rdrobust: An R Package for Robust Nonparametric Inference in Regression-Discontinuity Designs

by Sebastian Calonico, Matias D. Cattaneo and Rocío Titiunik

Abstract This article describes the R package **rdrobust**, which provides data-driven graphical and inference procedures for RD designs. The package includes three main functions: `rdrobust`, `rdbwselect` and `rdplot`. The first function (`rdrobust`) implements conventional local-polynomial RD treatment effect point estimators and confidence intervals, as well as robust bias-corrected confidence intervals, for average treatment effects at the cutoff. This function covers sharp RD, sharp kink RD, fuzzy RD and fuzzy kink RD designs, among other possibilities. The second function (`rdbwselect`) implements several bandwidth selectors proposed in the RD literature. The third function (`rdplot`) provides data-driven optimal choices of evenly-spaced and quantile-spaced partition sizes, which are used to implement several data-driven RD plots.

Introduction

The regression-discontinuity (RD) design is a widely employed quasi-experimental research design in social, behavioral and related sciences; for reviews see Imbens and Lemieux (2008) and Lee and Lemieux (2010). In this design, units are assigned to treatment based on whether their value of an observed covariate is above or below a known cutoff, and the probability of receiving treatment conditional on this covariate jumps discontinuously at the cutoff. This jump induces "variation" in treatment assignment that may be regarded, under appropriate assumptions, as being unrelated to potential confounders. Thus, inference in RD designs is typically conducted using only observations near the cutoff or threshold, where the discontinuous change in the probability of treatment assignment occurs. Due to its local nature, RD average treatment effects estimators are usually constructed using local-polynomial nonparametric regression, and statistical inference is based on large-sample approximations.

This article gives an introduction to the R package **rdrobust** (Calonico et al., 2015b), which offers an array of data-driven local-polynomial and partitioning-based inference procedures for RD designs. We introduce three main functions implementing several data-driven nonparametric point and confidence intervals estimators, bandwidth selectors, and plotting procedures useful for RD empirical applications:

- `rdrobust()`. This function implements the bias-corrected robust (to "large" bandwidth choices) inference procedure proposed by Calonico, Cattaneo, and Titiunik (2014a, CCT hereafter), as well as many other RD inference procedures employing local-polynomial regression. The function `rdrobust` offers bias-corrected robust confidence intervals for average treatment effects at the cutoff for sharp RD, sharp kink RD, fuzzy RD and fuzzy kink RD designs.

- `rdbwselect()`. This function implements several data-driven bandwidth selectors for RD designs based on the recent work of Imbens and Kalyanaraman (2012, IK hereafter) and CCT. Although this command may be used as a stand-alone bandwidth selector in RD applications, its main purpose is to provide fully data-driven bandwidth choices to be used by our main function `rdrobust()`.

- `rdplot()`. This function implements several data-driven optimal choices of evenly-spaced and quantile-spaced bins, which are useful to produce RD plots that either approximate the regression function by local sample averages or represent the overall variability of the data in a disciplined way. These optimal choices are based on an integrated mean squared error expansion of appropriately constructed partitioning estimators, as discussed in Calonico, Cattaneo, and Titiunik (in press); see also Cattaneo and Farrell (2013) for related results. These binned sample means and partition size chosen are used to construct the popular RD plots commonly found in RD applications in a fully automatic way.

We first provide a brief review of all the methods implemented in **rdrobust**, and then discuss an empirical illustration using some of the features of our functions `rdrobust()`, `rdbwselect()` and `rdplot()`. A full description of the capabilities of the package **rdrobust** is available in its manual and help files. A companion Stata (StataCorp., 2013) package is described in Calonico, Cattaneo, and Titiunik (2014b).

Review of methods

We now present the basic RD framework, describe the population parameters of interest, introduce the local-polynomial based estimators, review different inference procedures, and briefly summarize the popular RD plots. Regularity conditions and other technical aspects underlying the estimands and estimators may be found in the references given throughout. Recent results and further details on RD designs are discussed in IK and CCT (see also the supplemental appendix), and references therein.

