Doubly-Robust Binary DiD

Description

Doubly-robust estimator for binary outcomes combining a nonlinear outcome regression model with inverse probability weighting via propensity score. Consistent if EITHER the outcome model OR the propensity score is correctly specified.

Usage

```
binary_did_dr(
  data,
  yname,
  tname,
  idname,
  treat_period,
  control_period,
  dname = NULL,
  gname = NULL,
  xformula = ~1,
  outcome_model = c("logit", "probit"),
  se_type = c("robust", "cluster", "analytical"),
  cluster_var = NULL
)
```

Arguments

<code>data</code>	A data frame (long format).
<code>yname</code>	Character. Binary outcome variable name.
<code>tname</code>	Character. Time period variable name.
<code>idname</code>	Character. Unit ID variable name.
<code>treat_period</code>	Numeric. The treatment (post) period.
<code>control_period</code>	Numeric. The pre-treatment baseline period.
<code>dname</code>	Character. Treatment indicator variable name (optional).
<code>gname</code>	Character. Cohort variable name (optional).
<code>xformula</code>	One-sided formula for covariates. Default <code>~1</code> .
<code>outcome_model</code>	Character. "logit" (default) or "probit".
<code>se_type</code>	Character. SE type: "robust" (default), "cluster", or "analytical".
<code>cluster_var</code>	Character. Clustering variable (if <code>se_type = "cluster"</code>).

Value

A list of class `binary_did_dr`.

Examples

```
dat <- sim_binary_panel(n = 500, nperiods = 4, prop_treated = 0.5)
dat2 <- dat[dat$period %in% c(2, 3), ]
res <- binary_did_dr(dat2, "y", "period", "id", 3, 2, gname = "g",
  outcome_model = "logit")
print(res)
```

binary_did_logit

Binary Outcome DiD: Logit Estimator

Description

Estimates a 2x2 difference-in-differences model with a binary outcome using logistic regression on the log-odds scale, reporting both the log-odds DiD coefficient and the average partial effect (APE) on the probability scale.

Usage

```
binary_did_logit(
  data,
  yname,
  tname,
  idname,
  treat_period,
  control_period,
  dname = NULL,
  gname = NULL,
  xformula = ~1,
  se_type = c("robust", "cluster", "analytical"),
  cluster_var = NULL
)
```

Arguments

data	A data frame (long format).
yname	Character. Binary outcome variable name.
tname	Character. Time period variable name.
idname	Character. Unit ID variable name.
treat_period	Numeric. The treatment (post) period.
control_period	Numeric. The pre-treatment baseline period.
dname	Character. Treatment indicator variable name (optional).
gname	Character. Cohort variable name (optional).
xformula	One-sided formula for covariates. Default ~1.
se_type	Character. SE type: "robust" (default), "cluster", or "analytical".
cluster_var	Character. Clustering variable (if se_type = "cluster").

Value

A list of class `binary_did_logit`.

Examples

```

dat <- sim_binary_panel(n = 500, nperiods = 4, prop_treated = 0.5)
dat2 <- dat[dat$period %in% c(2, 3), ]
res <- binary_did_logit(dat2, yname = "y", tname = "period",
                        idname = "id", treat_period = 3,
                        control_period = 2, gname = "g")

print(res)

```

binary_did_probit	<i>Binary Outcome DiD: Probit Estimator</i>
-------------------	---

Description

Estimates 2x2 DiD with binary outcome using probit regression. Parallel trends assumed on the probit (inverse-normal) scale.

Usage

```

binary_did_probit(
  data,
  yname,
  tname,
  idname,
  treat_period,
  control_period,
  dname = NULL,
  gname = NULL,
  xformula = ~1,
  se_type = c("robust", "cluster", "analytical"),
  cluster_var = NULL
)

```

Arguments

data	A data frame (long format).
yname	Character. Binary outcome variable name.
tname	Character. Time period variable name.
idname	Character. Unit ID variable name.
treat_period	Numeric. The treatment (post) period.
control_period	Numeric. The pre-treatment baseline period.
dname	Character. Treatment indicator variable name (optional).
gname	Character. Cohort variable name (optional).
xformula	One-sided formula for covariates. Default ~1.
se_type	Character. SE type: "robust" (default), "cluster", or "analytical".
cluster_var	Character. Clustering variable (if se_type = "cluster").

Value

A list of class `binary_did_probit`.

