The Stata Journal (2024)
24, Number 3, pp. 371–401 DOI: 10.1177/1536867X241276108

Getting away from the cutoff in regression discontinuity designs

Filippo Palomba
Princeton University
Princeton, NJ
fpalomba@princeton.edu
https://orcid.org/0000-0002-6400-3142

Abstract. Regression discontinuity (RD) designs are highly popular in economic research because of their strong internal validity and straightforward intuition. While RD estimates are local in nature, several recent articles propose methods that generalize RD estimates to units outside a small neighborhood of the cutoff. In this article, I introduce the `getaway` package, which implements the method proposed by Angrist and Rokkanen (2015, *Journal of the American Statistical Association* 110: 1331–1344) to extrapolate treatment-effect estimates "away from the cutoff", relying on a classical unconfoundedness condition. Additionally, the package features a data-driven algorithm designed to identify a set of covariates that fulfills the unconfoundedness assumption. It also incorporates a toolkit intended for testing and visualization purposes.

Keywords: st0751, getaway, ciasearch, ciatest, ciares, ciacs, getawayplot, regression discontinuity designs, treatment effects

1 Introduction

Since its first appearance in Thistlethwaite and Campbell (1960) and rigorous formalization in Hahn, Todd, and van der Klaauw (2001), the regression discontinuity (RD) design has gained extreme popularity in empirical work in several fields of economics: political (Lee, Moretti, and Butler 2004; Pettersson-Lidbom 2008; Meyersson 2014; Bronzini and Iachini 2014); development (Ozier 2018); health (Ludwig and Miller 2007); crime (Pinotti 2017); education (Angrist and Lavy 1999; Cellini, Ferreira, and Rothstein 2010; Duflo, Dupas, and Kremer 2011; Pop-Eleches and Urquiola 2013); and public (Lalive 2008; Battistin et al. 2009; Coviello and Mariniello 2014) and corporate finance (Flammer 2015).

In the typical RD design, units are assigned to treatment depending on the value of a covariate (score or running variable) being above or below a certain threshold (cutoff). This creates a conditional probability of being assigned to treatment that jumps at the cutoff point, generating random variation in treatment status that can be used to identify causal parameters. Intuitively, units whose score value is close to the cutoff can be thought of as lying on different sides by chance, "as if" they were randomly assigned to treatment. By comparing their posttreatment outcomes, one can therefore identify a local average treatment effect (τ^{RD}, henceforth).

Because the above-mentioned identification assumption holds only in a neighborhood of the cutoff, the identified causal parameter refers to a very narrow population. Recently, scholars have proposed alternative ways to extend the causal estimate to different populations. Battistin and Rettore (2008) establish conditions for interpreting τ^{RD} as the average treatment effect on the treated, utilizing one-sided noncompliance in a fuzzy RD. Dong and Lewbel (2015) show how to identify the derivative of the treatment effect at the cutoff under a local policy invariance assumption. Bertanha (2020) identifies average treatment effects computed over general counterfactual distributions of individuals, rather than over those units in a neighborhood of the cutoff. In fuzzy RDs, Bertanha and Imbens (2020) propose a testing procedure to extrapolate the average treatment effect for compliers to other subpopulations (at the cutoff). Finally, Cattaneo et al. (2021) use the presence of multiple cutoffs and a parallel-trend-type assumption to extrapolate the average treatment effect to values of the score comprised between (at least) two cutoffs.

In this article, I focus on the methodology proposed in Angrist and Rokkanen (2015) (henceforth, AR) and introduce the `getaway` package to implement it. To extrapolate treatment effects away from the cutoff, AR exploit additional information contained in explanatory variables other than the score to estimate treatment effects away from the cutoff. The methodology hinges on a conditional independence assumption (CIA), which requires mean independence between potential outcomes and the score variable conditional on a vector of other covariates, together with a common support condition. Therefore, in contrast to the other procedures described above, AR leverages additional information from an external set of variables to identify treatment effects for (potentially) any other population, as long as the CIA holds.

AR describe a setting with a single-cutoff RD design. Leveraging existing results in the RD literature, one can extend such a setting to the case with more than one cutoff. If no unit is located exactly at each cutoff ("exogenously" determined cutoffs), all the results in Cattaneo et al. (2021) hold, and a simple normalizing-and-pooling (NP) estimator—together with a vector of covariates satisfying the CIA—can be used to extrapolate treatment effects away from the cutoff. If instead a marginally exposed unit is located exactly at each cutoff ("endogenously" determined cutoffs), the identification of average causal effects could be troublesome (Fort et al. 2022). In this case, the probability limit of the standard NP estimator may differ from the desired causal parameter—for example, the average treatment effect at the cutoff—despite identification conditions holding at each single cutoff. The suggested solution is to recover identification using a "site fixed-effect" estimation strategy. The site fixed-effect estimator can be obtained by augmenting the standard RD regression with fixed effects at the site level, where a "site" is defined as the group of units facing the same cutoff.

In addition, `getaway` is equipped with a data-driven algorithm to search for a vector of covariates that satisfies the CIA. This algorithm was originally proposed to choose a model for the propensity score in Imbens and Rubin (2015). At each iteration, the algorithm selects the covariate that makes the CIA condition more likely to be satisfied, which turns out to be the covariate with the highest p-value (lowest test statistic) when testing the CIA implications in the data. Although the proposed algorithm performs multiple hypothesis tests, I explicitly do not rely on any form of multiple testing adjustment. The reason is that the algorithm aims at being conservative; thus, rejecting the null more often than the level of the test would simply add some redundant covariates to the ones selected to satisfy the CIA condition.

Last, the article describes each of the six different commands contained in `getaway`: `ciasearch` applies a data-driven algorithm that selects an adequate set of covariates to "get away" from the cutoff; `ciatest` tests the CIA assumption for a given set of covariates; `ciares` produces graphical visualization of an implication of the CIA; `ciacs` finds the common support region and reports histograms for the estimated propensity score; `getaway` parametrically estimates treatment effects away from the cutoff; `getawayplot` plots the estimated potential outcomes as functions of the score variable.

The rest of the article is organized as follows. Section 2 gives an overview of the methods implemented in the `getaway` package. Sections 3–8 describe the syntax of `ciasearch`, `ciatest`, `ciacs`, `ciares`, `getaway`, and `getawayplot`, respectively. Section 9 gives numerical illustrations, and section 9 concludes.

2 Overview of methods

In an RD design, all observed units receive a score. The score (or running variable) is the dimension along which units are ordered and assigned to the treatment or the control group. Units with a value of the score above the cutoff are assigned to treatment; those with score values lower than the cutoff are assigned to the control group.[1]

To formalize, let $i \in \{1, 2, \dots, n\}$ be an index for the observed units, Y_i be an outcome variable of interest, D_i be a dummy variable denoting assignment to treatment, X_i be the (scalar) running variable, and c the cutoff.[2] In an RD design, the assignment to treatment follows a deterministic rule known at least to the researcher; that is,

$$D_i = \begin{cases} 1, & \text{if } X_i \geq c \\ 0, & \text{if } X_i < c \end{cases}$$

If there is full compliance with the treatment, which means that assignment to treatment and actual treatment status are identical, the RD design is said to be sharp. If this is not the case, then the RD design with imperfect compliance is called fuzzy.[3]

1. For simplicity, I assume that assignment to treatment is a nondecreasing function of the score. Of course, the opposite holds if units with low values are assigned to treatment.
2. The reader interested in the case of multiple running variables should refer to Keele and Titiunik (2015), Cattaneo, Idrobo, and Titiunik (2024), and references therein.
3. For brevity, in this work I describe only the sharp case, but all the arguments presented here hold also for the fuzzy case with appropriate modifications.

2.1 Treatment effect at the cutoff

The key feature of the RD design is that the probability of being assigned to the treatment conditional on the score changes discontinuously at the cutoff. Indeed, the RD design exploits this source of exogenous variation to identify an average treatment effect locally at the cutoff. Drawing on the potential outcome framework (Rubin 1974), I define $Y_i(0)$ and $Y_i(1)$ as the potential outcomes that would be observed if $D_i = 0$ or $D_i = 1$, respectively. Hence, we can define the observed outcome as

$$Y_i = Y_i(1)D_i + Y_i(0)(1 - D_i)$$

and the local average treatment effect as

$$\tau^{\text{RD}} := \mathbb{E}\left\{Y_i(1) - Y_i(0) \mid X_i = c\right\} \tag{1}$$

From (1), it follows that to identify τ^{RD}, a researcher would have to compare treatment and control units when $X_i = c$. However, the RD design constitutes an extreme case of lack of common support, because units belonging to the treatment group have different score values from units in the control group. To solve this problem, Hahn, Todd, and van der Klaauw (2001) propose to compare units lying immediately to the right and to the left of the cutoff. Heuristically, units close to the cutoff should be similar to each other. Formally, the authors show that τ^{RD} is identified as long as the following two assumptions hold:

1. The probability of receiving the treatment jumps at the cutoff c; that is,

$$\lim_{x \downarrow c} \Pr(D_i = 1 \mid X_i = x) \neq \lim_{x \uparrow c} \Pr(D_i = 1 \mid X_i = x)$$

2. Potential outcomes are continuous at the cutoff; that is,

$$\lim_{x \to c} \mathbb{E}\left\{Y_i(0) \mid X_i = x\right\}, \quad \lim_{x \to c} \mathbb{E}\left\{Y_i(1) \mid X_i = x\right\}$$

are continuous in X_i at c.