Setup and notation

We adopt the potential outcomes framework commonly employed in the treatment effects literature (e.g., Heckman and Vytlacil 2007 and Imbens and Wooldridge 2009). Let $\{(Y_i(0), Y_i(1), T_i(0), T_i(1), X_i)' : i = 1, 2, \ldots, n\}$ be a random sample from $(Y(0), Y(1), T(0), T(1), X)'$, where $Y(1)$ and $Y(0)$ denote the potential outcomes with and without treatment, respectively, $T(0)$ and $T(1)$ denote potential treatment status, and the scalar regressor $X_i \in \mathbb{R}$ is the so-called "running variable" or "score" determining treatment assignment based on whether it exceeds a known cutoff. In particular, unit i is assigned treatment if $X_i > \bar{x}$ and not assigned treatment if $X_i < \bar{x}$, for some known fixed value $\bar{x} \in \mathbb{R}$. $f(x)$ denotes the continuous (Lebesgue) density of X_i.

This setup allows for imperfect compliance, which in the RD literature is known as the *fuzzy RD design*. The case of perfect treatment compliance is usually called the *sharp RD design*. In either case, the observed outcome and treatment status are, respectively,

$$Y_i = \begin{cases} Y_i(0) & \text{if } X_i < \bar{x} \\ Y_i(1) & \text{if } X_i \geq \bar{x} \end{cases} \quad \text{and} \quad T_i = \begin{cases} T_i(0) & \text{if } X_i < \bar{x} \\ T_i(1) & \text{if } X_i \geq \bar{x} \end{cases}.$$

The observed data is $\{(Y_i, T_i, X_i)' : i = 1, 2, \ldots, n\}$, a random sample from a large population. In sharp RD designs, $T_i = \mathbb{1}(X_i \geq \bar{x})$ with $\mathbb{1}(\cdot)$ denoting the indicator function, which leads to $\mathbb{P}[T_i = 0 | X_i < \bar{x}] = 1 = \mathbb{P}[T_i = 1 | X_i \geq \bar{x}]$. More generally, in fuzzy RD designs, treatment assignment and treatment status may differ (imperfect compliance). Therefore, for each unit i, the scalar random variable $Y_i \in \mathbb{R}$ denotes the outcome of interest, and $T_i \in \{0, 1\}$ denotes actual treatment take-up ($T_i = 1$ treatment taken, $T_i = 0$ treatment not taken).

We introduce some additional notation. For $\nu \in \mathbb{Z}_+ = \{0, 1, 2, \ldots\}$, define

$$\mu_{Y+}^{(\nu)}(\bar{x}) = \lim_{x \to \bar{x}^+} \frac{\partial^\nu}{\partial x^\nu} \mu_{Y+}(x), \qquad \mu_{Y-}^{(\nu)}(\bar{x}) = \lim_{x \to \bar{x}^-} \frac{\partial^\nu}{\partial x^\nu} \mu_{Y-}(x),$$

with $\mu_{Y+}(x) = \mathbb{E}[Y(1)|X = x]$ and $\mu_{Y-}(x) = \mathbb{E}[Y(0)|X = x]$, and

$$\mu_{T+}^{(\nu)}(\bar{x}) = \lim_{x \to \bar{x}^+} \frac{\partial^\nu}{\partial x^\nu} \mu_{T+}(x), \qquad \mu_{T-}^{(\nu)}(\bar{x}) = \lim_{x \to \bar{x}^-} \frac{\partial^\nu}{\partial x^\nu} \mu_{T-}(x),$$

with $\mu_{T+}(x) = \mathbb{E}[T(1)|X = x]$ and $\mu_{T-}(x) = \mathbb{E}[T(0)|X = x]$. Whenever there is no ambiguity, we drop the subindex denoting the dependent variable or the point of evaluation in the conditional expectations and other functions.

Population parameters of interest

We focus on average treatment effects at the cutoff in the sharp RD, fuzzy RD, sharp kink RD and fuzzy kink RD designs, although the results cover other possibilities. For further details on the interpretation of these estimands and regularity conditions see, among others, Hahn et al. (2001), Porter (2003), Lee (2008), IK, CCT, Card et al. (2014), and references therein.

- **Sharp RD designs**. Two popular parameters of interest are the sharp RD average treatment effect at the threshold, denoted by τ_0, and the sharp kink RD average treatment effect at the threshold, denoted by τ_1/κ, with κ a known scale constant, and using the generic notation:

$$\tau_\nu := \tau_{Y,\nu}(\bar{x}) = \frac{\partial^\nu}{\partial x^\nu} \mathbb{E}[Y_i(1) - Y_i(0)|X_i = x]\Big|_{x = \bar{x}} = \mu_{Y+}^{(\nu)} - \mu_{Y-}^{(\nu)}, \qquad \nu \in \mathbb{Z}_+,$$

  where the definition drops the subindex denoting the dependent random variable and evaluation point for notational simplicity. The last equality is a nonparametric identification result that holds under mild continuity conditions: the parameter τ_ν can be written as a function of observed data because $\mu_{Y+}^{(\nu)} = \mu_{Y+}^{(\nu)}(\bar{x})$ and $\mu_{Y-}^{(\nu)} = \mu_{Y-}^{(\nu)}(\bar{x})$ are estimable from the observed data; $\tau_\nu = \tau_{Y,\nu}$ is a difference of two (one-sided) nonparametric regression functions at $\bar{x}$.

