Getting away from the cutoff in Regression Discontinuity Designs

Filippo Palomba*

Princeton University

February 1, 2022

Abstract

Regression Discontinuity (RD) designs are extremely popular in economic research due to their strong internal validity and straightforward intuition. While RD estimates are local in nature, several recent papers propose methods that generalize RD estimates to units outside a small neighborhood of the cutoff. I introduce the Stata `getaway` package, which implements the method proposed by Angrist and Rokkanen (2015) to extrapolate estimates "away from the cutoff", and generalizes such approach to RD designs with multiple cutoffs. In addition, I present a data-driven algorithm that searches for a set of covariates satisfying the assumptions needed to extrapolate treatment effects over the support of the running variable.

Keywords: 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 et al. (2001), the RD design gained extreme popularity in empirical work in several fields of economics: political (Bronzini and Iachini, 2014; Lee et al., 2004; Meyersson, 2014; Pettersson-Lidbom, 2008), development (Baez and Camacho, 2011; Ozier, 2018), health (Ludwig and Miller, 2007), crime (Pinotti, 2017), education (Angrist and Lavy, 1999; Cellini

*Princeton University, Department of Economics. E-mail: `fpalomba@princeton.edu`.

et al., 2010; Duflo et al., 2011; Pop-Eleches and Urquiola, 2013), public (Battistin et al., 2009; Coviello and Mariniello, 2014; Lalive, 2008), and corporate finance (Flammer, 2015).

In the typical design, units are assigned to treatment depending on the value of a covariate (*score* or *running variable*) being below or above a certain threshold (*cutoff*). Hence, the conditional probability of being assigned to the treatment jumps at the cutoff, inducing as-good-as-random variation in treatment status that can be used to identify causal parameters. Intuitively, units whose score value is close to the cutoff ended up laying on different sides by chance, thus belong to different quasi-experimental groups. For sufficiently small differences in score values, it is possible to claim that such units are "as if" they were randomly assigned to treatment and compare their post treatment outcomes. With this comparison it is possible to identify a local average treatment effect (τ^{RD}, henceforth).

However, such RD estimates are local by construction. Indeed, the above mentioned identification assumption holds only in a neighborhood of the cutoff, so that the identified causal parameter refers to a very specific population. Several recent works tried to extend the causal estimate to wider populations. Battistin and Rettore (2008) derive conditions under which τ^{RD} can be interpreted as an average treatment effect on the treated. They exploit one-sided non-compliance in fuzzy RD to test for selection bias comparing control units with treated units that self-select out of the intervention. Dong and Lewbel (2015) show how to identify the gradient of the treatment effect at the cutoff under a local policy invariance assumption. In fuzzy RDs, Bertanha and Imbens (2020) propose a testing procedure to assess presence of selection bias between compliers and non-compliers. If the null of no selection bias is not rejected, one can then extrapolate the average treatment effect for compliers to other subpopulations (at the cutoff). Bertanha (2020) identifies average treatment effects computed over general counterfactual distributions of individuals, rather than over those units in a neighborhood of the cutoff. Bennett (2020) extends the RD causal parameter to a wider neighborhood around the cutoff relying on the predictive power of covariates in such intervals. In particular, the author introduces a data-driven "generalization bandwidth" to identify the largest population the local average treatment effect can refer to. 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). The authors exploit additional information contained in explanatory variables other than the score to estimate treatment effects away from the cutoff. The only assumption needed is 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. Moreover, the CIA has implications that can be tested with standard hypothesis tests.

The setting presented in Angrist and Rokkanen (2015) is explained using a RD design with a single cutoff. In what follows, I show how to extend the results in Angrist and Rokkanen (2015) to the case with more than one cutoff. Generalizing to the multiple cutoffs case requires extra attention. 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), Fort et al. (2022) show that the identification of average causal effects could be problematic. In this case the standard NP estimator may differ from the desired causal parameter -e.g. 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" (SFE) estimation strategy. The SFE 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, I propose a data-driven algorithm in the spirit of Imbens and Rubin (2015) that searches for a vector of covariates that satisfies the CIA and, hence, allows users to extrapolate treatment effects away from the cutoff. The logic of the algorithm is to select at each iteration the covariate that makes the CIA condition more likely to be satisfied. In practice, this boils down to selecting the covariate with the highest p-value (lowest test statistic) when testing the CIA implications in the data.

Finally, this article introduces the `getaway` package which estimates treatment effects away from the cutoff as described in Angrist and Rokkanen (2015) in the more general framework of RD with multiple cutoffs. The `getaway` package contains six different commands: `ciasearch` applies a data-driven algorithm that selects an adequate set of covariates to "get away" from the cutoff; `ciatest` tests the CIA assumption; `ciares` and `ciacs` produce graphical visualizations of the CIA assumption; `getaway` estimates parametrically treatment effects away from the cutoff; `getawayplot` shows 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, 4, 6, 5, 7, and 8 describe the syntax of `ciasearch`, `ciatest`, `ciacs`, `ciares`, `getaway`, and `getawayplot`, respectively. Section 9

gives numerical illustrations, and Section 10 concludes. The latest version of this software can be found at https://github.com/filippopalomba/getaway-package.

2 Overview of Methods

In a 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 a RD design, the assignment to treatment follows a deterministic rule known at least to the researcher, i.e.

$$D_i = \begin{cases} 1, & \text{if } X_i \geq c \\ 0, & \text{if } X_i < c \end{cases} \tag{2.1}$$

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]

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 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, it is possible to define the observed outcome as

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

[1] For simplicity, I assume that assignment to treatment is a non-decreasing 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 et al. (2020), 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.

and the local average treatment effect as

$$\tau^{RD} := \mathbb{E}\left[Y_i(1) - Y_i(0) \mid X_i = c\right]. \tag{2.2}$$

From (2.2), 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, as units belonging to the treatment group have different score values from units in the control group (Cattaneo et al., 2019). To solve this problem, Hahn et al. (2001) propose to compare units laying immediately to the right and to the left of the cutoff. This can be seen as a local random experiment. Thus, these units should be similar in terms of observable and unobservable characteristics, minimizing selection bias.