Examples

```
dat <- sim_binary_panel(n = 500, nperiods = 4, prop_treated = 0.5)
dat2 <- dat[dat$period %in% c(2, 3), ]
res <- binary_did_probit(dat2, "y", "period", "id", 3, 2, gname = "g")
print(res)
```

count_did_poisson	<i>Count Outcome DiD: Poisson Estimator</i>
-------------------	---

Description

Estimates DiD for count outcomes using a Poisson quasi-maximum likelihood (QMLE) estimator with a log-linear parallel trends assumption. The treatment effect is a multiplicative rate ratio.

Usage

```
count_did_poisson(
  data,
  yname,
  tname,
  idname,
  treat_period,
  control_period,
  dname = NULL,
  gname = NULL,
  xformula = ~1,
  offset = NULL,
  se_type = c("robust", "cluster", "analytical"),
  cluster_var = NULL
)
```

Arguments

<code>data</code>	A data frame (long format).
<code>yname</code>	Character. Binary outcome variable name.
<code>tname</code>	Character. Time period variable name.
<code>idname</code>	Character. Unit ID variable name.
<code>treat_period</code>	Numeric. The treatment (post) period.
<code>control_period</code>	Numeric. The pre-treatment baseline period.
<code>dname</code>	Character. Treatment indicator variable name (optional).
<code>gname</code>	Character. Cohort variable name (optional).

xformula	One-sided formula for covariates. Default ~1.
offset	Character. Name of offset variable. Default NULL.
se_type	Character. SE type: "robust" (default), "cluster", or "analytical".
cluster_var	Character. Clustering variable (if se_type = "cluster").

Value

A list of class count_did_poisson.

Examples

```
dat <- sim_count_panel(n = 400, nperiods = 6, prop_treated = 0.4)
dat2 <- dat[dat$period %in% c(2, 4), ]
res <- count_did_poisson(dat2, "y", "period", "id", 4, 2, gname = "g")
print(res)
```

nonlinear_aggte	<i>Aggregate ATT(g,t) Estimates for Nonlinear DiD</i>
-----------------	---

Description

Aggregates the group-time average treatment effects from [nonlinear_attgt](#) into interpretable summary parameters. Provides event-study (dynamic), group-level, calendar-time, and overall ATT aggregations - each appropriate for nonlinear settings.

Usage

```
nonlinear_aggte(
  obj,
  type = c("dynamic", "group", "calendar", "simple"),
  na.rm = TRUE,
  min_periods = 1L,
  weights = c("equal", "sample")
)
```

Arguments

obj	An object of class nonlinear_attgt from nonlinear_attgt .
type	Character. The aggregation type: "dynamic": Event-study / dynamic treatment effects. Averages ATT(g,t) across groups g for each relative time e = t - g. "group": Group-specific ATT. Averages over post-treatment periods within each treated cohort g. "calendar": Calendar-time ATT. Averages over groups for each calendar time t. "simple": Overall average ATT, weighted by cohort size.

<code>na.rm</code>	Logical. Remove NA ATT(g,t) estimates. Default TRUE.
<code>min_periods</code>	Integer. Minimum number of ATT(g,t) observations required for an aggregated estimate to be reported. Default 1.
<code>weights</code>	Character. Weighting scheme for aggregation: • "equal": Equal-weight across (g,t) cells (default). • "sample": Weight by treated sample size in each (g,t).

Value

An object of class `nonlinear_aggte` with slots:

agg Data frame with aggregated ATT, SE, and CI.

type The aggregation type used.

overall_att Scalar overall ATT estimate.

overall_se SE for overall ATT.

Examples

```
set.seed(1)
dat <- sim_binary_panel(n = 400, nperiods = 8, prop_treated = 0.5)
res <- nonlinear_attgt(dat, yname = "y", tname = "period",
                      idname = "id", gname = "g",
                      outcome_model = "logit")
agg <- nonlinear_aggte(res, type = "dynamic")
plot(agg)
```

nonlinear_attgt

Nonlinear Staggered DiD: Group-Time ATT Estimation

Description

Computes group-time average treatment effects on the treated (ATT(g,t)) for staggered difference-in-differences designs with nonlinear outcomes. Supports both **panel** data (same units across periods) and **repeated cross-section** (RCS) data (independent samples per period).

For panel data the package follows Callaway & Sant'Anna (2021) and uses within-unit outcome changes to estimate counterfactual trends. For repeated cross-sections it uses the Wooldridge (2023) pooled QMLE with a treatment-by-period interaction (non-DR) or an IPW-augmented version (doubly-robust). Both modes optionally accept sampling weights and a clustering variable.

