

Package: spaci (via r-universe)

October 7, 2026

Title Causal Effect Estimation Under Spatial Confounding and Interference

Version 0.2.0

Description Implements the distance-adjusted propensity score with interference (iDAPS) and recoverU+ methods for estimating the average treatment effect on the treated (ATT) from spatial observational data in the presence of both spatial confounding and spatial interference. iDAPS matches units on a data-driven composite of propensity-score distance, spatial proximity and neighbourhood-exposure distance. recoverU+ is a doubly robust estimator that augments the propensity-score and control-outcome models with a partially recovered spatial confounder and a neighbourhood-exposure term. The package also provides the naive propensity score, DAPS and recoverU comparators, a simulator for the spatial confounding/interference data-generating process and spatial inference tools

License MIT + file LICENSE

Encoding UTF-8

Roxygen list(markdown = TRUE)

Depends R (>= 4.1.0)

Imports stats

Suggests clue, geoR, ggplot2, readxl, testthat (>= 3.0.0), knitr, rmarkdown

Config/testthat/edition 3

VignetteBuilder knitr

URL <https://github.com/Ogunsolaia/spaci>

BugReports <https://github.com/Ogunsolaia/spaci/issues>

Config/roxygen2/version 8.0.0

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

Date/Publication 2026-10-07 20:18:51 UTC

RemoteUrl <https://github.com/ogunsolaia/spaci>

RemoteRef HEAD

RemoteSha 6f91cdfb67a2a2144dfb7b7ac0af4a0232ab1e1

Contents

bias_bound	2
boot_spatial	3
daps	5
idaps	6
naive_ps	7
neighbourhood_exposure	8
plot.idaps_fit	9
plot_ate	9
print.idaps_fit	10
print.spaci_bias_bound	11
print.spaci_boot	11
print.spaci_randtest	12
print.spaci_vcov	12
rand_test	13
recoverU	14
recoverUplus	15
simulate_spatial_causal	16
spatial_ate	17
vcov_hac	18

Index	20
--------------	-----------

bias_bound	<i>Post-matching bias bound for an iDAPS fit</i>
------------	--

Description

Estimates an upper bound on the confounding bias of an `idaps()` matching estimate, following the variogram bias bound. A matched-pair difference is a spatial-differencing operation, so the estimator's bias decomposes into three pieces controlled by the matched sample:

$$|\text{bias}| \leq \|\theta_2\| \|\overline{\Delta X}\| + |\theta_3| |\overline{\Delta E}| + c |\theta_U| \sqrt{2\gamma_U(d)},$$

where $\Delta X, \Delta E$ are within-pair covariate and neighbourhood- exposure differences, γ_U is the (semi)variogram of the unmeasured spatial confounder evaluated at the matched distances d , and c is a selection constant (see `c_overlap`). The covariate and exposure coefficients are taken from a working outcome model, and the confounding term from the fitted spatial variogram of that model's residuals (whose sill carries $\theta_U^2 \text{Var}(U)$). The confounding term shrinks as the matched distances shrink, so a smaller caliper yields a smaller bound.

Usage

```

bias_bound(
  object,
  Y,
  Z,
  X,
  coords,
  tau = 0.1,
  c_overlap = 1.2,
  matern_method = c("mle", "geoR"),
  matern_nu = 0.5
)

```

Arguments

object	An <code>idaps_fit</code> returned by <code>idaps()</code> (supplies the matched pairs and neighbourhood exposure).
Y, Z, X, coords	The same outcome, treatment, covariates and coordinates passed to <code>idaps()</code> .
tau	Exposure-kernel bandwidth, used only if object lacks a stored exposure (default 0.1).
c_overlap	Selection inflation constant on the confounding term (default 1.2). Matching selects units with systematically larger confounder values, so the marginal variogram slightly understates the within-pair confounder differences; $c_overlap \approx 1.2$ restores a finite-sample-valid bound (set to 1 for the population-level bound).
matern_method, matern_nu	Passed to the residual variogram fit; see <code>recoverU()</code> .

Value

An object of class `spaci_bias_bound`: a list with the covariate, exposure and confounding bound terms (`term_X`, `term_E`, `term_U`), their total, the mean matched distance, and the fitted variogram parameters.