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, i.e.

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

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

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

In practice, this 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, i.e. those units such that $X_i \in [c - h, c + h]$. This is nothing more than a particular version of the well-known bias-variance trade-off. Indeed, choosing a wider bandwidth (higher 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]

[4]The reader interested in how to find the optimal bandwidth may want to refer to Imbens and Kalyanaraman (2012) or Calonico et al. (2014). 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.

For this reason, despite having strong internal validity, standard RD designs may lack external validity. By construction the causal parameter τ^{RD} is local, as 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 et al., 1999). Indeed, in RD designs omitted variable bias can only stem 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 only rely 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

Angrist and Rokkanen (2015) 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 function of some observed ($\mathbf{w}_i$) and unobserved ($\mathbf{u}_i$) covariates, i.e. $X_i = g(\mathbf{w}_i, \mathbf{u}_i)$. 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 it is possible to use them to move away from the cutoff. Such a condition is termed the "*conditional independence assumption*" and it holds if and only if

$$\mathbb{E}\left[Y_i(j) \mid X_i, \mathbf{w}_i\right] = \mathbb{E}\left[Y_i(j) \mid \mathbf{w}_i\right], \quad j = 0, 1 \tag{2.3}$$

and

$$0 < \Pr[D_i = 1 \mid \mathbf{w}_i] < 1, \text{ a.s.} \tag{2.4}$$

In words, condition (2.3) 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 (2.4) is

a matching-style common support assumption that requires the treatment dummy to be non-degenerate within each cell induced by the vector $\mathbf{w}_i$. The CIA breaks the link between the running variable and the dependent variable, 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 a RD design is the running variable. To fix ideas, suppose a researcher is interested in identifying the treatment effect at $X_i \in A$, where A is a non-empty set contained in the support of the running variable. In this case the estimand of interest is $\tau_A := \mathbb{E}[Y_i(1) - Y_i(0) \mid X_i \in A]$. If (2.3) and (2.4) hold, then τ_A can be estimated as

$$\tau_A = \mathbb{E}[\mathbb{E}[Y_i \mid \mathbf{w}_i, D_i = 1] - \mathbb{E}[Y_i \mid \mathbf{w}_i, D_i = 0] \mid X_i \in A] \tag{2.5}$$

Example. *If a researcher is interested in the Average Treatment on the Treated (ATT), then* $A = [c, \infty)$ *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 Non-Treated (ATNT) is of interest, then $A = (-\infty, c)$ *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 τ_A in (2.5) it is sufficient to rely on the plug-in principle. The crucial part regards the estimation of the inner conditional expectation $\mathbb{E}[Y_i \mid \mathbf{w}_i, D_i]$. Angrist and Rokkanen (2015) 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

$$\mathbb{E}[Y_i \mid \mathbf{w}_i, D_i = 1] = \mathbf{w}_i'\boldsymbol{\beta}_R, \quad \mathbb{E}[Y_i \mid \mathbf{w}_i, D_i = 0] = \mathbf{w}_i'\boldsymbol{\beta}_L$$

   Then, substituting it in (2.5) we get the following linear reweighting estimator (Kline, 2011)

$$\mathbb{E}[Y_i(1) - Y_i(0) \mid X_i \in A] = (\boldsymbol{\beta}_R - \boldsymbol{\beta}_L)'\mathbb{E}[\mathbf{w}_i \mid X_i \in A]. \tag{2.6}$$

2. A propensity score weighting estimator in the spirit of Hirano et al. (2003)

$$\mathbb{E}\left[Y_i(1) - Y_i(0) \mid X_i \in A\right] = \mathbb{E}\left[\frac{Y_i[D_i - p(X_i)]}{p(X_i)[1 - p(X_i)]} \cdot \frac{\Pr[X_i \in A \mid \mathbf{w}_i]}{\Pr[X_i \in A]}\right], \tag{2.7}$$

where $p(X_i)$ is a probability model for the propensity score.

2.2.3 Testing

Conditions (2.3) and (2.4) have (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 dependent variable and the running variable there is no omitted variable bias. Thus, testing the CIA boils down to testing if X_i has a statistically significant effect on Y_i conditionally on $\mathbf{w}_i$. Formally, the validity of the CIA on the interval $[h_L, h_R]$ can be tested empirically by running the following regressions

$$\begin{aligned} Y_i &= \alpha_L + X_i\gamma_{L,1} + \cdots + X_i^p\gamma_{L,p} + \mathbf{w}_i'\boldsymbol{\beta}_L + \varepsilon_i, \quad \text{if} \quad c - h_L \leq X_i < c, \\ Y_i &= \alpha_R + X_i\gamma_{R,1} + \cdots + X_i^q\gamma_{R,q} + \mathbf{w}_i'\boldsymbol{\beta}_R + \nu_i, \quad \text{if} \quad c \leq X_i \leq c + h_R, \end{aligned} \tag{2.8}$$

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 the CIA is satisfied by the vector of covariates $\mathbf{w}_i$ on the interval $[h_L, h_R]$.