Usage

```

nonlinear_attgt(
  data,
  yname,
  tname,
  gname,
  idname = NULL,
  data_type = c("panel", "repeated_cross_section"),
  weightsname = NULL,
  cluster_var = NULL,
  xformula = ~1,
  outcome_model = c("logit", "probit", "poisson", "negbin", "linear"),
  estimand = c("att", "ape", "odds_ratio"),
  control_group = c("nevertreated", "notyetreated"),
  doubly_robust = TRUE,
  boot = FALSE,
  nboot = 999,
  boot_type = c("multiplier", "empirical"),
  alpha = 0.05,
  parallel = FALSE,
  pl_cores = 2L,
  anticipation = 0L
)

```

Arguments

<code>data</code>	A data frame in long format.
<code>yname</code>	Character. Outcome variable column.
<code>tname</code>	Character. Time period column.
<code>gname</code>	Character. Treatment cohort column (the period when a unit/group first receives treatment; 0 or Inf for never-treated).
<code>idname</code>	Character or NULL. Unit identifier column. Required for <code>data_type = "panel"</code> . Optional for <code>data_type = "repeated_cross_section"</code> .
<code>data_type</code>	Character. "panel" (default) or "repeated_cross_section".
<code>weightsname</code>	Character or NULL. Column name of sampling weights (e.g. survey design weights). Used in all model fits (outcome regression, propensity score, pooled QMLE) when supplied. Default NULL (equal weights).
<code>cluster_var</code>	Character or NULL. Column name to cluster standard errors on (e.g. "state"). Analytical SEs use <code>sandwich::vcovCL()</code> and the bootstrap resamples whole clusters. Default NULL (HC1 robust SEs / row resampling).
<code>xformula</code>	A one-sided formula for covariates (e.g. <code>~ x1 + x2</code>). Default <code>~ 1</code> .
<code>outcome_model</code>	Character. One of "logit", "probit", "poisson", "negbin", "linear".
<code>estimand</code>	Character. "att" (default), "ape" (average partial effect on probability scale), or "odds_ratio".
<code>control_group</code>	Character. "nevertreated" (default) or "notyetreated".

doubly_robust	Logical. Use the doubly-robust estimator. Default TRUE.
boot	Logical. Bootstrap inference. Default FALSE.
nboot	Integer. Bootstrap iterations. Default 999.
boot_type	Character. "multiplier" or "empirical".
alpha	Numeric. Significance level. Default 0.05.
parallel	Logical. Parallel bootstrap. Default FALSE.
pl_cores	Integer. Cores for parallel bootstrap.
anticipation	Integer. Periods of anticipation allowed. Default 0.

Value

An object of class nonlinear_attgt.

References

- Callaway, B., & Sant'Anna, P. H. C. (2021). Difference-in-differences with multiple time periods. *Journal of Econometrics*, 225(2), 200-230.
- Wooldridge, J. M. (2023). Simple approaches to nonlinear difference-in-differences with panel data. *The Econometrics Journal*, 26(3).
- Roth, J., & Sant'Anna, P. H. C. (2023). When is parallel trends sensitive to functional form? *Econometrica*, 91(2), 737-747.
- Sant'Anna, P. H. C., & Zhao, J. (2020). Doubly robust difference-in-differences estimators. *Journal of Econometrics*, 219(1), 101-122.

Examples

```
# ---- Panel example (v0.1.0 syntax - unchanged) ----
set.seed(42)
dat <- sim_binary_panel(n = 500, nperiods = 6, prop_treated = 0.4)
result <- nonlinear_attgt(
  data = dat, yname = "y", tname = "period",
  idname = "id", gname = "g",
  outcome_model = "logit"
)
summary(result)

# ---- Repeated cross-section example ----
set.seed(7)
rcs <- sim_binary_rcs(n_per_period = 400, nperiods = 6, prop_treated = 0.4)
res_rcs <- nonlinear_attgt(
  data = rcs, yname = "y", tname = "period", gname = "g",
  outcome_model = "logit",
  data_type = "repeated_cross_section"
)
summary(res_rcs)
```

nonlinear_bounds

*Nonparametric Bounds for Binary Outcomes in Staggered DiD***Description**

Computes sharp nonparametric bounds on the ATT for binary outcomes in staggered difference-in-differences designs, following the partial identification approach. These bounds require NO functional form assumptions on the outcome model - only an assumption about the direction or magnitude of selection.