In particular, the local average treatment effect is identified as

$$\tau^{\text{RD}} = \mathbb{E}\left\{Y_i(1) - Y_i(0) \mid X_i = c\right\} = \lim_{x \downarrow c} \mathbb{E}\left(Y_i \mid X_i = x\right) - \lim_{x \uparrow c} \mathbb{E}\left(Y_i \mid X_i = x\right)$$

In practice, estimation of τ^{RD} requires choosing an appropriate bandwidth for the running variable. A bandwidth is a segment of width h in the support of the running variable that defines the units used for estimation, that is, those units such that $X_i \in [c - h, c + h]$. Choosing the bandwidth h entails a bias-variance tradeoff. Indeed, choosing a wider bandwidth (large h) uses more units to estimate τ^{RD} (lower variance) but includes units that are far from the cutoff and may differ sensibly in terms of observable and unobservable characteristics (higher bias). The opposite holds for a smaller bandwidth.[4]

Thus, despite having strong internal validity, standard RD designs may lack external validity. By construction, the causal parameter τ^{RD} is local, because it measures the average treatment effect for units having $X_i = c$. Without imposing additional assumptions or exploiting further information, nothing can be said about units with different values of the running variable.

2.2 Treatment effect away from the cutoff

The RD estimator can be thought of as a special case of selection on observables (Heckman, LaLonde, and Smith 1999). Indeed, in RD designs omitted variable bias can stem only from the running variable X_i (Goldberger 2008). Large differences in the score of two units are likely to reflect discrepancies in terms of observables and unobservables. Ultimately, this is why i) RD designs rely only on units close to the cutoff to estimate τ^{RD}; ii) comparing units away from the cutoff is liable to selection bias; and iii) identification and estimation become harder when the target is a causal parameter of a population broader than the one composed of units at the cutoff.

2.2.1 Identification

AR build on the intuition developed in Goldberger (2008) and propose to identify and estimate treatment effects away from the cutoff relying on a set of predictors of the dependent variable other than the running variable. Using notation from Lee (2008), the authors model the running variable X_i as a measurable function $g(.,.)$ of some observed ($\mathbf{w}_i$) and unobserved ($\mathbf{u}_i$) covariates; that is, $X_i = g(\mathbf{w}_i, \mathbf{u}_i)$. In this model, conditional on $\mathbf{w}_i$, the only randomness in X_i, hence in D_i, comes through $\mathbf{u}_i$. If the observable predictors $\mathbf{w}_i$ make the running variable `ignorable`—that is, independent of potential outcomes—then one can use them to move away from the cutoff. Such a condition is termed the CIA, and it holds if and only if

$$\mathbb{E}\{Y_i(j) \mid X_i, \mathbf{w}_i\} = \mathbb{E}\{Y_i(j) \mid \mathbf{w}_i\}, \quad j = 0, 1 \tag{2}$$

and

$$0 < \Pr(D_i = 1 \mid \mathbf{w}_i) < 1, \text{ a.s.} \tag{3}$$

In words, condition (2) requires potential outcomes to be independent in mean of the running variable once the set of covariates $\mathbf{w}_i$ is taken into account. Condition (3) is a matching-style common support assumption that requires the treatment dummy to be nondegenerate within each cell induced by the vector $\mathbf{w}_i$. The CIA breaks the link

4. The reader interested in how to find the optimal bandwidth may want to refer to Imbens and Kalyanaraman (2012), Calonico, Cattaneo, and Titiunik (2014), Cattaneo, Frandsen, and Titiunik (2015), or Armstrong and Kolesár (2018). Moreover, in this work I assume for simplicity a symmetric bandwidth on each side of the cutoff. Generalizations to different bandwidths on each side of the cutoff ($h_L \neq h_R$) are straightforward and are an available option in the `getaway` package.

between the running variable and the potential outcomes, so that the vector $\mathbf{w}_i$ can be used in place of the running variable to identify and estimate various causal effects. This can be done precisely because the only possible source of bias in an RD design is the running variable. To fix ideas, suppose a researcher is interested in identifying the treatment effect at $X_i \in \mathcal{A}$, where $\mathcal{A}$ is a nonempty set contained in the support of the running variable. In this case the estimand of interest is $\tau_{\mathcal{A}} := \mathbb{E}\{Y_i(1) - Y_i(0) \mid X_i \in \mathcal{A}\}$. If (2) and (3) hold, then $\tau_{\mathcal{A}}$ can be estimated as

$$\tau_{\mathcal{A}} = \mathbb{E}\{\mathbb{E}(Y_i \mid \mathbf{w}_i, D_i = 1) - \mathbb{E}(Y_i \mid \mathbf{w}_i, D_i = 0) \mid X_i \in \mathcal{A}\} \tag{4}$$

▷ Example

If a researcher is interested in the average treatment on the treated (ATT), then $\mathcal{A} = [c, \infty)$ and

$$\tau_{[c,\infty)} = \mathbb{E}\{\mathbb{E}(Y_i \mid \mathbf{w}_i, D_i = 1) - \mathbb{E}(Y_i \mid \mathbf{w}_i, D_i = 0) \mid X_i \geq c\}$$

If the average treatment on the nontreated (ATNT) is of interest, then $\mathcal{A} = (-\infty, c)$ and

$$\tau_{(-\infty,c)} = \mathbb{E}\{\mathbb{E}(Y_i \mid \mathbf{w}_i, D_i = 1) - \mathbb{E}(Y_i \mid \mathbf{w}_i, D_i = 0) \mid X_i < c\}$$

◁

2.2.2 Estimation

To find an estimator for $\tau_{\mathcal{A}}$ in (4), it is sufficient to rely on the plugin principle. The crucial part regards the estimation of the inner conditional expectation $\mathbb{E}(Y_i \mid \mathbf{w}_i, D_i)$. AR propose two main estimators:

1. The conditional expectation $\mathbb{E}(Y_i \mid \mathbf{w}_i, D_i)$ can be modeled linearly in $\mathbf{w}_i$ as

$$\mathbb{E}(Y_i \mid \mathbf{w}_i, D_i = 1) = \mathbf{w}_i' \beta_R, \quad \mathbb{E}(Y_i \mid \mathbf{w}_i, D_i = 0) = \mathbf{w}_i' \beta_L$$

   Then, substituting it in (4), we get the following linear reweighting estimator (Kline 2011):

$$\mathbb{E}\{Y_i(1) - Y_i(0) \mid X_i \in \mathcal{A}\} = (\beta_R - \beta_L)' \mathbb{E}(\mathbf{w}_i \mid X_i \in \mathcal{A}) \tag{5}$$

2. A propensity-score weighting estimator in the spirit of Hirano, Imbens, and Ridder (2003) is

$$\mathbb{E}\{Y_i(1) - Y_i(0) \mid X_i \in \mathcal{A}\} = \mathbb{E}\left(\frac{Y_i\{D_i - p(X_i)\}}{p(X_i)\{1 - p(X_i)\}} \times \frac{\Pr(X_i \in \mathcal{A} \mid \mathbf{w}_i)}{\Pr(X_i \in \mathcal{A})}\right) \tag{6}$$

   where $p(X_i) := \Pr(D_i = 1 \mid X_i)$ is a probability model for the propensity score.

2.2.3 Testing

The CIA in (2) has (partially) testable implications because the RD design provides a test for the assumption that conditioning on the vector of observables $\mathbf{w}_i$ removes selection bias. Indeed, in RD designs the running variable is the only source of omitted variable bias; hence, if $\mathbf{w}_i$ breaks the link between the potential outcomes and the running variable, there is no omitted variable bias. Thus, testing the CIA boils down to testing whether X_i has a statistically significant effect on Y_i conditionally on $\mathbf{w}_i$. Formally, implications of the CIA can be tested empirically by running the following regressions,

$$\begin{aligned} Y_i &= \alpha_L + X_i\gamma_{L,1} + \cdots + X_i^p\gamma_{L,p} + \mathbf{w}_i'\beta_L + \varepsilon_i, \quad \text{if} \quad X_i < c \\ Y_i &= \alpha_R + X_i\gamma_{R,1} + \cdots + X_i^q\gamma_{R,q} + \mathbf{w}_i'\beta_R + \nu_i, \quad \text{if} \quad X_i \geq c \end{aligned}$$

and testing the null hypotheses that

$$H_0^{(L)} : \gamma_{L,1} = \cdots = \gamma_{L,p} = 0 \quad \text{and} \quad H_0^{(R)} : \gamma_{R,1} = \cdots = \gamma_{R,q} = 0$$

where $p \in \mathbb{N}$ and $q \in \mathbb{N}$ are the degree of the polynomial in the score to the left and to the right of the cutoff, respectively. If there is not enough evidence to reject these null hypotheses, then there is no evidence in favor of the CIA not being satisfied by the vector of covariates $\mathbf{w}_i$.

2.3 RD with multiple rankings

In the recent literature, the case of RD with multiple cutoffs has received a lot of attention (see Cattaneo, Titiunik, and Vazquez-Bare [2020] and references therein). A popular practice is to normalize all the cutoffs to a common value (zero) and pool together units belonging to different rankings. This practice defines the NP estimator mentioned previously.[5] Formally, let X_{is} be the running variable for unit i in site $s \in \{1, 2, \ldots, S\}$ and c_s be the cutoff value that determines assignment to treatment in site (ranking) s. Often, researchers create a normalized version of the score using the formula $\widetilde{X}_i = X_{is} - c_s$. NP allows researchers to rewrite the assignment-to-treatment function as $D_i = \mathbb{1}(\widetilde{X}_i > 0)$. By doing so, they obtain a single estimate using all available data and hence increase the statistical power and precision of the estimates. Cattaneo et al. (2021) outline that the NP estimand is a weighted average of the RD estimates at each cutoff, where the weights depend on the number of units around each cutoff. This means that this estimand averages out any source of treatment heterogeneity, so attention should be paid when interpreting this causal parameter.