2.3 Regression discontinuity with multiple rankings

Another generalization of the RD design is the case of multiple cutoffs. In the recent literature, a popular practice is to normalize all the cutoffs to a common value (zero) and pool together units of 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 and c_s be the

[5]In principle, it is possible to 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 lack statistical power.

cutoff value that determines assignment to treatment in site s. Often, researchers create a normalized version of the score using the formula $\widetilde{X}_i = X_{is} - c_s$. Normalizing-and-pooling allows researchers to rewrite the assignment-to-treatment function as $D_i = \mathbb{1}(\widetilde{X}_i > 0)$. By doing so, a single estimate is obtained using all available data and, hence, increasing the statistical power and precision of the estimates. Cattaneo et al. (2021) outlines 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 threshold is determined ex-ante in each ranking as in the setting described in Cattaneo et al. (2021) (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 from the fact that the score variable has a mass point exactly at the cutoff. As a consequence, 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 most efficient solution to this problem is introducing fixed-effect dummies at the site level. In this spirit, the `getaway` package extends the work done in Angrist and Rokkanen (2015) to the multiple cutoff case introducing the possibility to add site-level fixed effects to the RD estimand.

2.4 Data-driven algorithm

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

- a set $C_1 \subset C$ containing $k_1 < k$ covariates to be included in $\mathbf{w}_i$, but are not sufficient

to make the running variable ignorable. In principle, it could be that $C_1 = \varnothing$.

- a set $C_2 \subseteq C$ containing $k_2 \leq k$ *candidate* covariates to be included in $\mathbf{w}_i$ with the only purpose of making the running variable ignorable.

The algorithm searches for sets $\widetilde{C}_L, \widetilde{C}_R \subseteq C_2$ such that $\widetilde{C}_L \cup C_1$ and $\widetilde{C}_R \cup C_1$ make the running variable ignorable to the left and to the right of the cutoff, respectively. The algorithm is composed by the following steps:

1. let ι be the number of covariates in C_2 already selected.[6] Run the following set of regressions for $j = 1, \ldots, k_2 - \iota$,

$$\begin{aligned} Y_{is} &= \sum_{\ell=1}^{p} X_{is}^{\ell}\gamma_{L,\ell} + \mathbf{z}_{is}'\delta_L + \omega_{is}^{(j)}\mu_{L,j} + \alpha_{L,s} + \varepsilon_{is}, \quad \text{if } c_s - h_L \leq X_{is} < c_s, \\ Y_{is} &= \sum_{\ell=1}^{q} X_{is}^{\ell}\gamma_{R,\ell} + \mathbf{z}_{is}'\delta_R + \omega_{is}^{(j)}\mu_{R,j} + \alpha_{R,s} + \nu_{is}, \quad \text{if } c_s \leq X_{is} \leq c_s + h_R, \end{aligned} \tag{2.9}$$

   where $\mathbf{z}_{is}$ is the vector of k_1 covariates that are always included, h_L and h_R are the bandwidth to the left and to the right of the cutoff, respectively, $\alpha_{\bullet,s}$ are fixed effects at the site level (Fort et al., 2022), and $\omega_{is}^{(j)} \in C_2$ is the j-th candidate covariate. Notice that the RD design with a single cutoff or with multiple exogenous cutoffs are a particular specification of (2.9), in which the site fixed effects $\alpha_{\bullet,s}$ become a pooled constant $\alpha_{\bullet}$.

2. For each regression in (2.9) run the 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 F-tests $F^{j,L}$ and $F^{j,R}$.

3. Select the two covariates associated to 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 $\omega_{L,is}^{\star}$ and $\omega_{R,is}^{\star}$, 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, i.e. it can be that $\omega_{L,is}^{\star} \neq \omega_{R,is}^{\star}$.

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

4. Add $\omega^{\star}_{L,is}$ and $\omega^{\star}_{R,is}$ to $\widetilde{C}_L$ and $\widetilde{C}_R$, respectively, and to the regressions in (2.9).
5. Repeat steps 1-3 for the other candidate covariates.
6. Repeat step 5 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 α% level
    - all the covariates in $\tilde{C}$ have been included in $\widetilde{C}_L$ and $\widetilde{C}_R$

The basic idea behind the algorithm is to implement a *greedy approach* (James et al., 2013), 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 C_2.[7]

Two 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*, 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 $\tilde{z}_i$ 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. On the one hand, this feature makes the algorithm more demanding, as 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. [8]

3 The `ciasearch` command

This section describes the syntax of the command `ciasearch`, that implements the algorithm described in Section 2.4 that searches for a vector of covariates **w** satisfying the CIA condition (2.3). The common support condition (2.4) can be verified using the command `ciacs` (see Section 6) once a candidate **w** has been found with `ciasearch`.

[7] This exercise would soon become intractable from a computational point of view as 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 1023. This case is still tractable. However, adding just 10 other covariates drives the number of combinations over 1 million.

[8] While not problematic in population, in sample it could be that the union of two non-identical sets of covariates satisfying the CIA on separate sides of the cutoff does not satisfy the CIA simultaneously on both sides.

3.1 Syntax

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

where *varlist* specifies the set of candidates C_2 and the option `included` eventually specifies the set of always included covariates C_1.

3.2 Options

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

score(*varname*) specifies the running variable.

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

cutoff(#) specifies the value of the cutoff. Default is `c(0)`. The cutoff value is subtracted from the *score* variable and the bandwidth. In case 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. Default is `p(1 1)`.

robust estimates heteroskedasticity-robust standard errors.

vce(*varname*) clusters standard errors at the specified level.

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

alpha(#) specifies the level of I-type error in the CIA test. Default is `alpha(0.1)`. In this case, the higher the value of `alpha` the easier 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 (non-dichotomic) 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` with this option switched on, the algorithm forgets the value of the loss function at the iteration $j-1$ and selects the covariate providing the lower value of the loss function at iteration j. In other words, with this option switched on, the algorithm searches for the covariate that minimizes the loss function within a certain iteration. This can make the loss function non-strictly 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 (2.3) 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(`*string*`)` [`cutoff(#)`
`poly(`*numlist*`)` `robust` `vce(`*varname*`)` `site(`*varname*`)` `details` `noise`]

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

4.2 Options

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

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

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

`cutoff(#)` specifies the value of the cutoff. Default is `c(0)`. The cutoff value is subtracted from the *score* variable and the bandwidth. In case 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. Default is `p(1 1)`.