Examples

```

sim <- simulate_spatial_causal(n = 50, seed = 1)
fit <- idaps(sim$Y, sim$Z, sim$X, sim$coords, tau = 0.1, seed = 1)
bias_bound(fit, sim$Y, sim$Z, sim$X, sim$coords)

```

Description

Resamples square spatial blocks of units with replacement to build a bootstrap distribution that reflects spatial dependence. For the doubly robust estimators (recoverU/recoverU+) it block-resamples the estimator's influence values (a fast, stable linearised bootstrap that avoids re-kriging resampled coordinates); for the matching estimators it re-runs the whole pipeline on each resample (recomputing neighbourhood exposures, with a tiny coordinate jitter to avoid exact-duplicate ties).

Usage

```
boot_spatial(
  Y,
  Z,
  X,
  coords,
  method = c("recoverUplus", "recoverU", "idaps", "daps", "naive_ps"),
  block_size = "auto",
  B = 200,
  tau = 0.1,
  caliper = 0.25,
  matern_method = c("mle", "geoR"),
  matern_nu = 0.5,
  level = 0.95,
  seed = NULL
)
```

Arguments

Y, Z, X, coords	Data, as passed to the estimators.
method	Estimator to bootstrap: one of "recoverUplus", "recoverU", "idaps", "daps", "naive_ps".
block_size	Side length of the square blocks: "auto" (default, domain width $\times n^{(-1/4)}$, giving on the order of $\sqrt{n}$ blocks) or a positive number. The block side should exceed the spatial dependence range.
B	Number of bootstrap resamples (default 200).
tau, caliper, matern_method, matern_nu	Estimator arguments (as relevant).
level	Confidence level.
seed	Optional integer seed for reproducibility.

Value

An object of class `spaci_boot` with the bootstrap `se`, percentile `ci`, point estimate, and the retained resample estimates.

Examples

```
sim <- simulate_spatial_causal(n = 50, seed = 1)
boot_spatial(sim$Y, sim$Z, sim$X, sim$coords, method = "recoverUplus", B = 50)
```

daps

*Distance-adjusted propensity score (DAPS) matching ATT***Description**

Estimates the ATT by matching on a convex combination of propensity-score distance and spatial (Euclidean) distance, $\alpha D^{PS} + (1 - \alpha) D^{Spatial}$, adjusting for spatial confounding but not interference. The weight α is chosen over a grid to minimise covariate imbalance in the matched sample (Papadogeorgou et al., 2019).

Usage

```
daps(
  Y,
  Z,
  X,
  coords,
  caliper = 0.25,
  alpha_grid_step = 0.1,
  match_method = c("greedy", "optimal"),
  seed = NULL,
  level = 0.95
)
```

Arguments

Y	Numeric outcome vector.
Z	Binary treatment vector (0/1).
X	Covariate matrix or data frame.
coords	Two-column matrix or data frame of spatial coordinates.
caliper	Maximum acceptable matching distance (default 0.25).
alpha_grid_step	Grid step for the α search over $[0, 1]$.
match_method	Matching algorithm: "greedy" (default) matches treated units in random order to their nearest available control (depends on seed); "optimal" solves the 1:1 assignment minimising total distance (deterministic, needs the clue package).
seed	Optional integer seed for the (randomised) matching order.
level	Confidence level for the reported interval.

Value

An `idaps_fit` object; the selected α is reported in `weights`.

Examples

```
sim <- simulate_spatial_causal(n = 50, seed = 1)
daps(sim$Y, sim$Z, sim$X, sim$coords)
```

idaps

Distance-adjusted propensity score with interference (iDAPS)

Description

Estimates the ATT by matching on the composite distance of Equation (2.3), $\pi_1 D^{PS} + \pi_2 D^{Spatial} + \pi_3 D^{Interference}$, where the interference component is the absolute difference in neighbourhood exposure. The weights (π_1, π_2, π_3) sum to one and are chosen over a grid to minimise a composite balance score (covariate balance, spatial proximity and exposure balance), rather than tuned by hand. Setting $\pi_3 = 0$ recovers DAPS and $\pi_2 = \pi_3 = 0$ recovers naive PS.