The key insight for binary outcomes: Since Y is binary (0 or 1), the ATT is bounded by: - Lower bound: counterfactual never exceeds observed (pessimistic) - Upper bound: counterfactual never falls below observed (optimistic)

Under a Manski-style no-assumptions bound, plus refinements using the parallel trends assumption as a restriction.

Usage

```
nonlinear_bounds(
  data,
  yname,
  tname,
  idname,
  gname,
  xformula = ~1,
  control_group = c("nevertreated", "notyetreated"),
  bound_type = c("pt_only", "manski", "pt_monotone"),
  alpha = 0.05
)
```

Arguments

<code>data</code>	A long-format panel data frame.
<code>yname</code>	Character. Name of binary outcome variable (0/1).
<code>tname</code>	Character. Name of time period column.
<code>idname</code>	Character. Name of unit identifier.
<code>gname</code>	Character. Name of treatment cohort column.
<code>xformula</code>	One-sided formula for covariates. Default ' <code>~ 1</code> '.
<code>control_group</code>	Character. "nevertreated" (default) or "notyetreated".
<code>bound_type</code>	Character. Type of bound: "manski": No-assumptions Manski bounds (widest) "pt_monotone": Tighten using parallel trends + monotone treatment response "pt_only": Use only parallel trends restriction
<code>alpha</code>	Numeric. Significance level for confidence intervals on bounds.

Value

A data frame of sharp bounds (lb, ub) for $ATT(g,t)$, with bootstrap confidence intervals.

References

Manski, C. F. (1990). Nonparametric bounds on treatment effects. *American Economic Review*, 80(2), 319-323.

Callaway, B. (2021). Bounds on distributional treatment effect parameters. *Journal of Econometrics*, 222(2), 1084-1111.

Examples

```
set.seed(5)
dat <- sim_binary_panel(n = 300, nperiods = 6)
bounds <- nonlinear_bounds(dat, "y", "period", "id", "g")
print(bounds)
```

nonlinear_pretest

Pre-Treatment Parallel Trends Test for Nonlinear DiD

Description

Tests for pre-treatment violations of the parallel trends assumption in nonlinear staggered DiD settings. This is fundamentally different from the linear case because:

1. **Scale dependence**: Parallel trends on the probability scale does NOT imply parallel trends on the latent index scale (and vice versa). Tests are performed on the scale specified in 'outcome_model'.
2. **Roth-Sant'Anna sensitivity**: Computes sensitivity of post-treatment estimates to violations of magnitude delta in pre-period, following Roth & Sant'Anna (2023).
3. **Joint test**: Provides a joint chi-squared test of all pre-period $ATT(g,t) = 0$, accounting for correlation across (g,t) cells.

Usage

```
nonlinear_pretest(
  obj,
  plot = TRUE,
  alpha = 0.05,
  type = c("joint", "individual", "honestdid")
)
```

Arguments

<code>obj</code>	An object of class <code>nonlinear_attgt</code> .
<code>plot</code>	Logical. If TRUE (default), produces a pre-trends plot.
<code>alpha</code>	Numeric. Significance level. Default 0.05.
<code>type</code>	Character. Type of pre-trends test: • "joint": Joint chi-squared test (default) • "individual": Individual t-tests per pre-period cell • "honestdid": Sensitivity analysis a la Roth-Sant'Anna

Value

A list with:

pretest_results Data frame of pre-period ATT(g,t) with p-values.

joint_stat Joint test statistic.

joint_pval P-value for joint test.

conclusion Interpretive conclusion string.

References

Roth, J. (2022). Pretest with caution: Event-study estimates after testing for parallel trends. *American Economic Review: Insights**, 4(3), 305-322.

Roth, J., & Sant'Anna, P. H. C. (2023). When is parallel trends sensitive to functional form? *Econometrica**, 91(2), 737-747.

Examples

```
set.seed(99)
dat <- sim_binary_panel(n = 600, nperiods = 8, prop_treated = 0.5)
res <- nonlinear_attgt(dat, "y", "period", "id", "g",
                      outcome_model = "logit")
pt <- nonlinear_pretest(res)
print(pt)
```

Description

Print, summary, and plot methods for `nonlinear_attgt` and `nonlinear_aggte` objects.

odds_ratio_did	<i>Odds-Ratio DiD for Binary Outcomes</i>
----------------	---

Description

Estimates the odds-ratio difference-in-differences (OR-DiD) for binary outcomes. OR-DiD equals 1 under no treatment effect and is invariant to which group is labelled treatment.