`robust` estimates heteroskedasticity-robust standard errors.

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

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

`alpha(#)` specifies the level of I-type error in the CIA test. Default is `alpha(0.1)`. In this case, the higher the value of `alpha` the easier 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 full-sample and the restricted-sample. The restricted-sample is the sample composed by all units with no missing values in `outcome`, `score`, and *varlist*, whilst the full-sample is defined as those units with no missing entries just 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 the CIA condition (2.3) for a given input vector of covariates $\mathbf{w}_i$.

5.1 Syntax

`ciares` *varlist* [*if*] [*in*] `outcome(`*varname*`)` `score(`*varname*`)` `bandwidth(`*string*`)` [`cutoff(#)` `nbins(`*numlist*`)` `site(`*varname*`)` `cmpr(#)` `gphoptions(`*string*`)`]

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

5.2 Options

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

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

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

cutoff(#) specifies the value of the cutoff. Default is c(0). The cutoff value is subtracted from the *score* variable and the bandwidth. In case 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. Default is nbins(10 10).

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

cmpr(#) adds the conditional regression function of outcome on the score. The form of polynomials on the left and on the right can be modelled independently - eg. cmpr(2 3) for a second order polynomial on the left and a third order 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.

6 The ciacs command

This function describes the syntax of the command ciacs, that provides a graphical visualization of the common support condition (2.4) 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(*string*) [
cutoff(#) nbins(*numlist*) site(*varname*) asis gphoptions(*string*) pscore(*string*) probit
kdensity nograph]

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

6.2 Options

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

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

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

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

`cutoff(#)` specifies the value of the cutoff. Default is `c(0)`. The cutoff value is subtracted from the *score* variable and the bandwidth. In case 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. 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 pscore. This variable is added to the current dataset.

`probit` implements a probit model to estimate the pscore.

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

`nograph` suppresses any graphical output.

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 (2.6) or the propensity score reweighting estimator (2.7). By default, it estimates Average Treatment effect on the Treated (ATT) and Average Treatment effect on the Non-Treated (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 (2.6) is straightforward.

7.1 Syntax

`getaway` *varlist* [*if*] [*in*] `outcome(`*varname*`)` `score(`*varname*`)` `bandwidth(`*string*`)` [`cutoff(#)`
`method(`*string*`)` `site(`*varname*`)` `nquant(`*numlist*`)` `bootrep(#)` `qtleplot` `gphoptions(`*string*`)`
`gevar(`*string*`)` `asis`]

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

7.2 Options

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

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

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

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

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

`method(`*string*`)` allows to choose the estimation method between Linear Reweighting Estimator (*linear*) and Propensity Score Weighting Estimator (*pscore*). Default is `method(linear)`.

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

`bootrep(#)` sets the number of replications of the non-parametric bootstrap. Default is `bootrep(0)`. If `site` is specified a non-parametric block bootstrap is used.

`nquant(`*numlist*`)` specifies the number of quantiles in which the treatment effect must be estimated. It can be specified separately for each side of the cutoff. Default is `nquant(0)`. This option is mandatory if `qtleplot` is specified.

`qtleplot` plots estimated treatment effect over running variable quantiles together with bootstrapped standard errors. Also estimates and bootstrapped standard errors of the ATT and ATNT are reported.

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

`genvar(`*string*`)` specifies the name of the variable containing the distribution of treatment effects. Only with `method(linear)` option.

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

8 The `getawayplot` command

This section explains the syntax of the `getawayplot` command, which plots non-parametric 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(`*string*`)` [`cutoff(#)` `kernel(`*string*`)` `site(`*varname*`)` `degree(`*numlist*`)` `nbins(`*numlist*`)` `gphoptions(`*string*`)`]

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

8.2 Options

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

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

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

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

`cutoff(#)` specifies the value of the cutoff. Default is `c(0)`. The cutoff value is subtracted from the *score* variable and the bandwidth. In case 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 `lpoly`.

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

`degree(`*numlist*`)` 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. Default is `nbins(10 10)`.

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

9 Illustration of Methods

This section illustrates the main features of the `getaway` package using a simulated dataset, `simulated_getaway.dta`. The dataset contains 2,000 observations that are divided in 5 different groups (sites) of equal size.

```
. use "simulated_getaway.dta", clear

.
. 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, mc(cyan) m(o))                    ///
> (scatter Y X if site == 2, mc(midblue) m(o))                        ///
> (scatter Y X if site == 3, mc(ebblue) m(o))                         ///
> (scatter Y X if site == 4, mc(navy) m(o))                           ///
> (scatter Y X if site == 5, mc(dknavy) m(o)),                        ///
> xline(0, lc(red)) ylabel(,nogrid)                                   ///
> xtitle("Score") ytitle("Outcome") xlabel(-6(3)6) legend(off)
```