An example of this setting is a sequence of calls for tenders conducted in different areas of a country where local firms are ranked according to their bids. Firms operating in different regions will not compete against each other, and different cutoff values may determine their assignment to treatment. This is not problematic if the cutoff is

5. In principle, one can estimate a separate RD at each cutoff. However, in practice it usually happens that the available sample is limited in size and RDs estimated in this way typically lack statistical power.

determined ex ante in each ranking as in the setting described in Cattaneo et al. (2021) (the so-called “exogenous cutoff”). If instead the cutoff coincides with the marginal subject exposed to treatment in each ranking (“endogenous cutoff”), the traditional NP estimator does not estimate any causal parameter of interest.

As pointed out in Fort et al. (2022), in settings where the cutoffs are determined endogenously, the NP estimator uses inappropriate weights for those observations above the cutoff. This issue arises because the score variable has a mass point exactly at the cutoff. Thus, the NP estimator is biased with respect to τ^{RD}, and the size of the bias depends on the covariance between the potential outcomes of units when treated and their relative frequency, in a neighborhood of the cutoff. This bias goes away asymptotically, as the number of units in each ranking goes to infinity, which is unlikely the case in most applications. The safest solution to this problem is introducing fixed-effect dummies at the site level. In this spirit, the `getaway` package extends the work done in AR to the multiple cutoff case, introducing the possibility to add site-level fixed effects to the RD estimator.

2.4 Data-driven algorithm

The package relies on a data-driven algorithm in the spirit of Imbens and Rubin (2015, chap. 13.3) that searches for a vector of covariates satisfying the CIA condition. Formally, suppose there is a set of k covariates $\mathcal{C}$, which is the union of two disjoint sets:

- a set $\mathcal{C}_1 \subset \mathcal{C}$ containing $k_1 < k$ covariates to be included in $\mathbf{w}_i$ independently of their relationship with the outcome variable. These are covariates that the researcher views as a priori important in explaining the selection bias between units with high and low values of the score. In principle, in the absence of a relevant theory driving the choice of these covariates or simply if the researcher has little substantive knowledge, it could be that $\mathcal{C}_1 = \emptyset$.
- a set $\mathcal{C}_2 \subseteq \mathcal{C}$ containing $k_2 \leq k$ candidate covariates that could be included in $\mathbf{w}_i$ with the purpose of making the running variable ignorable.

▷ Example

Suppose that X_i is a score denoting the quality of a project of a firm, Y_i is employment growth, and the natural experiment takes place in Italy. Previous theoretical and empirical work suggests that employment growth is inversely related to a firm’s age (Jovanovic 1982). In addition, the Italian context is characterized by a prevalence of very small firms (Bartelsman, Scarpetta, and Schivardi 2005); thus, the presence of managers and white collar workers may be an important source of competitive advantage. Therefore, a researcher might want to include firm size, firm age, presence of manager in the firm, and presence of white collar workers in the firm in $\mathcal{C}_1$.

◁

The algorithm searches for sets $\widetilde{\mathcal{C}}_L, \widetilde{\mathcal{C}}_R \subseteq \mathcal{C}_2$ such that $\widetilde{\mathcal{C}}_L \cup \mathcal{C}_1$ and $\widetilde{\mathcal{C}}_R \cup \mathcal{C}_1$ make the running variable ignorable to the left and to the right of the cutoff, respectively. The algorithm is composed of the following steps:

1. Let ι be the number of covariates in $\mathcal{C}_2$ that have already been selected.[6] Run the following set of regressions for $j = 1, \ldots, k_2 - \iota$,

$$Y_{is} = \sum_{\ell=1}^{p} X_{is}^{\ell} \gamma_{L,\ell} + \mathbf{z}'_{L,is}\delta_L + \omega_{is}^{(j)}\mu_{L,j} + \alpha_{L,s} + \varepsilon_{is}, \quad \text{if } X_{is} < c_s$$
$$Y_{is} = \sum_{\ell=1}^{q} X_{is}^{\ell} \gamma_{R,\ell} + \mathbf{z}'_{R,is}\delta_R + \omega_{is}^{(j)}\mu_{R,j} + \alpha_{R,s} + \nu_{is}, \quad \text{if } X_{is} \geq c_s \tag{7}$$

   where $\mathbf{z}_{\bullet,is}$ is the vector of $k_1 + \iota$ covariates that have already been included, h is the bandwidth, $\alpha_{\bullet,s}$ are fixed effects at the site level (Fort et al. 2022), and $\omega_{is}^{(j)} \in \mathcal{C}_2$ is the jth candidate covariate.[7] Notice that the RD design with a single cutoff or with multiple exogenous cutoffs is a particular specification of (7), in which the site fixed effects $\alpha_{\bullet,s}$ become a pooled constant $\alpha_\bullet$.

2. For each regression in (7), conduct an F test for the null hypothesis that the CIA holds (separately) on each side of the cutoff,

$$H_0^{(L)} : \gamma_{L,1} = \cdots = \gamma_{L,p} = 0 \quad \text{and} \quad H_0^{(R)} : \gamma_{R,1} = \cdots = \gamma_{R,q} = 0$$

   and store the two F statistics $F^{j,L}$ and $F^{j,R}$.

3. Select the two covariates associated with each smallest F statistic in the two sets

$$\mathcal{F}^L = \{F^{1,L}, F^{2,L}, \ldots, F^{k_2-\iota,L}\}, \quad \mathcal{F}^R = \{F^{1,R}, F^{2,R}, \ldots, F^{k_2-\iota,R}\}$$

   Denote these two variables with $\omega^\star_{L,is}$ and $\omega^\star_{R,is}$, respectively. Notice that nothing prevents the variable with the smallest F statistic on the left of the cutoff to differ from one on the right of the cutoff; that is, it can be that $\omega^\star_{L,is} \neq \omega^\star_{R,is}$.

4. Add $\omega^\star_{L,is}$ and $\omega^\star_{R,is}$ to $\widetilde{\mathcal{C}}_L$ and $\widetilde{\mathcal{C}}_R$, respectively, and to $\mathbf{z}_{\bullet,is}$ in (7).

5. Repeat steps 1–4 until one of the following stopping criteria is reached:

   - The null hypothesis that the running variable is not significantly different from 0 cannot be rejected at the $\alpha\%$ level, where α is chosen by the researcher;
   - all the covariates in $\widetilde{\mathcal{C}}$ have been included in $\widetilde{\mathcal{C}}_L$ and $\widetilde{\mathcal{C}}_R$.

6. Alternatively, ι can be thought of as the iteration of the algorithm minus one.

7. When the algorithm starts ($\iota = 0$), we run the regression of Y_{is} onto the covariates in $\mathcal{C}_1$, a polynomial in the running variable, and (if present) the site fixed effects, and we add one single candidate covariate.

The basic idea behind the algorithm is to implement a greedy approach (James et al. 2023), meaning that the best variable is selected at each particular step, rather than looking ahead and picking a variable that will lead to a larger reduction in the loss function in some future step. This is done to avoid testing all the possible combinations of the elements of $\mathcal{C}_2$.[8]

Three caveats are needed. First, as already outlined, it can happen that the algorithm selects two different sets of covariates on each side of the cutoff. The proposed heuristic here is to define the final set of covariates as the union between the sets of covariates satisfying the CIA on each side of the cutoff.

Second, in some cases the heuristic approach above might not work. Indeed, while not being a problem in population, in the sample it could be that the union of two nonidentical sets of covariates satisfying the CIA on separate sides of the cutoff does not satisfy the CIA simultaneously on both sides. Thus, a more stringent version of the algorithm can be implemented. This alternative algorithm has a different step 3, in which it selects a unique covariate $\tilde{\omega}_i$ that minimizes a single loss function of the form $\mathcal{L}(\mathcal{F}^{j,L}, \mathcal{F}^{j,R})$, rather than minimizing $\mathcal{L}(\mathcal{F}^L)$ and $\mathcal{L}(\mathcal{F}^R)$ separately. When required to choose a single covariate at each step, the package `getaway` uses $\mathcal{L}(\mathcal{F}^{j,L}, \mathcal{F}^{j,R}) = \min\{\mathcal{F}^{j,L}, \mathcal{F}^{j,R}\}$. On the one hand, this feature makes the algorithm more demanding, because it selects a set of covariates that satisfies the CIA condition on both sides of the cutoff at the same time. On the other hand, this version has the advantage of not relying on the heuristic solution of using the union of the two selected groups of covariates.

Third, an obvious but well-due remark. If the algorithm above does not yield a set of covariates that satisfies the CIA, then there are two solutions: the researcher either gives up extrapolation away from the cutoff or gathers more information (that is, variables). A similar remark applies when the common support condition (3) is not satisfied.

3 The ciasearch command

This section describes the syntax of the command `ciasearch`, which implements the algorithm described in section 2.4, which searches for a vector of covariates $\mathbf{w}$ satisfying the CIA condition (2). The common support condition (3) can be verified using the command `ciacs` (see section 6) once a candidate $\mathbf{w}$ has been found with `ciasearch`.

8. This exercise would soon become intractable from a computational point of view because it involves estimating $\sum_{i=1}^{k_2} \binom{k_2}{i}$ different regressions. To quantify this issue, with 10 covariates, the number of different combinations to be tested for is 1,023. This case is still tractable. However, adding just 10 other covariates drives the number of combinations to over 1 million.

3.1 Syntax

`ciasearch` *varlist* [*if*] [*in*], `outcome(`*varname*`)` `score(`*varname*`)` `bandwidth(`*#*`)` [`cutoff(`*#*`)` `included(`*varlist*`)` `poly(`*numlist*`)` `robust` `vce(`*varname*`)` `site(`*varname*`)` `alpha(`*#*`)` `quad` `unique` `force` `noprint`]

where *varlist* specifies the set of candidates $\mathcal{C}_2$ and the option `included()` eventually specifies the set of always-included covariates $\mathcal{C}_1$.

3.2 Options

`outcome(`*varname*`)` specifies the dependent variable of interest. `outcome()` is required.

`score(`*varname*`)` specifies the running variable. `score()` is required.