Usage

```
idaps(
  Y,
  Z,
  X,
  coords,
  tau = 0.1,
  caliper = 0.25,
  pi_grid_step = 0.1,
  normalize = TRUE,
  match_method = c("greedy", "optimal"),
  seed = NULL,
  level = 0.95
)
```

Arguments

Y	Numeric outcome vector.
Z	Binary treatment vector (0/1).
X	Covariate matrix or data frame.
coords	Two-column matrix or data frame of spatial coordinates.
tau	Positive spatial decay parameter of the exposure kernel (default 0.1); see neighbourhood_exposure() .
caliper	Maximum acceptable matching distance (default 0.25).
pi_grid_step	Grid step for the weight search over the simplex.
normalize	Logical; row-normalise the exposure kernel (default TRUE).
match_method	Matching algorithm: "greedy" (default) matches treated units in random order to their nearest available control (depends on seed); "optimal" solves the 1:1 assignment minimising total distance (deterministic, needs the clue package).
seed	Optional integer seed for the (randomised) matching order.
level	Confidence level for the reported interval.

Value

An idaps_fit object; the selected weights are reported in weights.

Examples

```
sim <- simulate_spatial_causal(n = 50, seed = 1)
idaps(sim$Y, sim$Z, sim$X, sim$coords, tau = 0.1)
```

naive_ps

*Naive propensity-score matching ATT***Description**

Estimates the ATT by greedy 1:1 matching on the (normalised) difference in estimated propensity scores. This is the baseline that ignores both spatial confounding and spatial interference.

Usage

```
naive_ps(
  Y,
  Z,
  X,
  caliper = 0.25,
  match_method = c("greedy", "optimal"),
  seed = NULL,
  level = 0.95
)
```

Arguments

Y	Numeric outcome vector.
Z	Binary treatment vector (0/1).
X	Covariate matrix or data frame.
caliper	Maximum acceptable matching distance (default 0.25).
match_method	Matching algorithm: "greedy" (default) matches treated units in random order to their nearest available control (depends on seed); "optimal" solves the 1:1 assignment minimising total distance (deterministic, needs the clue package).
seed	Optional integer seed for the (randomised) matching order.
level	Confidence level for the reported interval.

Value

An idaps_fit object.

Examples

```
sim <- simulate_spatial_causal(n = 50, seed = 1)
naive_ps(sim$Y, sim$Z, sim$X)
```

neighbourhood_exposure

Neighbourhood exposure via a spatial kernel

Description

Constructs the neighbourhood-exposure mapping $E_i = \sum_{j \neq i} G_{ij} A_j$ that summarises spatial interference, where G_{ij} is an exponential kernel of the distance between units i and j , $G_{ij} = \exp(-d_{ij}/\tau)$ with $G_{ii} = 0$. This is the exposure used by `idaps()` and the `recoverU` family.

Usage

```
neighbourhood_exposure(coords, Z, tau = 0.1, normalize = TRUE)
```

Arguments

<code>coords</code>	A two-column matrix or data frame of spatial coordinates.
<code>Z</code>	Binary treatment vector (0/1), one entry per row of <code>coords</code> .
<code>tau</code>	Positive spatial decay (bandwidth) parameter of the exponential kernel. Smaller values concentrate exposure on nearer neighbours.
<code>normalize</code>	Logical; if TRUE (default) each row of the kernel is normalised to sum to one, so that exposure is a weighted average of neighbours' treatment. If FALSE, the raw kernel weights are used, matching the unnormalised definition in Equation (2.4) of the report.

Value

A list with components

E Numeric vector of neighbourhood exposures, one per unit.

G The (possibly row-normalised) kernel matrix with zero diagonal.

Examples

```
set.seed(1)
coords <- cbind(runif(20), runif(20))
Z <- rbinom(20, 1, 0.5)
ex <- neighbourhood_exposure(coords, Z, tau = 0.1)
head(ex$E)
```

plot.idaps_fit	<i>Plot a single estimate</i>
----------------	-------------------------------

Description

Plot a single estimate

Usage

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

Arguments

x	An idaps_fit object.
...	Passed to plot_ate() .

Value

The plotted data frame, invisibly.

plot_ate	<i>Forest plot of ATT estimates across methods</i>
----------	--

Description

Draws a forest (caterpillar) plot of the estimated ATT and its confidence interval for each method, with a reference line at zero and, optionally, at the true effect. This reproduces the style of Figure 2.5 of the report and is the recommended way to compare methods visually.

Usage

```
plot_ate(
  x,
  true_att = NULL,
  null_line = 0,
  col = "#4C72B0",
  pt_col = "#1A1A1A",
  xlab = "Average treatment effect on the treated (ATT)",
  main = "Estimated ATT and confidence intervals",
  ...
)
```

Arguments

x	Either the data frame returned by <code>spatial_ate()</code> (columns Method, ATT, Lower, Upper) or a named list of <code>idaps_fit</code> objects.
true_att	Optional true effect; drawn as a dashed vertical reference line when supplied.
null_line	Position of the "no effect" reference line (default 0); set to NA to omit.
col, pt_col	Colours for the confidence intervals and point estimates.
xlab, main	Axis label and title.
...	Passed to <code>graphics::plot()</code> .