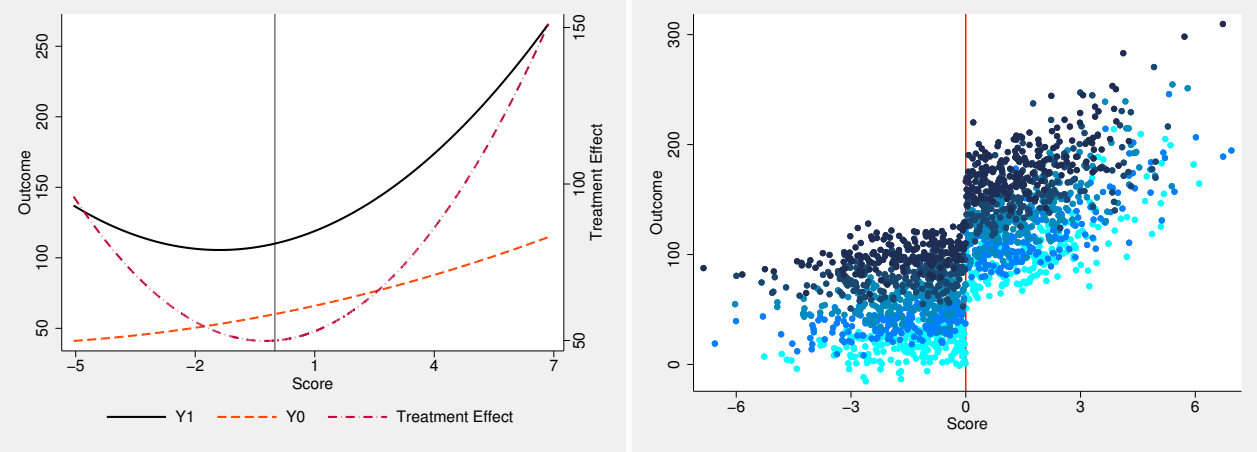
In this dataset, `Y` is the outcome variable, `X` is the standardized running variable, `T` is the treatment dummy, `cutoff` is a variable containing the corresponding cutoff for each unit, `site` is a variable containing the site identifier for each unit, and `w1`-`w10` are 10 covariates (henceforth, $\omega_i, i = 1, \ldots, 10$) to be used to validate the CIA assumption. In each site the cutoff is defined endogenously as $c - s = \text{median}(X_{is})$. Each covariate is generated according to $\omega_i \sim \mathtt{N}(\mu_i, 1)$, where $\mu_i \sim \mathtt{U}(-1, 1)$. The running variable is created as

$$X_{is} = \omega_{1,is}\phi_1 + \omega_{2,is}\phi_2 + \nu_{is}, \quad \nu_{is} \sim \mathtt{N}(0,1), \ \phi_j \sim \mathtt{U}(1,2), \ j = 1,2$$

The outcome variable is simulated according to the following data generating process (DGP)

$$Y_{is} = \alpha_s + T_{is}\beta + \mathbf{w}'_{is}\boldsymbol{\gamma} + \mathbf{w}'_{is}\mathbf{w}_{is}\delta + T_{is}\mathbf{w}'_{is}\boldsymbol{\lambda} + T_{is}\mathbf{w}'_{is}\mathbf{w}_{is}\rho + \varepsilon_{is}$$

where $\varepsilon_{is} \sim \mathtt{N}(0, 10)$, $\beta = 50$, $\boldsymbol{\gamma} = (5, 5)'$, $\delta = 0.5$, $\boldsymbol{\lambda} = (1, 1)'$, $\rho = 2$. Finally, the fixed effect is $\alpha_s = 20s$, $s = 1, \ldots, 5$. Figure 1i shows the (averaged across groups) potential outcomes as a function of the running variable. The figure shows that the DGP features heterogeneous treatment effects with respect to the score (red dashed line). Figure 1ii plots the observed outcome variable for each site (the lightest blue is the group with $s = 1$, the darkest $s = 5$).



(i) *Potential Outcomes and Treatment Effect* **(ii)** *Observed Data*

Figure 1: *Simulation Results*

The data generating process considered in this example implies that

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

which is the vector of covariates that satisfies the CIA condition (2.3).

The command `ciasearch` allows to search among the 10 "candidate" covariates `w1-w10` the ones (if any) satisfying the CIA. The basic syntax is illustrated in the following snippet.

```
. ciasearch w1 w2 w3 w4 w5 w6 w7 w8 w9 w10, o(Y) s(X) c(0) b(7) ///
> site(site) quad noprint p(2) alpha(0.2)
------------------------------------------------------------------------------

Algorithm Path:

 Searching for a set of covariates validating the CIA on the left of the cutoff ...
Iteration #1 finished || Loss Function (>.2) 5.85839395460e-10 || Selected w2
Iteration #2 finished || Loss Function (>.2) .2108217225940878 || Selected w1

 Searching for a set of covariates validating the CIA on the right of the cutoff ...
Iteration #1 finished || Loss Function (>.2) 6.26487537864e-64 || Selected w2_sq
Iteration #2 finished || Loss Function (>.2) .000078257236205 || Selected w2Xw1
```

```
Iteration #3 finished || Loss Function (>.2) .0514292245204344 || Selected w1
Iteration #4 finished || Loss Function (>.2) .1267036981809298 || Selected w2
Iteration #5 finished || Loss Function (>.2) .8402292326904474 || Selected w1_sq
```

```
Results
Algorithm Converged - Selected Covariates on the Left: w2 w1
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, i.e. 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.[9] 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 test are presented in the next snippet.

```
. generate X2 = X^2
. generate interaction = X*T
. generate interaction2 = X2*T
. generate w1sq = w1^2
. generate w2sq = w2^2
. generate w2Xw1 = w2*w1
.
. ciatest w1 w2 w1sq w2sq w2Xw1, o(Y) s(X) c(0) b(7) p(2) site(site)
```

```
            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 = .1)
```

The output above shows the main statistics obtained from running a regression similar to (2.9) 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 `p(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

[9] The chosen set of covariates on the left is a sub-vector 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 0.2 (with the `alpha(0.2)` option).

the hypothesis tests by showing the F-statistic and the corresponding p-value. The last row reports the number of observation used.

Graphical evidence can be supportive to 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, i.e.

$$Y_{is} = \alpha_{L,s} + \mathbf{w}'_{is}\phi_L + \varepsilon_{is}, \quad \text{if} \quad -h_L \leq X_{is} < 0,$$
$$Y_{is} = \alpha_{R,s} + \mathbf{w}'_{is}\phi_R + \nu_{is}, \quad \text{if} \quad 0 \leq X_i < h_R,$$

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

- store the residuals of these regressions, namely

$$e^L_{is} := (Y_{is} - \widehat{\alpha}_{L,s} + \mathbf{w}'_{is}\widehat{\phi_L}), \quad e^R_{is} := (Y_{is} - \widehat{\alpha}_{R,s} + \mathbf{w}'_{is}\widehat{\phi_R})$$

- plot e^L_{is} and e^R_{is} on the running variable X_{is}

The implication being tested here is that once the variation in $\mathbf{w}_{is}$ is taken into account, if the CIA is satisfied, the running should have no explanatory power on the outcome variable. Therefore, the residuals (e^L_i, e^R_i) 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. Due to 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.