Usage

```
odds_ratio_did(
  data,
  yname,
  tname,
  idname,
  treat_period,
  control_period,
  dname = NULL,
  gname = NULL,
  xformula = ~1
)
```

Arguments

data	A data frame (long format).
yname	Character. Binary outcome variable name.
tname	Character. Time period variable name.
idname	Character. Unit ID variable name.
treat_period	Numeric. The treatment (post) period.
control_period	Numeric. The pre-treatment baseline period.
dname	Character. Treatment indicator variable name (optional).
gname	Character. Cohort variable name (optional).
xformula	One-sided formula for covariates. Default ~1.

Value

A list of class odds_ratio_did.

Examples

```
dat <- sim_binary_panel(n = 500, nperiods = 4, prop_treated = 0.5)
dat2 <- dat[dat$period %in% c(2, 3), ]
res <- odds_ratio_did(dat2, "y", "period", "id", 3, 2, gname = "g")
print(res)
```

plot.nonlinear_aggte *Plot Aggregated DiD Estimates*

Description

Plots event-study, group-level, calendar, or overall aggregated ATT estimates from [nonlinear_aggte](#).

Usage

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

Arguments

x	An object of class nonlinear_aggte.
...	Additional arguments (unused).

Value

A ggplot2 object.

plot.nonlinear_attgt *Plot ATT(g,t) Estimates*

Description

Produces a faceted scatter plot of ATT(g,t) estimates with confidence intervals, one panel per treatment cohort.

Usage

```
## S3 method for class 'nonlinear_attgt'  
plot(x, ..., alpha = 0.05, point_size = 2)
```

Arguments

x	An object of class nonlinear_attgt.
...	Additional arguments (unused).
alpha	Numeric. Significance level for CI. Default 0.05.
point_size	Numeric. Size of estimate points. Default 2.

Value

A ggplot2 object.

sim_binary_panel

*Simulate Binary Panel Data with Staggered Treatment***Description**

Generates a simulated panel dataset with staggered treatment adoption and a binary outcome. Useful for testing and illustrating nonlinear DiD methods.

The data-generating process is:

$$Y_{it} = \mathbf{1}\{\alpha_i + \lambda_t + \delta_{it} \cdot D_{it} + \epsilon_{it} > 0\}$$

where α_i is a unit fixed effect, λ_t is a time fixed effect, δ_{it} is the treatment effect (heterogeneous across cohorts), and ϵ_{it} is logistic noise.

Usage

```
sim_binary_panel(
  n = 500L,
  nperiods = 6L,
  prop_treated = 0.5,
  n_cohorts = 3L,
  true_att = 0.3,
  base_prob = 0.3,
  unit_fe_sd = 0.5,
  add_covariates = TRUE,
  seed = NULL
)
```

Arguments

n	Integer. Number of units. Default 500.
nperiods	Integer. Number of time periods. Default 6.
prop_treated	Numeric. Proportion of units ever treated. Default 0.5.
n_cohorts	Integer. Number of treatment cohorts (groups). Default 3.
true_att	Numeric or vector. True ATT for each cohort. Default 0.3.
base_prob	Numeric. Baseline probability $P(Y=1)$ for untreated. Default 0.3.
unit_fe_sd	Numeric. Std. dev. of unit fixed effects. Default 0.5.
add_covariates	Logical. Add pre-treatment covariates. Default TRUE.
seed	Integer. Random seed. Default NULL.

Value

A data frame in long format. Columns: id (unit identifier), period (time period 1 to nperiods), y (binary outcome 0/1), g (treatment cohort; 0 = never treated), D (treatment indicator), x1 and x2 (covariates, if add_covariates = TRUE), and alpha_i (true unit fixed effect, for validation).

Examples

```
dat <- sim_binary_panel(n = 1000, nperiods = 8, prop_treated = 0.6,
                      n_cohorts = 4, true_att = c(0.2, 0.4, 0.3, 0.5))
head(dat)
table(dat$g)
```

sim_binary_rcs	<i>Simulate Binary Repeated Cross-Section Data with Staggered Treatment</i>
----------------	---

Description

Generates a simulated repeated cross-section (RCS) dataset with staggered treatment adoption and a binary outcome. At each time period an independent random sample is drawn from the population; no unit is observed more than once. This mirrors settings such as repeated population health surveys (e.g. BRFSS, NHIS) or administrative records linked by group membership rather than individual identifiers.