- **Fuzzy RD designs**. The parameters of interest take the form

$$\varsigma_\nu := \varsigma_\nu(\bar{x}) = \frac{\frac{\partial^\nu}{\partial x^\nu}\mathbb{E}[Y_i(1) - Y_i(0)|X_i = x]\big|_{x=\bar{x}}}{\frac{\partial^\nu}{\partial x^\nu}\mathbb{E}[T_i(1) - T_i(0)|X_i = x]\big|_{x=\bar{x}}} = \frac{\tau_{Y,\nu}}{\tau_{T,\nu}} = \frac{\mu_{Y+}^{(\nu)} - \mu_{Y-}^{(\nu)}}{\mu_{T+}^{(\nu)} - \mu_{T-}^{(\nu)}}, \qquad \nu \in \mathbb{Z}_+,$$

where the last two equalities represent a nonparametric identification result that holds under mild continuity conditions. Note that $\mu_{Y+}^{(\nu)} = \mu_{Y+}^{(\nu)}(\bar{x})$, $\mu_{Y-}^{(\nu)} = \mu_{Y-}^{(\nu)}(\bar{x})$, $\mu_{T+}^{(\nu)} = \mu_{T+}^{(\nu)}(\bar{x})$ and $\mu_{T-}^{(\nu)} = \mu_{T-}^{(\nu)}(\bar{x})$ are all estimable from the observed data, since $\tau_{Y,\nu}$ and $\tau_{T,\nu}$ are each a difference of two (one-sided) regression functions at $\bar{x}$, and ς_ν is just their ratio. In the RD literature, the two main parameters of interest are ς_0, the fuzzy RD average treatment effect at the cutoff, and ς_1, the fuzzy kink RD average treatment effect at the cutoff.

Local polynomial estimators

Statistical inference in the RD design reduces to nonparametric regression inference at the induced boundary point $\bar{x}$, employing observations at either side of the threshold separately. Local-polynomial estimators have become the preferred choice of nonparametric estimator in the RD literature because of their excellent boundary properties (Fan and Gijbels, 1996). The package **rdrobust** implements local polynomial estimators of various orders for the two variables Y_i and T_i depending on the RD design considered, and also includes different bandwidths selectors and alternative confidence intervals estimators. All these features are briefly reviewed in the following subsections, but first we introduce the local polynomial RD estimators of order p in general. To reduce repetition, we describe the estimators using a generic outcome variable Z which either takes the value Y or T, depending on the outcome variable under consideration. For $Z \in \{Y, T\}$ and $\nu, p \in \mathbb{Z}_+$ with $\nu \le p$,

$$\hat{\tau}_{Z,\nu,p}(x; h_n) = \hat{\mu}_{Z+,p}^{(\nu)}(x; h_n) - \hat{\mu}_{Z-,p}^{(\nu)}(x; h_n),$$

$$\hat{\mu}_{Z+,p}^{(\nu)}(x; h_n) = \mathbf{e}_\nu' \hat{\boldsymbol{\beta}}_{Z+,p}(x; h_n) \qquad \text{and} \qquad \hat{\mu}_{Z-,p}^{(\nu)}(x; h_n) = \mathbf{e}_\nu' \hat{\boldsymbol{\beta}}_{Z-,p}(x; h_n),$$

$$\hat{\boldsymbol{\beta}}_{Z+,p}(x; h_n) = \arg\min_{\boldsymbol{\beta}\in\mathbb{R}^{p+1}} \sum_{i=1}^n \mathbb{1}(X_i \ge x)(Z_i - \mathbf{r}_p(X_i - x)'\boldsymbol{\beta})^2 K_{h_n}(X_i - x),$$

$$\hat{\boldsymbol{\beta}}_{Z-,p}(x; h_n) = \arg\min_{\boldsymbol{\beta}\in\mathbb{R}^{p+1}} \sum_{i=1}^n \mathbb{1}(X_i < x)(Z_i - \mathbf{r}_p(X_i - x)'\boldsymbol{\beta})^2 K_{h_n}(X_i - x),$$

where here $\mathbf{r}_p(x) = (1, x, \ldots, x^p)'$, $\mathbf{e}_\nu$ is the conformable $(\nu+1)$-th unit vector (e.g., $\mathbf{e}_1 = (0, 1, 0)'$ if $p = 2$), $K_h(u) = K(u/h)/h$ with $K(\cdot)$ a kernel function and h_n is a positive bandwidth sequence.