```
. ciares w1 w2 w1sq w2sq w2Xw1, o(Y) s(X) b(7) site(site) nb(10 10) ///
> gphoptions(xlabel(-6(3)6) title("") graphregion(color(white))     ///
> plotregion(color(white)) scheme(s2manual) )
```

Up to this point just the first of the two conditions underlying the CIA has been tested and validated. However, also the common support condition (2.4) needs to be checked. The command `ciacs` serves this precise purpose.

```
. ciacs w1 w2 w1sq w2sq w2Xw1, o(Y) assign(T) s(X) c(0) b(7) ///
> site(site) pscore(pscore) gphoptions(title("") graphregion(color(white)) ///
```

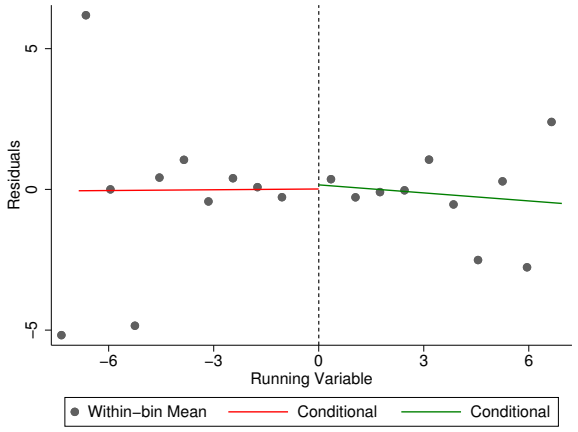


Figure 2: *Visualization of the CIA*

```
> plotregion(color(white)) scheme(s2manual) )
```

```
                        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

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

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 commons 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(varname)` that 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 >= r(CSmin) & pscore <= r(CSmax)
. tabulate incs T
```

	Treatment Dummy		
incs	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 either using histograms or kernel density estimators. Figure 3 shows the result obtained using histograms.

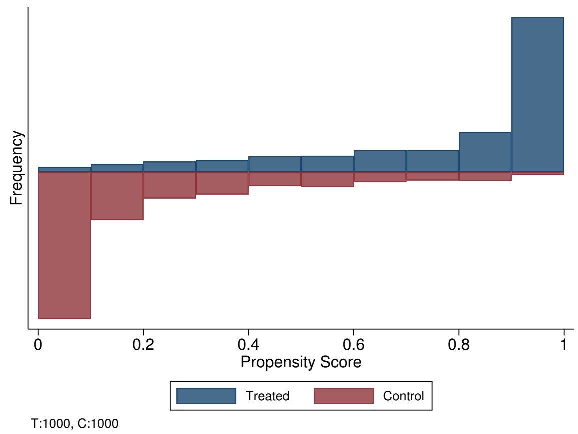


Figure 3: *Visualization of Common Support condition*

Now that both condition (2.3) and condition (2.4) have been verified, the commands `getaway` and `getawayplot` can be safely used to extrapolate treatment effects and estimate potential outcomes along the support of the running variable, respectively.

```
. getaway w1 w2 w1sq w2sq w2Xw1 if incs, o(Y) s(X) c(0) b(7) site(site) ///
> qtleplot nquant(5 5) boot(200) gphoptions(title("")) genvar(effect_est)
Bootstrapping standard errors ... (200)
----+--- 1 ---+--- 2 ---+--- 3 ---+--- 4 ---+--- 5
..................................................    50
..................................................   100
..................................................   150
..................................................   200
--------------------------------------------------------------------------------
             Extrapolation Results
--------------------------------------------------------------------------------
Outcome Variable            Y
Running Variable            X