`bandwidth(`*#*`)` specifies the value for the bandwidth to be used for estimation. The user can specify a different bandwidth for each side. `bandwidth()` is required.

`cutoff(`*#*`)` specifies the value of the cutoff. The default is `cutoff(0)`. The cutoff value is subtracted from the `score()` variable and the bandwidth. When multiple cutoffs are present, provide the pooled cutoff.

`included(`*varlist*`)` specifies the set of covariates that are always included in the testing regression.

`poly(`*numlist*`)` specifies the degree of the polynomial function in the running variable. The user can specify a different degree for each side. The default is `poly(1 1)`.

`robust` estimates heteroskedasticity-robust standard errors.

`vce(`*varname*`)` specifies the clustered standard errors at the specified level.

`site(`*varname*`)` specifies the variable identifying the site to add site fixed effects.

`alpha(`*#*`)` specifies the level of type I error in the CIA test. The default is `alpha(0.1)`. In this case, the higher the value of `alpha()`, the easier it will be to reject the null hypothesis that the CIA condition holds. Notice that `alpha()` implicitly defines the threshold value for algorithm convergence.

`quad` adds to *varlist* squared terms of each (nondichotomic) covariate in *varlist* and interactions of all the covariates in *varlist*

`unique` runs a single algorithm on both sides. This version selects a unique set of covariates that satisfies the CIA condition on both sides of the cutoff at the same time.

`force` causes the algorithm to forget the value of the loss function at the iteration $j-1$ and select the covariate providing the lower value of the loss function at iteration j. In other words, with this option, the algorithm searches for the covariate that minimizes the loss function within a certain iteration. This can make the loss function nonstrictly decreasing in the number of iterations but allows the algorithm to select covariates that provide a sensible gain only after some steps.

`noprint` suppresses within-iteration results.

4 The ciatest command

This section describes the syntax of the `ciatest` command, which tests whether the CIA condition (2) holds for a given input vector of covariates $\mathbf{w}_i$. The command `ciatest` is the "manual" version of `ciasearch`.

4.1 Syntax

`ciatest` *varlist* [*if*] [*in*], `outcome(`*varname*`)` `score(`*varname*`)` `bandwidth(`#`)` [`cutoff(`#`)` `poly(`*numlist*`)` `robust` `vce(`*varname*`)` `site(`*varname*`)` `alpha(`#`)` `details` `noise`]

where *varlist* specifies the vector $\mathbf{w}_i$.

4.2 Options

`outcome(`*varname*`)` specifies the dependent variable of interest. `outcome()` is required.

`score(`*varname*`)` specifies the running variable. `score()` is required.

`bandwidth(`#`)` specifies the value for the bandwidth to be used for estimation. The user can specify a different bandwidth for each side. `bandwidth()` is required.

`cutoff(`#`)` specifies the value of the cutoff. The default is `cutoff(0)`. The cutoff value is subtracted from the `score()` variable and the bandwidth. When multiple cutoffs are present, provide the pooled cutoff.

`poly(`*numlist*`)` specifies the degree of the polynomial function in the running variable. The user can specify a different degree for each side. The default is `poly(1 1)`.

`robust` specifies the estimated heteroskedasticity-robust standard errors.

`vce(`*varname*`)` specifies the clustered standard errors at the specified level.

`site(`*varname*`)` specifies the variable identifying the site to add site fixed effects.

`alpha(`#`)` specifies the level of type I error in the CIA test. The default is `alpha(0.1)`. In this case, the higher the value of `alpha()`, the easier it will be to reject the null hypothesis that the CIA condition holds. Notice that `alpha()` implicitly defines the threshold value for algorithm convergence.

`details` reports results of additional tests in the output. The `details` option reports the main statistics of the simple regression of `outcome()` on `score()` in both the restricted sample and full sample. The restricted sample is the sample composed by all units with no missing values in `outcome()`, `score()`, and *varlist*, while the full sample is defined as those units with no missing entries only in `outcome()` and `score()`. This additional check is particularly useful when there are missing values in *varlist*.

`noise` prints all testing regression outputs.

5 The ciares command

This section describes the syntax of the `ciares` command, which provides a graphical visualization of the CIA condition (2) for a given input vector of covariates $\mathbf{w}_i$.

5.1 Syntax

`ciares` *varlist* [*if*] [*in*], `outcome(`*varname*`)` `score(`*varname*`)` `bandwidth(`#`)`
[`cutoff(`#`)` `nbins(`*numlist*`)` `site(`*varname*`)` `cmpr(`*numlist*`)`
`gphoptions(`*string*`)` `scatterplotopt(`*string*`)` `scatter2plotopt(`*string*`)`
`lineLplotopt(`*string*`)` `lineRplotopt(`*string*`)` `lineL2plotopt(`*string*`)`
`lineR2plotopt(`*string*`)` `legendopt(`*string*`)`]

where *varlist* specifies the vector $\mathbf{w}_i$.

5.2 Options

`outcome(`*varname*`)` specifies the dependent variable of interest. `outcome()` is required.

`score(`*varname*`)` specifies the running variable. `score()` is required.

`bandwidth(`#`)` specifies the value for the bandwidth to be used for estimation. The user can specify a different bandwidth for each side. `bandwidth()` is required.

`cutoff(`#`)` specifies the value of the cutoff. The default is `cutoff(0)`. The cutoff value is subtracted from the `score()` variable and the bandwidth. When multiple cutoffs are present, provide the pooled cutoff.

`nbins(`*numlist*`)` specifies the number of bins in which the average of residuals should be computed. The number of bins can be specified for each side of the cutoff. The default is `nbins(10 10)`.

`site(`*varname*`)` specifies the variable identifying the site to add site fixed effects.

`cmpr(`*numlist*`)` adds the conditional regression function of `outcome()` on the `score()`. The form of polynomials on the left and on the right can be modeled independently—for example, `cmpr(2 3)` for a second-order polynomial on the left and a third-order one on the right.

`gphoptions(`*string*`)` specifies graphical options to be passed on to the underlying `graph` command. These options overwrite the default formatting options of the command.

`scatterplotopt(`*string*`)` specifies graphical options to be passed on to the underlying scatterplot.

`scatter2plotopt(`*string*`)` specifies graphical options to be passed on to the underlying scatterplot (secondary axis).

`lineLplotopt(`*string*`)` specifies graphical options to be passed on to the underlying line plot (left of cutoff).

`lineRplotopt(`*string*`)` specifies graphical options to be passed on to the underlying line plot (right of cutoff).

`lineL2plotopt(`*string*`)` specifies graphical options to be passed on to the underlying line plot (left of cutoff, secondary axis).

`lineR2plotopt(`*string*`)` specifies graphical options to be passed on to the underlying line plot (right of cutoff, secondary axis).

`legendopt(`*string*`)` specifies graphical options to be passed on to the underlying plot legend.

6 The ciacs command

This section describes the syntax of the command `ciacs`, which provides a graphical visualization of the common support condition (3) required to validate the CIA. The command `ciacs` allows the user to fit a logistic or probit model for the treatment variable D_i using $\mathbf{w}_i$ (and fixed effects if required) as explanatory variables. Furthermore, `ciacs` allows to graphically visualize whether the common support assumption holds.

6.1 Syntax

`ciacs` *varlist* [*if*] [*in*], `outcome(`*varname*`)` `assign(`*varname*`)` `score(`*varname*`)`
`bandwidth(`#`)` [`cutoff(`#`)` `nbins(`*numlist*`)` `site(`*varname*`)` `asis`
`gphoptions(`*string*`)` `pscore(`*string*`)` `probit` `kdensity` `nograph`
`legendopt(`*string*`)` `barTopt(`*string*`)` `barCopt(`*string*`)` `lineTopt(`*string*`)`
`lineCopt(`*string*`)`]

where *varlist* specifies the vector $\mathbf{w}_i$.

6.2 Options

`outcome(`*varname*`)` specifies the dependent variable of interest. This option is used just to mark the sample on which the *p* score is estimated. `outcome()` is required.

`assign(`*varname*`)` sets the assignment to treatment variable. `assign()` is required.

`score(`*varname*`)` specifies the running variable. `score()` is required.

`bandwidth(`#`)` specifies the value for the bandwidth to be used for estimation. The user can specify a different bandwidth for each side. `bandwidth()` is required.

`cutoff(`#`)` specifies the value of the cutoff. The default is `cutoff(0)`. The cutoff value is subtracted from the `score()` variable and the bandwidth. When multiple cutoffs are present, provide the pooled cutoff.

`nbins(`*numlist*`)` specifies the number of bins in which the average of residuals should be computed. The number of bins can be specified for each side of the cutoff. The default is `nbins(10 10)`.

`site(`*varname*`)` specifies the variable identifying the site to add site fixed effects.

`asis` forces retention of perfect predictor variables and their associated perfectly predicted observations.

`gphoptions(`*string*`)` specifies graphical options to be passed on to the underlying `graph` command. These options overwrite the default formatting options of the command.

`pscore(`*string*`)` specifies the name of the variable containing the *p* score rather than the default logit model. This variable is added to the current dataset.

`probit` implements a probit model to estimate the *p* score.

`kdensity` displays kernel densities rather than histograms, which is the default.

`nograph` suppresses any graphical output.

`legendopt(`*string*`)` specifies graphical options to be passed on to the underlying plot legend.

`barTopt(`*string*`)` specifies graphical options to be passed on to the underlying bar chart for the treated units.

`barCopt(`*string*`)` specifies graphical options to be passed on to the underlying bar chart for the control units.

`lineTopt(`*string*`)` specifies graphical options to be passed on to the underlying density line for the treated units.

`lineCopt(`*string*`)` specifies graphical options to be passed on to the underlying density line for the control units.