Using the generic notation above, the sharp RD estimators are:

$$\hat{\tau}_{\nu,p}(h_n) := \hat{\tau}_{Y,\nu,p}(\bar{x}; h_n), \qquad \nu \le p.$$

Similarly, for the fuzzy RD designs we have the RD estimators:

$$\hat{\varsigma}_{\nu,p}(h_n) := \frac{\hat{\tau}_{Y,\nu,p}(h_n)}{\hat{\tau}_{T,\nu,p}(h_n)}, \qquad \hat{\tau}_{Y,\nu,p}(h_n) := \hat{\tau}_{Y,\nu,p}(\bar{x}; h_n), \qquad \hat{\tau}_{T,\nu,p}(h_n) := \hat{\tau}_{T,\nu,p}(\bar{x}; h_n), \qquad \nu \le p.$$

Assuming the bandwidth $h_n \to 0$ and other regularity conditions hold, consistency of these estimators follows easily from well-known properties of local polynomial estimators. In applications, the most common choices are $p = 1$ for τ_0 (local-linear sharp RD estimator), $p = 2$ for τ_1 (local-quadratic sharp kink RD estimator), $p = 1$ for ς_0 (local-linear fuzzy RD estimator), and $p = 2$ for ς_1 (local-quadratic fuzzy kink RD estimator).

The function rdrobust() implements the above RD point estimators with options c to set $\bar{x}$, deriv to set ν, p to set p, kernel to set $K(\cdot)$, and h to set h_n, among others.

Bandwidth selectors

The main obstacle in the practical implementation of RD local polynomial estimators is bandwidth selection. The package **rdrobust** implements the main two approaches for bandwidth selection available in the literature: (i) plug-in rules based on mean squared error (MSE) expansions, and (ii) cross validation. IK provide a comprehensive review of these approaches.

- **Direct plug-in rules**. Direct plug-in (DPI) approaches to bandwidth selection are based on a mean

squared error (MSE) expansion of the sharp RD estimators, leading to the MSE-optimal choice

$$h_{\text{MSE},\nu,p} = C_{\text{MSE},\nu,p}\, n^{-\frac{1}{2p+3}}, \qquad C_{\text{MSE},\nu,p} = \left(\frac{(1+2\nu)V_{\nu,p}}{2(p+1-\nu)B_{\nu,p}^2} \right)^{\frac{1}{2p+3}},$$

where $B_{\nu,p}$ and $V_{\nu,p}$ are the leading asymptotic bias and variance of the RD estimator, respectively. The package **rdrobust** implements two data-driven versions of this MSE-optimal bandwidth, one proposed by IK (available for $\nu = 0$ only) and the other proposed by CCT (valid for all choices of ν). Both implementations include regularization as originally discussed in IK, although the option scalereg allows users to remove it. The IK implementation, denoted by $\hat{h}_{\text{IK},0,p}$, may be viewed as a nonparametric first-generation plug-in rule (e.g., Wand and Jones, 1995), sometimes called a DPI-1 (direct plug-in of order 1) selector. The CCT implementation, denoted by $\hat{h}_{\text{CCT},\nu,p}$, may be viewed as a second-generation plug-in bandwidth selection approach. Further details on implementation of the bandwidth selectors may be found in IK, CCT and their supplemental materials.

- **Cross validation**. This bandwidth choice is implemented as follows:

$$\hat{h}_{\text{CV},p} = \arg\min_{h>0} \text{CV}_\delta(h), \qquad \text{CV}_\delta(h) = \sum_{i=1}^{n} \mathbb{1}\left(X_{-,[\delta]} \le X_i \le X_{+,[\delta]} \right) \left(Y_i - \hat{\mu}_p\left(X_i; h\right) \right)^2,$$

where

$$\hat{\mu}_p(x;h) = \left\{ \begin{array}{ll} \mathbf{e}_0' \hat{\boldsymbol{\beta}}_{Y+,p}(x,h) & \text{if } x > \bar{x} \\ \mathbf{e}_0' \hat{\boldsymbol{\beta}}_{Y-,p}(x,h) & \text{if } x < \bar{x} \end{array} \right.,$$

and, for $\delta \in (0,1)$, $X_{-,[\delta]}$ and $X_{+,[\delta]}$ denote the δ-th quantile of $\{X_i : X_i < \bar{x}\}$ and $\{X_i : X_i > \bar{x}\}$, respectively. See IK for further discussion on this alternative approach, which is valid only for sharp RD designs ($\nu = 0$).