Number of observations
              Treated       720
              Control       958
Cutoff                      0
Bandwidth                   7
Bootstrap Iterations        200
Site Fixed Effects          site
Method                      linear
```

Main Estimates

	Estimate	SE
ATNT	54.336	1.080
ATT	58.589	0.839

Within-Quantile Estimates

	Estimate	SE	Xlb	Xub
Left_1	56.598	2.530	-5.338	-2.441
Left_2	53.989	1.363	-2.435	-1.632
Left_3	53.176	0.989	-1.624	-0.987
Left_4	53.553	0.625	-0.981	-0.515
Left_5	54.359	0.565	-0.514	-0.001
Right_1	55.907	0.618	0.001	0.408
Right_2	56.560	1.041	0.411	0.824
Right_3	58.663	1.039	0.826	1.326
Right_4	59.418	1.068	1.330	1.958
Right_5	62.397	1.015	1.974	5.125

CIA Covariates: w1 w2 w1sq w2sq w2Xw1

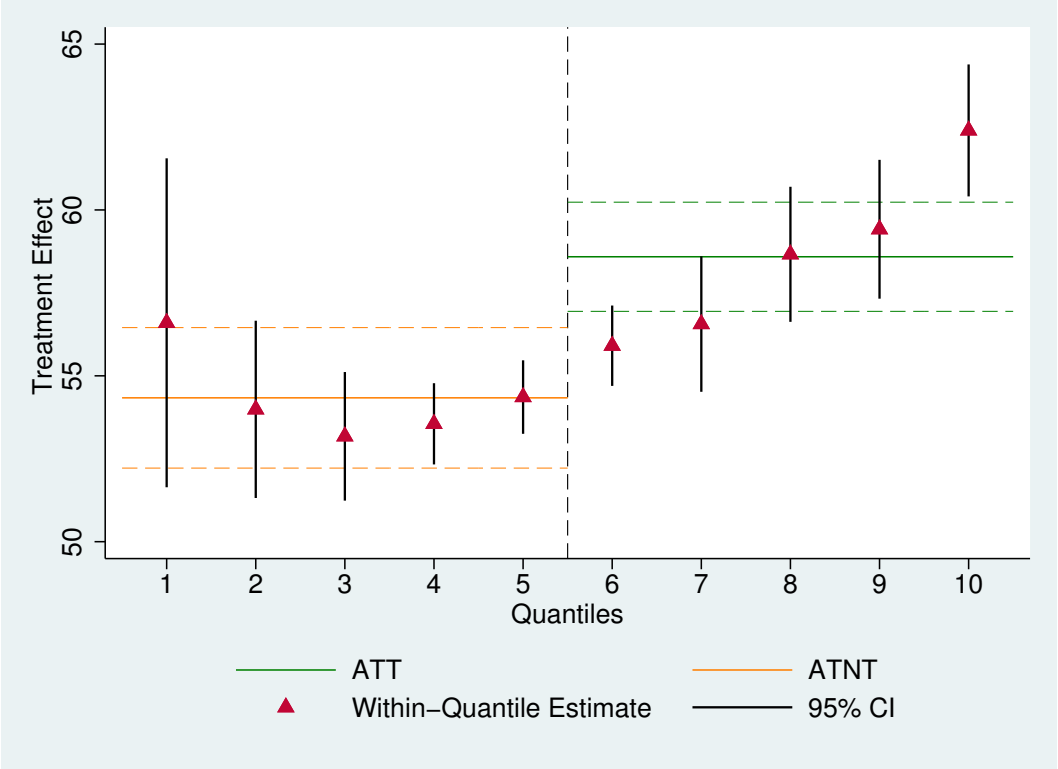


Figure 4: *Estimated treatment effects over the support of the running variable*

The command `getaway` prints the main specifics of the estimation procedure (outcome variable, running variable, number of observations, cutoff, bandwidth, ...) and two tables with the results. The first table contains the estimates for average treatment on the treated

$\tau_{[0,h_R)}$ and the average treatment on the non-treated $\tau_{(h_L,0)}$, 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); lower and upper bound 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 (red triangles), together with bootstrapped standard errors (vertical black solid lines) and the estimates of $\tau_{[0,h_R)}$ (green solid line to the right of the cutoff) and $\tau_{(h_L,0)}$ (orange 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 1i.

Moreover, the option `genvar(effect_est)` creates a new variables called `effect_est` that contains the treatment effect for each unit estimated according to either (2.6) (default) or (2.7). In this case, since the true treatment effect is known, it is possible to compare it with the estimated one.

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. 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.