Value

The plotted data frame, invisibly.

Examples

```
sim <- simulate_spatial_causal(n = 50, seed = 1)
res <- spatial_ate(sim$Y, sim$Z, sim$X, sim$coords, seed = 1)
plot_ate(res, true_att = sim$true_att)
```

print.idaps_fit	<i>Print method for idaps_fit objects</i>
-----------------	---

Description

Print method for `idaps_fit` objects

Usage

```
## S3 method for class 'idaps_fit'
print(x, digits = 4, ...)
```

Arguments

x	An <code>idaps_fit</code> object.
digits	Number of significant digits.
...	Ignored.

Value

x, invisibly.

```
print.spaci_bias_bound
```

Print method for a bias bound

Description

Print method for a bias bound

Usage

```
## S3 method for class 'spaci_bias_bound'  
print(x, digits = 4, ...)
```

Arguments

x	A spacibias_bound object.
digits	Number of significant digits.
...	Ignored.

Value

x, invisibly.

```
print.spaci_boot
```

Print method for a spatial block bootstrap

Description

Print method for a spatial block bootstrap

Usage

```
## S3 method for class 'spaci_boot'  
print(x, digits = 4, ...)
```

Arguments

x	A spaciboot object.
digits	Significant digits.
...	Ignored.

Value

x, invisibly.

`print.spaci_randtest` *Print method for a randomization test*

Description

Print method for a randomization test

Usage

```
## S3 method for class 'spaci_randtest'  
print(x, digits = 4, ...)
```

Arguments

<code>x</code>	A <code>spaci_randtest</code> object.
<code>digits</code>	Significant digits.
<code>...</code>	Ignored.

Value

`x`, invisibly.

`print.spaci_vcov` *Print method for a spatial HAC variance*

Description

Print method for a spatial HAC variance

Usage

```
## S3 method for class 'spaci_vcov'  
print(x, digits = 4, ...)
```

Arguments

<code>x</code>	A <code>spaci_vcov</code> object.
<code>digits</code>	Significant digits.
<code>...</code>	Ignored.

Value

`x`, invisibly.

rand_test

*Conditional randomization test for no direct effect***Description**

Design-based test of the sharp null that the treatment has no direct effect on any unit. Treatment is redrawn from the estimated propensity score (so the reference distribution respects the observed treatment mechanism), neighbourhood exposures are recomputed, and the estimator is recomputed for each draw. The two-sided p-value is $(1 + \#\{|\hat{\tau}^{(r)}| \geq |\hat{\tau}|\})/(1 + R)$.

Usage

```
rand_test(
  Y,
  Z,
  X,
  coords,
  method = c("recoverUplus", "recoverU", "idaps", "daps", "naive_ps"),
  R = 200,
  tau = 0.1,
  caliper = 0.25,
  matern_method = c("mle", "geoR"),
  matern_nu = 0.5,
  seed = NULL
)
```

Arguments

Y, Z, X, coords	Data, as passed to the estimators.
method	Estimator to bootstrap: one of "recoverUplus", "recoverU", "idaps", "daps", "naive_ps".
R	Number of randomization draws (default 200).
tau, caliper, matern_method, matern_nu	Estimator arguments (as relevant).
seed	Optional integer seed for reproducibility.

Value

An object of class `spaci_randtest` with the observed statistic, the p-value and the null draws.

Examples

```
sim <- simulate_spatial_causal(n = 50, seed = 1)
rand_test(sim$Y, sim$Z, sim$X, sim$coords, method = "naive_ps", R = 50)

# refits the full recoverU+ pipeline on every draw, so it is slower
rand_test(sim$Y, sim$Z, sim$X, sim$coords, method = "recoverUplus", R = 20)
```

 recoverU

recoverU: doubly robust ATT with a recovered spatial confounder

Description

Doubly robust ATT estimator that adjusts for spatial confounding through a partially recovered spatial confounder (Pokal et al., 2023), but does not adjust the propensity-score / control-outcome models for spatial interference. Provided as a comparator to [recoverUpplus\(\)](#).