The function rdbwselect() implements the above bandwidth selectors.

Asymptotic properties and confidence intervals

We briefly review the main asymptotic properties of the local polynomial RD estimators, with particular emphasis on the properties of the associated confidence interval estimators. Specifically, we discuss three types of confidence intervals (CI) based on Gaussian approximations: (i) conventional CI (assuming "small" bias), (ii) bias-corrected CI (not necessarily requiring undersmoothing), and (iii) robust bias-corrected CI (not necessarily requiring undersmoothing).

- **Optimal point estimators**. The package **rdrobust** implements the following data-driven RD treatment effect point estimators.

$$\begin{array}{llll} \text{Sharp RD:} & \hat{\tau}_{\nu,p}(\hat{h}_{\text{CCT},\nu,p}), & \hat{\tau}_{0,p}(\hat{h}_{\text{IK},0,p}), & \hat{\tau}_{0,p}(\hat{h}_{\text{CV},p}), & \nu \le p. \\ \text{Fuzzy RD:} & \hat{\varsigma}_{\nu,p}(\hat{h}_{\text{CCT},\nu,p}), & \hat{\varsigma}_{0,p}(\hat{h}_{\text{IK},0,p}), & \hat{\varsigma}_{0,p}(\hat{h}_{\text{CV},p}), & \nu \le p. \end{array}$$

These estimators are constructed employing MSE-optimal bandwidth choices for the sharp RD case, which means that $\hat{\tau}_{\nu,p}(\hat{h}_{\text{CCT},\nu,p})$, $\hat{\tau}_{0,p}(\hat{h}_{\text{IK},0,p})$ and $\hat{\tau}_{0,p}(\hat{h}_{\text{CV},p})$ may be interpreted as consistent and (asymptotically) MSE-optimal point estimators of τ_ν. For the fuzzy RD cases, the bandwidth choices employed are technically optimal only for the numerator of the estimators, but since the rate of the MSE-optimal bandwidth choice does not differ from the sharp RD case, the estimators $\hat{\varsigma}_{\nu,p}(\hat{h}_{\text{CCT},\nu,p})$, $\hat{\varsigma}_{0,p}(\hat{h}_{\text{IK},\nu,p})$ and $\hat{\varsigma}_{0,p}(\hat{h}_{\text{CV},p})$ may also be viewed as consistent and (asymptotically) MSE-optimal point estimators of ς_ν.

- **Sharp RD confidence intervals**. Confidence intervals accompanying the point estimators discussed above rely on the following distributional approximation:

$$\sqrt{nh_n^{1+2\nu}} \left(\hat{\tau}_{\nu,p}(h_n) - \tau_\nu - h_n^{p+1-\nu} B_{\nu,p} \right) \to_d \mathcal{N}(0, V_{\nu,p}), \qquad \nu \le p, \tag{1}$$

where $B_{\nu,p}$ and $V_{\nu,p}$ denote, respectively, the asymptotic bias and variance of the RD estimator.

<u>Conventional confidence intervals</u>. An asymptotic $100(1-\alpha)$-percent confidence interval for τ_ν is

$$\text{CI}_{1-\alpha}(h_n) = \left[\hat{\tau}_{\nu,p}(h_n) \pm \Phi_{1-\alpha/2}^{-1} \sqrt{\frac{V_{\nu,p}}{nh_n^{1+2\nu}}} \right],$$

where Φ_a^{-1} denotes the appropriate quantile of the Gaussian distribution (e.g., 1.96 for $a = .975$). This approach is valid only if the leading bias in (1) is "small". This smoothing bias is ignored by relying on an "undersmoothing" argument, that is, by assuming the bandwidth chosen is "small" enough so that the bias is negligible. In practice, however, this procedure may be difficult to implement because most bandwidth selectors, such as $h_{\text{MSE},\nu,p}$, will not satisfy the conditions required for undersmoothing. This fact implies that most empirical bandwidth selectors could in principle lead to a non-negligible leading bias in the distributional approximation, which in turn will bias the associated confidence intervals.