7 The getaway command

This section describes the syntax of the `getaway` command, which allows the user to estimate and plot treatment effects away from the cutoff. The command implements either the linear reweighting estimator (5) or the propensity-score reweighting estimator (6). By default, it estimates ATT and ATNT, but it also allows for estimation of other causal parameters of interest on finer intervals of the running variable. Indeed, the command allows to partition the support of the running variable in quantile-spaced bins and to estimate treatment effects within these bins. Obtaining these estimates following (5) is straightforward.

7.1 Syntax

`getaway` *varlist* [*if*] [*in*], `outcome(`*varname*`)` `score(`*varname*`)` `bandwidth(`*#*`)`
[`cutoff(`*#*`)` `method(`*string*`)` `site(`*varname*`)` `nquant(`*numlist*`)` `probit`
`trimming(`*numlist*`)` `bootrep(`*#*`)` `clevel(`*#*`)` `reghd` `qtleplot` `genvar(`*string*`)`
`asis` `gphoptions(`*string*`)` `qtleplotopt(`*string*`)` `qtleciplotopt(`*string*`)`
`attplotopt(`*string*`)` `attciplotopt(`*string*`)` `atntplotopt(`*string*`)`
`atntciplotopt(`*string*`)`]

where *varlist* specifies the vector $\mathbf{w}_i$.

7.2 Options

`outcome(`*varname*`)` specifies the dependent variable of interest. `outcome()` is required.

`score(`*varname*`)` specifies the running variable. `score()` is required.

`bandwidth(`*#*`)` specifies the value for the bandwidth to be used for estimation. The user can specify a different bandwidth for each side. `bandwidth()` is required.

`cutoff(`*#*`)` specifies the value of the cutoff. The default is `cutoff(0)`. The cutoff value is subtracted from the `score()` variable and the bandwidth. When multiple cutoffs are present, provide the pooled cutoff.

`method(`*string*`)` allows to choose the estimation method between the linear reweighting estimator (`linear`) and propensity-score weighting estimator (`pscore`). The default is `method(linear)`.

`site(`*varname*`)` specifies the variable identifying the site to add site fixed effects.

`nquant(`*numlist*`)` specifies the number of quantiles in which the treatment effect must be estimated. It can be specified separately for each side. The default is `nquant(0 0)`. It should be specified if `qtleplot` is used.

`probit` uses a probit model to estimate the *p* score rather than the default logit model. It is effective only if `method("pscore")` is used.

`trimming(`*numlist*`)` specifies a lower and an upper bound for the *p* score. Units with a *p* score outside such intervals are trimmed and not used in estimation and inference. It is effective only if `method("pscore")` is used, and in such a case, the default is `trimming(0.1 0.9)`, according to Crump et al. (2009).

`bootrep(`*#*`)` sets the number of replications of the nonparametric bootstrap. The default is `bootrep(0)`. If `site()` is specified, a nonparametric block bootstrap is used.

`clevel(`*#*`)` specifies the confidence level for the confidence intervals reported in the plot. The default is `clevel(95)`.

`reghd` allows site fixed effects to differ on each side of the cutoff. If the number of observations per ranking is not sufficiently high, it might yield inconsistent estimates for the treatment effects away from the cutoff. It relies on the `reghdfe` command (Correia 2014).

`qtleplot` plots estimated treatment effects over running variable quantiles together with bootstrapped standard errors. It also estimates and reports bootstrapped standard errors of the ATT and ATNT.

`genvar(`*string*`)` specifies the name of the variable containing the distribution of treatment effects. It is used only with `method(linear)`.

`asis` forces retention of perfect predictor variables and their associated perfectly predicted observations in *p*-score estimation. It is to be used only with `method(pscore)`.

`gphoptions(`*string*`)` specifies graphical options to be passed on to the underlying `graph` command.

`qtleplotopt(`*string*`)` specifies graphical options to be passed on to the underlying scatterplot.

`qtleciplotopt(`*string*`)` specifies graphical options to be passed on to the underlying spike plot for the confidence intervals for the treatment effect at each quantile of the running variable.

`attplotopt(`*string*`)` specifies graphical options to be passed on to the underlying line plot for the average treatment effect on the treated.

`attciplotopt(`*string*`)` specifies graphical options to be passed on to the underlying line plot for the confidence interval of the average treatment effect on the treated.

`atntplotopt(`*string*`)` specifies graphical options to be passed on to the underlying line plot for the average treatment effect on the nontreated.

`atntciplotopt(`*string*`)` specifies graphical options to be passed on to the underlying line plot for the confidence interval of the average treatment effect on the nontreated.

8 The getawayplot command

This section explains the syntax of the `getawayplot` command, which plots nonparametric estimates of the actual and counterfactual regression functions using kernel-weighted local polynomial smoothers.

8.1 Syntax

`getawayplot` *varlist* [*if*] [*in*], `outcome(`*varname*`)` `score(`*varname*`)` `bandwidth(`#`)` [`cutoff(`#`)` `kernel(`*string*`)` `site(`*varname*`)` `degree(`#`)` `nbins(`*numlist*`)` `clevel(`#`)` `nostderr` `gphoptions(`*string*`)` `scatterplotopt(`*string*`)` `areaplotopt(`*string*`)` `lineplotopt(`*string*`)` `lineCFplotopt(`*string*`)` `legendopt(`*string*`)`]

where *varlist* specifies the vector $\mathbf{w}_i$.

8.2 Options

`outcome(`*varname*`)` specifies the dependent variable of interest. `outcome()` is required.

`score(`*varname*`)` specifies the running variable. `score()` is required.

`bandwidth(`#`)` specifies the value for the bandwidth to be used for estimation. The user can specify a different bandwidth for each side. `bandwidth()` is required.

`cutoff(`#`)` specifies the value of the cutoff. The default is `cutoff(0)`. The cutoff value is subtracted from the `score()` variable and the bandwidth. When multiple cutoffs are present, provide the pooled cutoff.

`kernel(`*string*`)` specifies the kernel function. The default is `kernel(epanechnikov)`. To see the full list of available kernel functions, see [R] **lpoly**.

`site(`*varname*`)` specifies the variable identifying the site to add site fixed effects.

`degree(`#`)` specifies the degree of the local polynomial smooth. The default is `degree(0)`.

`nbins(`*numlist*`)` specifies the number of bins for which the counterfactual average is shown in the final graph. The default is `nbins(10 10)`.

`clevel(`#`)` specifies the confidence level for the confidence bands reported in the plot. The default is `clevel(95)`.

`nostderr` specifies that standard errors not be computed and plotted.

`gphoptions(`*string*`)` specifies graphical options to be passed on to the underlying graph command.

`scatterplotopt(`*string*`)` specifies graphical options to be passed on to the underlying scatterplot.

`areaplotopt(`*string*`)` specifies graphical options to be passed on to the underlying confidence bands plot.

`lineplotopt(`*string*`)` specifies graphical options to be passed on to the underlying line plot for observed potential outcomes.

`lineCFplotopt(`*string*`)` specifies graphical options to be passed on to the underlying line plot for counterfactual potential outcomes.

`legendopt(`*string*`)` specifies graphical options to be passed on to the underlying plot legend.

9 Illustration of methods

This section illustrates the main features of the `getaway` package using a simulated dataset, `simulated_getaway.dta`. The dataset contains $n = 2000$ observations, which are divided in $S = 5$ different groups (sites) of equal size. In this dataset, `Y` is the outcome variable, `T` is the treatment dummy, `X` is the standardized running variable, `cutoff` is a variable containing the corresponding cutoff for each unit, `site` is a variable containing the site identifier for each unit, and `w1`–`w10` are 10 covariates (henceforth, $\omega_i, i = 1, \ldots, 10$) to be used to validate the CIA assumption. In each site, the cutoff is defined endogenously as $c_s = \text{median}(X_{is})$.

```
. use simulated_getaway
. summarize Y T X cutoff
```

Variable	Obs	Mean	Std. dev.	Min	Max
Y	2,000	100.5602	54.3153	-15.2992	309.6667
T	2,000	.5	.500125	0	1
X	2,000	.0463897	2.058906	-6.855689	6.944369
cutoff	2,000	.9372802	.0830187	.81286	1.039276

```
. tabulate cutoff
```

Site Cutoff	Freq.	Percent	Cum.
.81286	400	20.00	20.00
.8890854	400	20.00	40.00
.9293263	400	20.00	60.00
1.015854	400	20.00	80.00
1.039276	400	20.00	100.00
Total	2,000	100.00	

```
. twoway (scatter Y X if site == 1, msymbol(o))
> (scatter Y X if site == 2, msymbol(o))
> (scatter Y X if site == 3, msymbol(o))
> (scatter Y X if site == 4, msymbol(o))
> (scatter Y X if site == 5, msymbol(o)),
> xline(0) ylabel(,nogrid)
> xtitle("Score") ytitle("Outcome") xlabel(-6(3)6) legend(off)
```