Usage

```
recoverU(
  Y,
  Z,
  X,
  coords,
  tau = 0.1,
  normalize = TRUE,
  matern_method = c("mle", "geoR"),
  matern_nu = 0.5,
  level = 0.95
)
```

Arguments

Y	Numeric outcome vector.
Z	Binary treatment vector (0/1).
X	Covariate matrix or data frame.
coords	Two-column matrix or data frame of coordinates.
tau	Positive spatial decay parameter for the exposure kernel used in the initial outcome model (default 0.1).
normalize	Logical; row-normalise the exposure kernel (default TRUE).
matern_method	Matern estimation engine: "mle" (default, no external dependency) or "geoR" (reproduces the reference analysis).
matern_nu	Fixed Matern smoothness for the "mle" engine (default 0.5, an exponential covariance, which is numerically stable); pass NULL to estimate the smoothness freely. Ignored by the "geoR" engine.
level	Confidence level for the reported interval.

Value

An `idaps_fit` object.

Examples

```
sim <- simulate_spatial_causal(n = 50, seed = 1)
recoverU(sim$Y, sim$Z, sim$X, sim$coords)
```

recoverUplus	<i>recoverU+</i> : doubly robust ATT under spatial confounding and interference
--------------	---

Description

The recoverU+ estimator augments the doubly robust ATT with both a partially recovered spatial confounder and a neighbourhood-exposure term, so that the propensity-score and control-outcome models adjust for spatial confounding and spatial interference simultaneously (Equations 2.10-2.12). Compared with [recoverU\(\)](#), the neighbourhood exposure G is included in the PS and control-outcome models.

Usage

```
recoverUplus(
  Y,
  Z,
  X,
  coords,
  tau = 0.1,
  normalize = TRUE,
  matern_method = c("mle", "geoR"),
  matern_nu = 0.5,
  level = 0.95
)
```

Arguments

Y	Numeric outcome vector.
Z	Binary treatment vector (0/1).
X	Covariate matrix or data frame.
coords	Two-column matrix or data frame of coordinates.
tau	Positive spatial decay parameter for the exposure kernel used in the initial outcome model (default 0.1).
normalize	Logical; row-normalise the exposure kernel (default TRUE).
matern_method	Matern estimation engine: "mle" (default, no external dependency) or "geoR" (reproduces the reference analysis).
matern_nu	Fixed Matern smoothness for the "mle" engine (default 0.5, an exponential covariance, which is numerically stable); pass NULL to estimate the smoothness freely. Ignored by the "geoR" engine.
level	Confidence level for the reported interval.

Value

An idaps_fit object.

Examples

```
sim <- simulate_spatial_causal(n = 50, seed = 1)
recoverUplus(sim$Y, sim$Z, sim$X, sim$coords)
```

```
simulate_spatial_causal
```

Simulate spatial data with confounding and interference

Description

Draws one data set from the data-generating process of the report (Equation 2.13 and Section 2.3): an unmeasured spatial confounder $U(s)$ drawn from an exponential Gaussian random field drives both treatment and outcome, and the outcome further depends on a neighbourhood-exposure term so that spatial interference is present. The true ATT is true_att.

Usage

```
simulate_spatial_causal(
  n = 150,
  true_att = 2,
  beta0 = 2.5,
  beta1 = 1,
  beta2 = 0.5,
  theta_spatial = 0.4,
  gamma_interference = 1.5,
  sigma_eps = 1,
  delta_u = 2,
  u_phi = 0.2,
  tau_exp = 0.1,
  normalize = TRUE,
  seed = NULL
)
```

Arguments

n	Number of units.
true_att	True average treatment effect on the treated.
beta0, beta1, beta2	Outcome-model intercept and covariate coefficients.
theta_spatial	Coefficient of the spatial confounder in the outcome.
gamma_interference	Coefficient of neighbourhood exposure in the outcome (interference strength).

sigma_eps	Outcome noise standard deviation.
delta_u	Strength of the spatial confounder in the treatment model (controls the degree of spatial confounding).
u_phi	Range parameter of the confounder's exponential covariance.
tau_exp	Spatial decay parameter of the exposure kernel.
normalize	Logical; row-normalise the exposure kernel (default TRUE).
seed	Optional integer seed.

Details

The Gaussian random field is generated directly from its covariance matrix (exponential covariance, i.e. Matern with smoothness 1/2), so the simulator has no external dependency.

Value

A list with Y, Z, X (matrix with columns X1, X2), coords, the exposure E, the latent confounder U, and true_att.