```
. getawayplot w1 w2 w1sq w2sq w2Xw1, o(Y) s(X) c(0) b(7) k(triangle) ///
> d(2) nb(30) site(site) gphoptions(xlabel(-6(3)6))
```

Comparing Figure 5 with Figure 1i it is possible to 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.

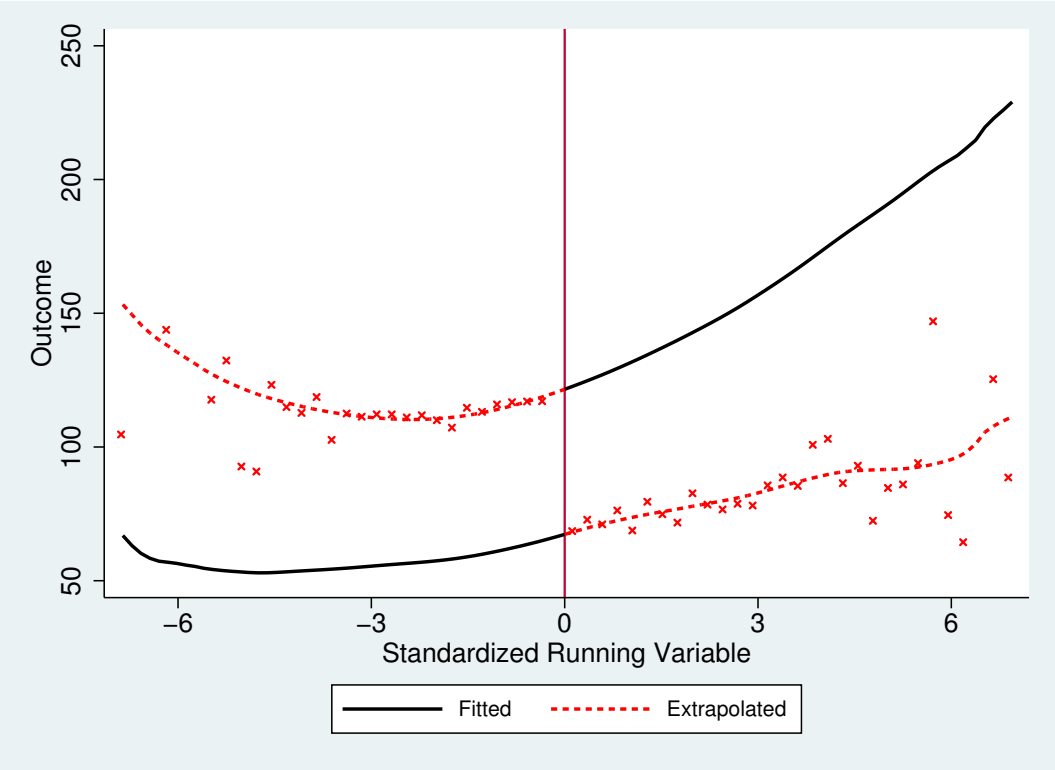


Figure 5: *Estimate potential outcomes over the support of the running variable*

11 Acknowledgements

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 are my own.

References

Angrist, J. D., and V. Lavy. 1999. Using Maimonides' rule to estimate the effect of class size on scholastic achievement. *The Quarterly journal of economics* 114(2): 533–575.

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(512): 1331–1344.

Baez, J., and A. Camacho. 2011. Assessing the long-term effects of conditional cash transfers on human capital: evidence from Colombia .

Battistin, E., A. Brugiavini, E. Rettore, and G. Weber. 2009. The retirement consumption puzzle: evidence from a regression discontinuity approach. *American Economic Review* 99(5): 2209–26.

Battistin, E., and E. Rettore. 2008. Ineligibles and eligible non-participants as a double comparison group in regression-discontinuity designs. *Journal of Econometrics* 142(2): 715–730.

Bennett, M. 2020. How Far is too Far? Estimation of an Interval for Generalization of a Regression Discontinuity Design Away from the Cutoff. URL https://www.magdalenabennett.com/files/sub/MBennett_GRD.pdf.

Bertanha, M. 2020. Regression discontinuity design with many thresholds. *Journal of Econometrics* 218(1): 216–241.

Bertanha, M., and G. W. Imbens. 2020. External validity in fuzzy regression discontinuity designs. *Journal of Business & Economic Statistics* 38(3): 593–612.

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.

Calonico, S., M. D. Cattaneo, and R. Titiunik. 2014. Robust data-driven inference in the regression-discontinuity design. *The Stata Journal* 14(4): 909–946.

Cattaneo, M. D., N. Idrobo, and R. Titiunik. 2019. *A Practical Introduction to Regression Discontinuity Designs: Foundations*. Cambridge University Press.

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(536): 1941–1952. URL 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. *The Stata Journal* 20(4): 866–891.

Cellini, S. R., F. Ferreira, and J. Rothstein. 2010. The value of school facility investments: Evidence from a dynamic regression discontinuity design. *The Quarterly Journal of Economics* 125(1): 215–261.

Coviello, D., and M. Mariniello. 2014. Publicity requirements in public procurement: Evidence from a regression discontinuity design. *Journal of Public Economics* 109: 76–100.

Dong, Y., and A. Lewbel. 2015. Identifying the effect of changing the policy threshold in regression discontinuity models. *Review of Economics and Statistics* 97(5): 1081–1092.

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(5): 1739–74.

Flammer, C. 2015. Does corporate social responsibility lead to superior financial performance? A regression discontinuity approach. *Management Science* 61(11): 2549–2568.

Fort, M., A. Ichino, E. Rettore, and G. Zanella. 2022. Multi-cutoff RD designs with observations located at each cutoff: problems and solutions .

Goldberger, A. S. 2008. Selection bias in evaluating treatment effects: Some formal illustrations. In *Modelling and Evaluating Treatment Effects in Econometrics*. Emerald Group Publishing Limited.

Hahn, J., P. Todd, and W. Van der Klaauw. 2001. Identification and estimation of treatment effects with a regression-discontinuity design. *Econometrica* 69(1): 201–209.

Heckman, J. J., R. J. LaLonde, and J. A. Smith. 1999. The economics and econometrics of active labor market programs. In *Handbook of labor economics*, 1865–2097. Vol. 3. Elsevier.

Hirano, K., G. W. Imbens, and G. Ridder. 2003. Efficient estimation of average treatment effects using the estimated propensity score. *Econometrica* 71(4): 1161–1189.

Imbens, G., and K. Kalyanaraman. 2012. Optimal bandwidth choice for the regression discontinuity estimator. *The Review of economic studies* 79(3): 933–959.

Imbens, G. W., and D. B. Rubin. 2015. *Causal inference in statistics, social, and biomedical sciences*. Cambridge University Press.

James, G., D. Witten, T. Hastie, and R. Tibshirani. 2013. *An introduction to statistical learning*. Vol. 112. Springer.

Keele, L. J., and R. Titiunik. 2015. Geographic boundaries as regression discontinuities. *Political Analysis* 23(1): 127–155.

Kline, P. 2011. Oaxaca-Blinder as a reweighting estimator. *American Economic Review* 101(3): 532–37.

Lalive, R. 2008. How do extended benefits affect unemployment duration? A regression discontinuity approach. *Journal of econometrics* 142(2): 785–806.

Lee, D. S. 2008. Randomized experiments from non-random selection in US House elections. *Journal of Econometrics* 142(2): 675–697.

Lee, D. S., E. Moretti, and M. J. Butler. 2004. Do voters affect or elect policies? Evidence from the US House. *The Quarterly Journal of Economics* 119(3): 807–859.

Ludwig, J., and D. L. Miller. 2007. Does Head Start improve children's life chances? Evidence from a regression discontinuity design. *The Quarterly journal of economics* 122(1): 159–208.

Meyersson, E. 2014. Islamic Rule and the Empowerment of the Poor and Pious. *Econometrica* 82(1): 229–269.

Ozier, O. 2018. The Impact of Secondary Schooling in Kenya A Regression Discontinuity Analysis. *Journal of Human Resources* 53(1): 157–188.

Pettersson-Lidbom, P. 2008. Do parties matter for economic outcomes? A regression-discontinuity approach. *Journal of the European Economic Association* 6(5): 1037–1056.

Pinotti, P. 2017. Clicking on heaven's door: The effect of immigrant legalization on crime. *American Economic Review* 107(1): 138–68.

Pop-Eleches, C., and M. Urquiola. 2013. Going to a better school: Effects and behavioral responses. *American Economic Review* 103(4): 1289–1324.

Rubin, D. B. 1974. Estimating causal effects of treatments in randomized and nonrandomized studies. *Journal of educational Psychology* 66(5): 688.

Thistlethwaite, D. L., and D. T. Campbell. 1960. Regression-discontinuity analysis: An alternative to the ex post facto experiment. *Journal of Educational Psychology* 51(6): 309.