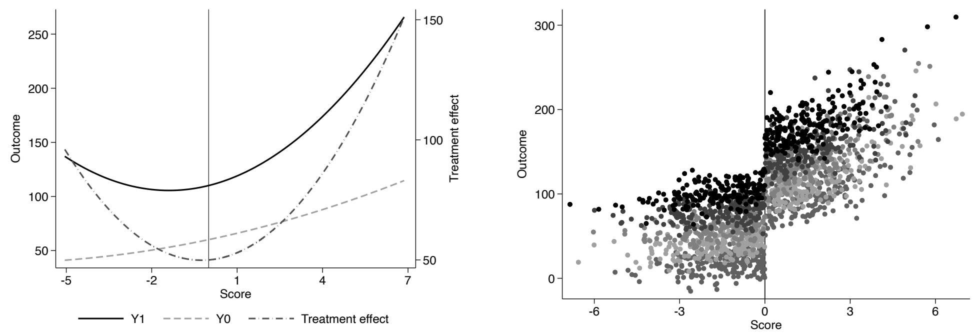
Each covariate is generated according to $\omega_i \overset{\text{i.i.d.}}{\sim} N(\mu_i, 1)$, where each μ_i has been extracted from a $U(-1, 1)$. The running variable is created as

$$X_{is} = \omega_{1,is}\phi_1 + \omega_{2,is}\phi_2 + \nu_{is}, \quad \nu_{is} \overset{\text{i.i.d.}}{\sim} N(0, 1), \;\; j = 1, 2$$

where ϕ_1, ϕ_2 are extracted from a $U(1, 2)$. Let $\omega_{is} := (\omega_{1,is}, \omega_{2,is})'$. The outcome variable is simulated according to the following data-generating process,

$$Y_{is} = \alpha_s + T_{is}\beta + \omega'_{is}\gamma + \omega'_{is}\omega_{is}\delta + T_{is}\omega'_{is}\lambda + T_{is}\omega'_{is}\omega_{is}\rho + \varepsilon_{is}$$

where $\varepsilon_{is} \overset{\text{i.i.d.}}{\sim} N(0, 10), \beta = 50, \gamma = (5, 5)', \delta = 0.5, \lambda = (1, 1)', \rho = 2$. Finally, the fixed effect is $\alpha_s = 20s$, $s = 1, \ldots, 5$. Figure 1(a) shows the (averaged across groups) potential outcomes as a function of the running variable. The figure shows that the data-generating process features heterogeneous treatment effects with respect to the score (gray dash-dotted line). Figure 1(b) plots the observed outcome variable for each site.



(a) Potential outcomes and treatment effect

(b) Observed data

Figure 1. Simulation results

The data-generating process considered in this example implies that

$$\mathbf{w}_{is} := (\omega_{1,is}, \omega_{2,is}, \omega^2_{1,is}, \omega^2_{2,is}, \omega_{1,is}, \omega_{2,is})$$

is the vector of covariates that satisfies the CIA condition (2).

The command `ciasearch` allows to search among the 10 "candidate" covariates `w1`–`w10` the ones (if any) satisfying the CIA. The command, through the option `quad`, also tests interactions and quadratic terms among the candidate covariates. The basic syntax is illustrated in the following snippet.[9]

9. In this case, I choose the bandwidth to be $h = 7$ just to include the whole sample.

```
. ciasearch w1 w2 w3 w4 w5 w6 w7 w8 w9 w10, outcome(Y) score(X) bandwidth(7)
> cutoff(0) site(site) quad noprint poly(2) alpha(0.5)
────────────────────────────────────────────────────────────────────────────────
Algorithm Path:
 Searching for a set of covariates validating the CIA on the left of the cutoff
> ...
Iteration #1 finished || Loss Function (>.5): 0.000 || Selected w2
Iteration #2 finished || Loss Function (>.5): 0.211 || Selected w1
Iteration #3 finished || Loss Function (>.5): 0.718 || Selected w2_sq
 Searching for a set of covariates validating the CIA on the right of the cutoff
> ...
Iteration #1 finished || Loss Function (>.5): 0.000 || Selected w2_sq
Iteration #2 finished || Loss Function (>.5): 0.000 || Selected w2Xw1
Iteration #3 finished || Loss Function (>.5): 0.051 || Selected w1
Iteration #4 finished || Loss Function (>.5): 0.127 || Selected w2
Iteration #5 finished || Loss Function (>.5): 0.840 || Selected w1_sq
────────────────────────────────────────────────────────────────────────────────

Results
Algorithm Converged - Selected Covariates on the Left: w2 w1 w2_sq
Algorithm Converged - Selected Covariates on the Right: w2_sq w2Xw1 w1 w2 w1_sq
```

By default, the algorithm searches for a set of covariates satisfying the CIA separately on each side of the cutoff. The command displays the result of each iteration, reporting the loss function value (in this case, minus the p-value of the test described in step 2 of the algorithm in section 3) and the selected covariate, that is, the one minimizing the loss function. In the particular case reported in the snippet, two different sets of covariates are selected on the two sides.[10] For circumstances like this one, the suggested rule of thumb is to test whether the CIA holds on both sides using the union of the two sets and eventually proceed further with the analysis using this joint set. The results of such a test are presented in the next snippet.

```
. generate w1sq = w1^2
. generate w2sq = w2^2
. generate w2Xw1 = w2*w1
. ciatest w1 w2 w1sq w2sq w2Xw1, outcome(Y) score(X) bandwidth(7) cutoff(0)
> poly(2) site(site) alpha(0.5)
──────────────────────────────────────────────────────────────────────────────
             CIA Test Results
               LEFT       RIGHT
 Coef_1  -.50775874  -.15850905
 Coef_2   -.1296132  -.01281982
 F-stat   .22724811    .1741112
p-value   .79676472   .84022923
      N        1000        1000
CIA condition satisfied! (alpha = .5)
──────────────────────────────────────────────────────────────────────────────
```

The output above shows the main statistics obtained from running a regression similar to (7) on each side of the cutoff and testing the CIA. The left column reports the results obtained to the left, and the right column the results to the right. The option

10. The chosen set of covariates on the left is a subvector of $\mathbf{w}_{is}$. This is because the algorithm selects the first vector that satisfies the stopping rule, which is reaching a p-value of at least 0.5 (with the `alpha(0.5)` option).

`poly(2)` fits a polynomial of second order in the running variable, and the first two rows display the regression coefficients corresponding to X_{is} and X_{is}^2. The third and fourth row summarize the hypothesis tests by showing the F statistic and the corresponding p-value. The last row reports the number of observations used.

Graphical evidence can support the statistical evidence obtained with `ciatest`. The `ciares` command allows the user to test graphically an implication of the CIA, according to the following procedure:

- Run the regressions of the outcome variable on the vector of covariates $\mathbf{w}_{is}$ to the left and to the right of the cutoff; that is,

$$Y_{is} = \alpha_{L,s} + \mathbf{w}_{is}'\phi_L + \varepsilon_{is}, \quad \text{if} \quad -X_{is} < c_s$$
$$Y_{is} = \alpha_{R,s} + \mathbf{w}_{is}'\phi_R + \nu_{is}, \quad \text{if} \quad X_{is} \geq c_s$$

  where $\alpha_{j,s}$, $j = L, R$ are site fixed effects that may be included.

- Store the residuals of these regressions; namely,

$$\hat{e}_{is}^L := \left(Y_{is} - \hat{\alpha}_{L,s} - \mathbf{w}_{is}'\hat{\phi}_L\right), \quad \hat{e}_{is}^R := \left(Y_{is} - \hat{\alpha}_{R,s} - \mathbf{w}_{is}'\hat{\phi}_R\right)$$

- Plot $\hat{e}_{is}^L$ and $\hat{e}_{is}^R$ on the running variable X_{is}.

The implication being heuristically tested here is that once the variation in $\mathbf{w}_{is}$ is accounted for, if the CIA is satisfied, the running should have no explanatory power on the outcome variable. Therefore, the residuals $(\hat{e}_i^L, \hat{e}_i^R)$ should be orthogonal to the running variable X_i. This means that if the CIA is satisfied, plotting the residuals over the score should ideally yield a horizontal line. Because of sampling variation, this never happens in practice. Hence, to correctly visualize this relationship and to sweep away sampling error, the command partitions the running variable in equally spaced bins and computes within-bin averages of the residuals. As figure 2 shows, `ciares` displays within-bin averages together with the linear regressions of the residuals on the running variable. The option `cmpr(1 1)` also plots the simple linear regression of Y_{is} on X_{is} for comparison.

```
. ciares w1 w2 w1sq w2sq w2Xw1, outcome(Y) score(X) bandwidth(7)
> site(site) nbins(10 10)
> gphoptions(xlabel(-6(3)6, nogrid) ylabel(,nogrid) title("")) cmpr(1 1)
```

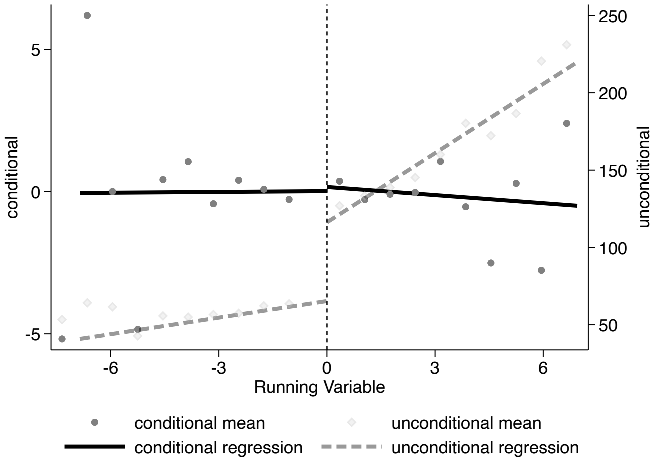


Figure 2. Visualization of the CIA

Up to this point, just the first of the two conditions underlying the CIA has been tested and validated. However, the common support condition (3) also needs to be checked. The command `ciacs` serves this precise purpose.

```
. ciacs w1 w2 w1sq w2sq w2Xw1, outcome(Y) score(X) bandwidth(7) assign(T)
> cutoff(0) site(site) pscore(pscore) gphoptions(title(""))
```

		Common Support		
	N	Out of CS	Lower Bound	Upper Bound
Control	1000	42	.00060943	.98419487
Treatment	1000	280	.00496089	.99999964

```
The common support is verified in the interval [0.0050,0.9842],
which contains 958 control units and 720 treated units.
```

The command estimates the propensity score for all units and computes the bounds for the common support as

$$\text{lb} := \max\left\{\min_{i:T_{is}=0} \widehat{p}_{is}, \min_{i:T_{is}=1} \widehat{p}_{is}\right\} \quad \text{and} \quad \text{ub} := \min\left\{\max_{i:T_{is}=0} \widehat{p}_{is}, \max_{i:T_{is}=1} \widehat{p}_{is}\right\}$$

where $\widehat{p}_{is}$ is the estimated propensity score. Using these bounds, the command identifies 42 control units and 280 treatment units that are outside the common support region. Those units should not be considered in the analysis and should be flagged. This can be easily done in two steps. First, specify the option `pscore()`, which generates a new variable containing the estimated propensity score. Second, use the bounds of the common support stored in `return list` to mark units not in the common support.

```
. generate incs = pscore >= e(CSmin) & pscore <= e(CSmax)
. tabulate incs T
```

incs	Treatment Dummy 0	1	Total
0	42	280	322
1	958	720	1,678
Total	1,000	1,000	2,000

In addition, the command `ciacs` allows the user to visualize the common support using either histograms or kernel density estimators. Figure 3 shows the result obtained using histograms.