Examples

```
sim <- simulate_spatial_causal(n = 50, seed = 42)
str(sim)
```

spatial_ate

Estimate the ATT with all available methods

Description

Convenience wrapper that runs the naive propensity score, DAPS, iDAPS, recoverU and recoverU+ estimators on the same data and collects the point estimates, standard errors and confidence intervals in a single data frame.

Usage

```
spatial_ate(
  Y,
  Z,
  X,
  coords,
  tau = 0.1,
  caliper = 0.25,
  matern_method = c("mle", "geoR"),
  level = 0.95,
  seed = NULL,
  methods = c("Naive PS", "DAPS", "iDAPS", "recoverU", "recoverU+")
)
```

Arguments

Y	Numeric outcome vector.
Z	Binary treatment vector (0/1).
X	Covariate matrix or data frame.
coords	Two-column matrix or data frame of spatial coordinates.
tau	Positive spatial decay parameter of the exposure kernel (default 0.1); see neighbourhood_exposure() .
caliper	Maximum acceptable matching distance (default 0.25).
matern_method	Matern estimation engine for the recoverU family, "mle" (default) or "geoR".
level	Confidence level for the reported interval.
seed	Optional integer seed for the (randomised) matching order.
methods	Character vector selecting which estimators to run. Defaults to all five.

Value

A data frame with one row per method (Method, ATT, SE, Lower, Upper), with the fitted objects attached as the "fits" attribute.

Examples

```
sim <- simulate_spatial_causal(n = 50, seed = 1)
spatial_ate(sim$Y, sim$Z, sim$X, sim$coords, seed = 1)
```

vcov_hac

*Spatial HAC (Conley) variance for a doubly robust ATT***Description**

Computes a spatial heteroskedasticity- and autocorrelation-consistent (Conley) variance for a [recoverU\(\)](#) or [recoverUplus\(\)](#) estimate,

$$\widehat{\text{Var}}(\hat{\tau}) = \frac{1}{n^2} \sum_i \sum_j K\left(\frac{\|s_i - s_j\|}{b}\right) \hat{\psi}_i \hat{\psi}_j,$$

where $\hat{\psi}_i$ are the estimator's influence values and K is a kernel with bandwidth b . Unlike the default i.i.d. standard error, this accounts for spatial correlation of the influence values, which otherwise makes intervals too narrow.

Usage

```
vcov_hac(
  object,
  bandwidth = "auto",
  kernel = c("bartlett", "uniform"),
  level = 0.95
)
```

Arguments

object	A recoverU/recoverU+ idaps_fit (carries the influence values and coordinates). Matching estimators have no influence-value representation; use boot_spatial() for those.
bandwidth	Kernel bandwidth: "auto" (default, estimated from the influence-value autocorrelation range) or a positive number.
kernel	"bartlett" (default, guarantees a non-negative variance) or "uniform".
level	Confidence level for the returned interval.

Value

An object of class `spaci_vcov`: a list with the HAC variance, se, confidence interval ci, the att, and the bandwidth/kernel used.

Examples

```
sim <- simulate_spatial_causal(n = 50, seed = 1)
fit <- recoverUplus(sim$Y, sim$Z, sim$X, sim$coords)
vcov_hac(fit)
```

Index

`bias_bound`, [2](#)
`boot_spatial`, [3](#)
`boot_spatial()`, [19](#)

`daps`, [5](#)

`graphics::plot()`, [10](#)

`idaps`, [6](#)
`idaps()`, [2](#), [3](#), [8](#)

`naive_ps`, [7](#)
`neighbourhood_exposure`, [8](#)
`neighbourhood_exposure()`, [6](#), [18](#)

`plot.idaps_fit`, [9](#)
`plot_ate`, [9](#)
`plot_ate()`, [9](#)
`print.idaps_fit`, [10](#)
`print.spaci_bias_bound`, [11](#)
`print.spaci_boot`, [11](#)
`print.spaci_randtest`, [12](#)
`print.spaci_vcov`, [12](#)

`rand_test`, [13](#)
`recoverU`, [14](#)
`recoverU()`, [3](#), [15](#), [18](#)
`recoverUplus`, [15](#)
`recoverUplus()`, [14](#), [18](#)

`simulate_spatial_causal`, [16](#)
`spatial_ate`, [17](#)
`spatial_ate()`, [10](#)

`vcov_hac`, [18](#)