The data-generating process at period t for individual i belonging to treatment cohort g :

$$Y_{it} = \mathbf{1}\{\mu_0 + \lambda_t + \delta_g \cdot D_{gt} + \beta x_{1i} + \epsilon_{it} > 0\}$$

where $\mu_0 = \text{logit}(\text{base_prob})$, λ_t is a common time trend, δ_g is the cohort-specific treatment effect (on the log-odds scale), and $\epsilon_{it} \sim \text{Logistic}(0, 1)$ is i.i.d. noise. No unit-level fixed effect is included because individuals are not re-observed.

Usage

```
sim_binary_rcs(
  n_per_period = 500L,
  nperiods = 6L,
  prop_treated = 0.5,
  n_cohorts = 3L,
  true_att = 0.3,
  base_prob = 0.3,
  add_covariates = TRUE,
  seed = NULL
)
```

Arguments

n_per_period	Integer. Number of observations drawn per time period. Default 500.
nperiods	Integer. Number of time periods. Default 6.
prop_treated	Numeric. Proportion of individuals whose group is ever treated. Default 0.5.
n_cohorts	Integer. Number of treatment cohorts. Default 3.

true_att	Numeric or vector. True ATT (log-odds scale) for each cohort. Default 0.3.
base_prob	Numeric. Baseline $P(Y=1)$ in the absence of treatment. Default 0.3.
add_covariates	Logical. Add individual-level covariates x1 (continuous) and x2 (binary). Default TRUE.
seed	Integer. Random seed. Default NULL.

Details

There is no id column that repeats across periods. Use `nonlinear_attgt(..., data_type = "repeated_cross_section")` to analyse data of this type.

Value

A data frame in long format. One row per observation. Columns:

- obs_id** Unique observation identifier.
- period** Time period (1 to nperiods).
- y** Binary outcome (0/1).
- g** Treatment cohort of the observation's group (0 = never treated).
- D** Treatment indicator: 1 if the group is treated in this period.
- x1, x2** Individual-level covariates (if `add_covariates = TRUE`).

Examples

```
dat <- sim_binary_rcs(n_per_period = 500, nperiods = 6,
                     prop_treated = 0.5, true_att = 0.3, seed = 42)
head(dat)
table(dat$g, dat$period) # each cell is an independent sample

# Estimate ATT(g,t) under repeated cross-section design

res <- nonlinear_attgt(
  data = dat, yname = "y", tname = "period", gname = "g",
  outcome_model = "logit", data_type = "repeated_cross_section"
)
summary(res)
```

sim_count_panel

Simulate Count Panel Data with Staggered Treatment

Description

Generates simulated panel data with a count outcome (Poisson-distributed) and staggered treatment adoption. Treatment effect is multiplicative (rate ratio) on the count scale.

Usage

```
sim_count_panel(  
  n = 500L,  
  nperiods = 6L,  
  prop_treated = 0.5,  
  n_cohorts = 3L,  
  true_rr = 1.5,  
  base_rate = 5,  
  overdispersion = FALSE,  
  seed = NULL  
)
```

Arguments

n	Integer. Number of units. Default 500.
nperiods	Integer. Number of time periods. Default 6.
prop_treated	Numeric. Proportion of units ever treated. Default 0.5.
n_cohorts	Integer. Number of treatment cohorts. Default 3.
true_rr	Numeric or vector. True rate ratio for each cohort. Default 1.5 (50 percent increase in count).
base_rate	Numeric. Baseline Poisson rate. Default 5.
overdispersion	Logical. Add overdispersion (negative binomial). Default FALSE.
seed	Integer. Random seed.

Value

Long-format data frame with columns: id, period, y, g, D, x1.

Examples

```
dat <- sim_count_panel(n = 400, nperiods = 6, true_rr = 1.8)  
summary(dat$y)
```

Index

`binary_did_dr`, [4](#)
`binary_did_logit`, [6](#)
`binary_did_probit`, [7](#)

`count_did_poisson`, [8](#)

`nonlinear_aggte`, [9](#), [17](#)
`nonlinear_attgt`, [9](#), [10](#)
`nonlinear_bounds`, [13](#)
`nonlinear_pretest`, [14](#)
`NonlinearDiD` (`NonlinearDiD`-package), [2](#)
`NonlinearDiD`-package, [2](#)
`nonlineardid_methods`, [15](#)

`odds_ratio_did`, [16](#)

`plot.nonlinear_aggte`, [17](#)
`plot.nonlinear_attgt`, [17](#)

`sim_binary_panel`, [18](#)
`sim_binary_rcs`, [19](#)
`sim_count_panel`, [20](#)