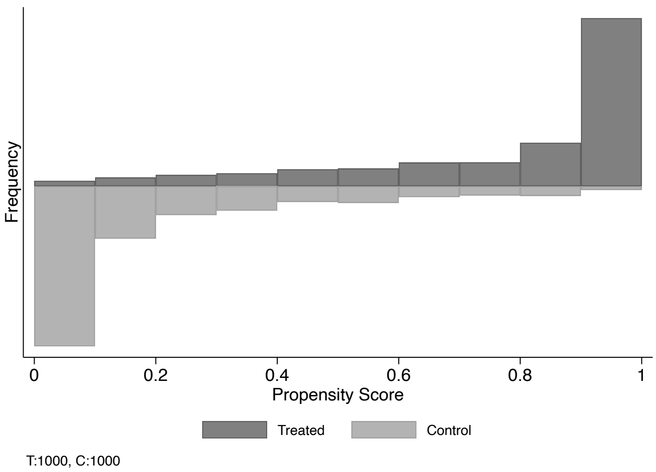


Figure 3. Visualization of common support condition

Now that both condition (2) and condition (3) have been verified, the commands `getaway` and `getawayplot` can be used to extrapolate treatment effects and estimate potential outcomes along the support of the running variable, respectively.

```
. getaway w1 w2 w1sq w2sq w2Xw1, outcome(Y) score(X) bandwidth(7)
> cutoff(0) site(site) qtleplot nquant(5 5) boot(100)
> gphoptions(title("") ylabel(,nogrid)) genvar(effect_est)
Bootstrapping standard errors ... (100)
----+--- 1 ---+--- 2 ---+--- 3 ---+--- 4 ---+--- 5
..................................................    50
..................................................   100
--------------------------------------------------------------------------------
             Extrapolation Results
--------------------------------------------------------------------------------
Outcome Variable            Y
Running Variable            X
Number of observations
              Treated       1000
              Control       1000
Cutoff                      0
Bandwidth                   7
Bootstrap Iterations        100
Site Fixed Effects          site
Method                      linear
Main Estimates

      Estimate        SE
ATNT    54.094     0.414
 ATT    64.665     1.105
Within-Quantile Estimates

          Estimate        SE       Xlb       Xub
 Left_1     58.736     1.098    -6.856    -2.589
 Left_2     53.448     0.599    -2.584    -1.726
 Left_3     52.461     0.368    -1.726    -1.031
 Left_4     52.561     0.219    -1.030    -0.533
 Left_5     53.263     0.204    -0.529    -0.001
Right_1     55.723     0.224     0.001     0.532
Right_2     58.078     0.779     0.532     1.099
Right_3     60.577     1.048     1.103     1.789
Right_4     65.985     0.934     1.793     2.658
Right_5     82.963     2.833     2.666     6.944
CIA Covariates: w1 w2 w1sq w2sq w2Xw1
--------------------------------------------------------------------------------
```

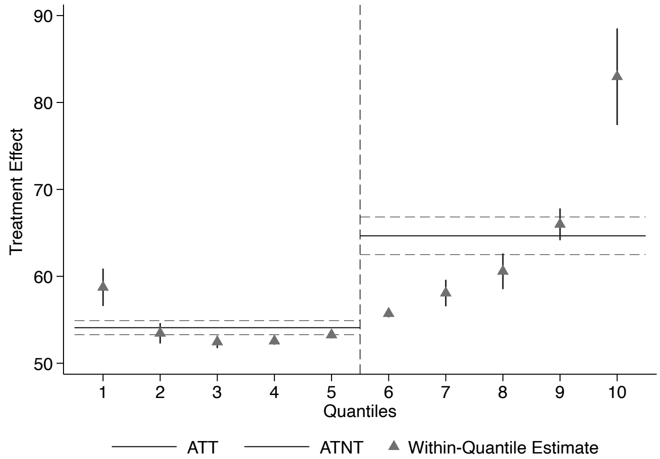


Figure 4. Estimated treatment effects over the support of the running variable

The command `getaway` prints the main specifics of the estimation procedure (outcome variable, running variable, number of observations, cutoff, bandwidth, and so forth) and two tables with the results. The first table contains the estimates for ATNT $\tau_{(-h,0)}$ and ATT $\tau_{[0,h)}$, together with bootstrapped standard errors. The second table reports the treatment effect within quantiles of the running variable (first column); bootstrapped standard errors (second column); and lower and upper bounds of the running variable in each quantile (third and fourth columns). The running variable is binned in 5 quantiles per side through the option `nquant(5 5)`. All the information contained in these two tables can also be visualized in a single graph by specifying the option `qtleplot`. Figure 4 shows the estimated treatment effect in each quantile of the running variable (gray triangles), together with bootstrapped standard errors (vertical black solid lines) and the estimates of $\tau_{[0,h)}$ (gray solid line to the right of the cutoff) and $\tau_{(-h,0)}$ (gray solid line to the left of the cutoff) with their bootstrapped standard errors (dashed horizontal lines). Estimated treatment effects recover the asymmetric U-shaped true treatment-effect function presented in figure 1(a). Moreover, the option `genvar(effect_est)` creates a new variable called `effect_est` that contains the treatment effect for each unit estimated according to either (5) (default) or (6).

Finally, the command `getawayplot` uses kernel-weighted local polynomial smoothers to recover the potential outcomes as functions of the running variable. Figure 5 displays the estimated potential outcomes as functions of the running variable (gray dotted lines) and reports 95% confidence bands by default. Estimates are obtained fitting kernel-weighted local polynomials to the predicted value of a regression of the outcome on $\mathbf{w}_{is}$ run separately on each side of the cutoff. Finally, within-bin averages are also reported (gray crosses).

```
. getawayplot w1 w2 w1sq w2sq w2Xw1, outcome(Y) score(X) bandwidth(7)
> cutoff(0) kernel(triangle) degree(2) nbins(30) site(site)
> gphoptions(xlabel(-6(3)6, nogrid) ylabel(,nogrid))
```

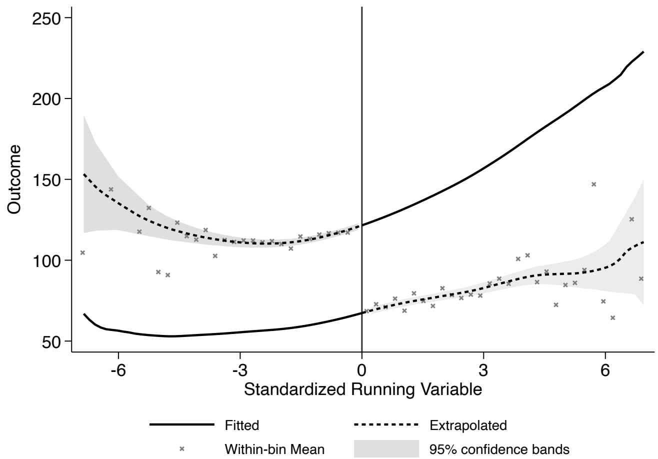


Figure 5. Estimate potential outcomes over the support of the running variable

Comparing figure 5 with figure 1(a), one can graphically compare the goodness of the estimation procedure.

10 Conclusion

This article introduced the `getaway` package, which estimates treatment effects away from the cutoff in RD designs under a CIA.

Further work on the `getaway` package is planned. First, the actual package works only for sharp RD designs; hence, its extension to fuzzy RD designs is necessary. Second, there are several other RD designs in the literature in which extrapolation of treatment effects might turn out to be useful. A nonexhaustive list of examples includes geographic RDs (Keele and Titiunik 2015), dynamic RDs (Cellini, Ferreira, and Rothstein 2010), and difference in discontinuities (Grembi, Nannicini, and Troiano 2016). Third, at the date of publication of this article, the package `getaway` estimates the conditional expectation functions of the potential outcomes using two different parametric alternatives. Next major upgrades of the package should also contain nonparametric estimators (for example, Nadaraya–Watson kernel regression and random forest for such functions).

11 Acknowledgments

I am obliged to Thomas Bearpark, Federico Cingano, Chiara Motta, Paolo Pinotti, and Enrico Rettore for all the useful conversations and suggestions. I thank Joshua Angrist and Miikka Rokkanen for making their code publicly available. All errors remain my own.

12 Programs and supplemental material

To install the software files as they existed at the time of publication of this article, type

```
. net sj 24-3
. net install st0751      (to install program files, if available)
. net get st0751          (to install ancillary files, if available)
```

The latest software version can be found at https://github.com/filippopalomba/getaway-package.

13 References

Angrist, J. D., and V. Lavy. 1999. Using Maimonides' rule to estimate the effect of class size on scholastic achievement. *Quarterly Journal of Economics* 114: 533–575. https://doi.org/10.1162/003355399556061.

Angrist, J. D., and M. Rokkanen. 2015. Wanna get away? Regression discontinuity estimation of exam school effects away from the cutoff. *Journal of the American Statistical Association* 110: 1331–1344. https://doi.org/10.1080/01621459.2015.1012259.

Armstrong, T. B., and M. Kolesár. 2018. Optimal inference in a class of regression models. *Econometrica* 86: 655–683. https://doi.org/10.3982/ECTA14434.

Bartelsman, E., S. Scarpetta, and F. Schivardi. 2005. Comparative analysis of firm demographics and survival: Evidence from micro-level sources in OECD countries. *Industrial and Corporate Change* 14: 365–391. https://doi.org/10.1093/icc/dth057.

Battistin, E., A. Brugiavini, E. Rettore, and G. Weber. 2009. The retirement consumption puzzle: Evidence from a regression discontinuity approach. *American Economic Review* 99: 2209–2226. https://doi.org/10.1257/aer.99.5.2209.

Battistin, E., and E. Rettore. 2008. Ineligibles and eligible non-participants as a double comparison group in regression-discontinuity designs. *Journal of Econometrics* 142: 715–730. https://doi.org/10.1016/j.jeconom.2007.05.006.

Bertanha, M. 2020. Regression discontinuity design with many thresholds. *Journal of Econometrics* 218: 216–241. https://doi.org/10.1016/j.jeconom.2019.09.010.

Bertanha, M., and G. W. Imbens. 2020. External validity in fuzzy regression discontinuity designs. *Journal of Business and Economic Statistics* 38: 593–612. https://doi.org/10.1080/07350015.2018.1546590.

Bronzini, R., and E. Iachini. 2014. Are incentives for R&D effective? Evidence from a regression discontinuity approach. *American Economic Journal: Economic Policy* 6(4): 100–134. https://doi.org/10.1257/pol.6.4.100.

Calonico, S., M. D. Cattaneo, and R. Titiunik. 2014. Robust data-driven inference in the regression-discontinuity design. *Stata Journal* 14: 909–946. https://doi.org/10.1177/1536867X1401400413.

Cattaneo, M. D., B. R. Frandsen, and R. Titiunik. 2015. Randomization inference in the regression discontinuity design: An application to party advantages in the U.S. Senate. *Journal of Causal Inference* 3: 1–24. https://doi.org/10.1515/jci-2013-0010.

Cattaneo, M. D., N. Idrobo, and R. Titiunik. 2024. *A Practical Introduction to Regression Discontinuity Designs: Extensions*. Cambridge: Cambridge University Press. https://doi.org/10.1017/9781009441896.

Cattaneo, M. D., L. Keele, R. Titiunik, and G. Vazquez-Bare. 2021. Extrapolating treatment effects in multi-cutoff regression discontinuity designs. *Journal of the American Statistical Association* 116: 1941–1952. https://doi.org/10.1080/01621459.2020.1751646.

Cattaneo, M. D., R. Titiunik, and G. Vazquez-Bare. 2020. Analysis of regression-discontinuity designs with multiple cutoffs or multiple scores. *Stata Journal* 20: 866–891. https://doi.org/10.1177/1536867X20976320.

Cellini, S. R., F. Ferreira, and J. Rothstein. 2010. The value of school facility investments: Evidence from a dynamic regression discontinuity design. *Quarterly Journal of Economics* 125: 215–261. https://doi.org/10.1162/qjec.2010.125.1.215.

Correia, S. 2014. reghdfe: Stata module to perform linear or instrumental-variable regression absorbing any number of high-dimensional fixed effects. Statistical Software Components S457874, Department of Economics, Boston College. https://ideas.repec.org/c/boc/bocode/s457874.html.

Coviello, D., and M. Mariniello. 2014. Publicity requirements in public procurement: Evidence from a regression discontinuity design. *Journal of Public Economics* 109: 76–100. https://doi.org/10.1016/j.jpubeco.2013.10.008.

Crump, R. K., V. J. Hotz, G. W. Imbens, and O. A. Mitnik. 2009. Dealing with limited overlap in estimation of average treatment effects. *Biometrika* 96: 187–199. https://doi.org/10.1093/biomet/asn055.

Dong, Y., and A. Lewbel. 2015. Identifying the effect of changing the policy threshold in regression discontinuity models. *Review of Economics and Statistics* 97: 1081–1092. https://doi.org/10.1162/REST_a_00510.

Duflo, E., P. Dupas, and M. Kremer. 2011. Peer effects, teacher incentives, and the impact of tracking: Evidence from a randomized evaluation in Kenya. *American Economic Review* 101: 1739–1774. https://doi.org/10.1257/aer.101.5.1739.

Flammer, C. 2015. Does corporate social responsibility lead to superior financial performance? A regression discontinuity approach. *Management Science* 61: 2549–2568. https://doi.org/10.1287/mnsc.2014.2038.

Fort, M., A. Ichino, E. Rettore, and G. Zanella. 2022. Multi-cutoff rd designs with observations located at each cutoff: Problems and solutions. Discussion Paper DP16974, Center for Economic Policy Research. https://ssrn.com/abstract=4026880.

Goldberger, A. S. 2008. Selection bias in evaluating treatment effects: Some formal illustrations. In *Advances in Econometrics.* Vol. 21, *Modelling and Evaluating Treatment Effects in Econometrics*, ed. T. Fomby, R. Carter Hill, D. L. Millimet, J. A. Smith, and E. J. Vytlacil, 1–31. Leeds, UK: Emerald. https://doi.org/10.1016/S0731-9053(07)00001-1.

Grembi, V., T. Nannicini, and U. Troiano. 2016. Do fiscal rules matter? *American Economic Journal: Applied Economics* 8(3): 1–30. https://doi.org/10.1257/app.20150076.

Hahn, J., P. Todd, and W. van der Klaauw. 2001. Identification and estimation of treatment effects with a regression-discontinuity design. *Econometrica* 69: 201–209. https://doi.org/10.1111/1468-0262.00183.

Heckman, J. J., R. J. LaLonde, and J. A. Smith. 1999. The economics and econometrics of active labor market programs. In Vol. 3A of *Handbook of Labor Economics*, ed. O. Ashenfelter and D. Card, 1865–2097. Amsterdam: Elsevier. https://doi.org/10.1016/S1573-4463(99)03012-6.

Hirano, K., G. W. Imbens, and G. Ridder. 2003. Efficient estimation of average treatment effects using the estimated propensity score. *Econometrica* 71: 1161–1189. https://doi.org/10.1111/1468-0262.00442.

Imbens, G. W., and K. Kalyanaraman. 2012. Optimal bandwidth choice for the regression discontinuity estimator. *Review of Economic Studies* 79: 933–959. https://doi.org/10.1093/restud/rdr043.

Imbens, G. W., and D. B. Rubin. 2015. *Causal Inference for Statistics, Social, and Biomedical Sciences: An Introduction.* New York: Cambridge University Press. https://doi.org/10.1017/CBO9781139025751.

James, G., D. Witten, T. Hastie, R. Tibshirani, and J. Taylor. 2023. *An Introduction to Statistical Learning: With Applications in Python.* Cham, Switzerland: Springer. https://doi.org/10.1007/978-3-031-38747-0.

Jovanovic, B. 1982. Selection and the evolution of industry. *Econometrica* 50: 649–670. https://doi.org/10.2307/1912606.

Keele, L. J., and R. Titiunik. 2015. Geographic boundaries as regression discontinuities. *Political Analysis* 23: 127–155. https://doi.org/10.1093/pan/mpu014.

Kline, P. 2011. Oaxaca–Blinder as a reweighting estimator. *American Economic Review* 101: 532–537. https://doi.org/10.1257/aer.101.3.532.

Lalive, R. 2008. How do extended benefits affect unemployment duration? A regression discontinuity approach. *Journal of Econometrics* 142: 785–806. https://doi.org/10.1016/j.jeconom.2007.05.013.

Lee, D. S. 2008. Randomized experiments from non-random selection in U.S. House elections. *Journal of Econometrics* 142: 675–697. https://doi.org/10.1016/j.jeconom.2007.05.004.

Lee, D. S., E. Moretti, and M. J. Butler. 2004. Do voters affect or elect policies? Evidence from the U. S. House. *Quarterly Journal of Economics* 119: 807–859. https://doi.org/10.1162/0033553041502153.

Ludwig, J., and D. L. Miller. 2007. Does Head Start improve children's life chances? Evidence from a regression discontinuity design. *Quarterly Journal of Economics* 122: 159–208. https://doi.org/10.1162/qjec.122.1.159.

Meyersson, E. 2014. Islamic rule and the empowerment of the poor and pious. *Econometrica* 82: 229–269. https://doi.org/10.3982/ECTA9878.

Ozier, O. 2018. The impact of secondary schooling in Kenya: A regression discontinuity analysis. *Journal of Human Resources* 53: 157–188. https://doi.org/10.3368/jhr.53.1.0915-7407R.

Pettersson-Lidbom, P. 2008. Do parties matter for economic outcomes? A regression-discontinuity approach. *Journal of the European Economic Association* 6: 1037–1056. https://doi.org/10.1162/JEEA.2008.6.5.1037.

Pinotti, P. 2017. Clicking on heaven's door: The effect of immigrant legalization on crime. *American Economic Review* 107: 138–168. https://doi.org/10.1257/aer.20150355.

Pop-Eleches, C., and M. Urquiola. 2013. Going to a better school: Effects and behavioral responses. *American Economic Review* 103: 1289–1324. https://doi.org/10.1257/aer.103.4.1289.

Rubin, D. B. 1974. Estimating causal effects of treatments in randomized and nonrandomized studies. *Journal of Educational Psychology* 66: 688–701. https://doi.org/10.1037/h0037350.

Thistlethwaite, D. L., and D. T. Campbell. 1960. Regression-discontinuity analysis: An alternative to the ex post facto experiment. *Journal of Educational Psychology* 51: 309–317. https://doi.org/10.1037/h0044319.

About the author

Filippo Palomba is a PhD student in economics at Princeton University.