Bias-corrected confidence intervals. As an alternative to undersmoothing, we can directly bias-correct the estimator by constructing an estimator of $B_{\nu,p}$, which is then subtracted from the RD point estimate in an attempt to eliminate the leading bias in (1). The resulting asymptotic $100(1 - \alpha)$-percent confidence interval for τ_ν is

$$\text{CI}_{1-\alpha}^{\text{bc}}(h_n, b_n) = \left[\left(\hat{\tau}_{\nu,p}(h_n) - h_n^{p+1-\nu} \, \hat{B}_{\nu,p,q} \right) \pm \Phi_{1-\alpha/2}^{-1} \sqrt{\frac{V_{\nu,p}}{nh_n^{1+2\nu}}} \right],$$

where $\hat{B}_{\nu,p,q}$ denotes the bias estimate, which is constructed using a possibly different bandwidth b_n. To implement this approach, CCT propose a MSE-optimal bandwidth choice of b_n for the bias estimator $\hat{B}_{\nu,p,q}$ taking the form $b_{\text{MSE},\nu,p,q} = C_{\nu,p,q}^{1/(2q+3)} \, n^{-1/(2q+3)}$, where $C_{\nu,p,q}$ denotes a constant depending on the data generating process. CCT also discuss an implementation procedure of $b_{\text{MSE},\nu,p,q}$, leading to the data-driven estimator denoted by $\hat{b}_{\text{CCT},\nu,p,q}$.

Robust bias-corrected confidence intervals. The confidence intervals discussed so far have some unappealing properties that may affect their performance in applications. On the one hand, the confidence intervals $\text{CI}_{1-\alpha}(h_n)$ require undersmoothing (or, alternatively, a "small" bias), which may lead to coverage distortions in cases where the bias is important. On the other hand, the bias-corrected confidence intervals $\text{CI}_{1-\alpha}^{\text{bc}}(h_n, b_n)$, while theoretically justified for a larger range of bandwidths, are usually regarded as having poor performance in empirical settings, also leading to potentially large coverage distortions in applications.

CCT propose an alternative, more robust confidence interval formula based on the bias-corrected RD treatment effect estimators, but employing a different variance for studentization purposes. Intuitively, the bias-corrected RD estimator does not perform well in finite-samples because the bias estimate introduces additional variability in the statistic, which is not accounted for when forming the associated confidence intervals $\text{CI}_{1-\alpha}^{\text{bc}}(h_n, b_n)$. Thus, CCT propose the asymptotic $100(1 - \alpha)$-percent confidence interval for τ_ν given by

$$\text{CI}_{1-\alpha}^{\text{rbc}}(h_n, b_n) = \left[\left(\hat{\tau}_{\nu,p}(h_n) - h_n^{p+1-\nu} \, \hat{B}_{\nu,p,q} \right) \pm \Phi_{1-\alpha/2}^{-1} \sqrt{V_{n,\nu,p,q}^{\text{bc}}} \right],$$

which includes the alternative variance formula $V_{n,\nu,p,q}^{\text{bc}}(h_n, b_n)$ accounting for the additional variability introduced by the bias estimate. See CCT and Calonico, Cattaneo, and Farrell (2015a) for further details.

- **Sharp RD variance estimation**. To construct fully feasible confidence intervals the unknown asymptotic variance is replaced by a consistent estimator thereof. The asymptotic variance formulas introduced above have a "sandwich" structure coming from the weighted least-squares structure of local polynomials. The package **rdrobust** offers two distinct valid variance estimators, employing either "plug-in estimated residuals" or "fixed-matches estimated residuals".

  Plug-in estimated residuals. In this approach, the unknown residuals are estimated directly using the RD local polynomial estimator. Usually the same bandwidth h_n is employed, although his choice may not be optimal and could lead to poor finite-sample performance of the variance estimator. $\check{V}_{n,\nu,p}$ and $\check{V}_{n,\nu,p,q}^{\text{bc}}$ denote the resulting estimators of $V_{n,\nu,p}$ and $V_{n,\nu,p,q}^{\text{bc}}$, respectively.

  Fixed-matches estimated residuals. CCT propose an alternative variance estimator employing a different construction for the residuals, motivated by the work of Abadie and Imbens (2006). This estimator is constructed using a simple nearest-neighbor (or fixed-matches) estimator for the residuals. $\hat{V}_{n,\nu,p}$ and $\hat{V}_{n,\nu,p,q}^{\text{bc}}$ denote the resulting estimators of $V_{n,\nu,p}$ and $V_{n,\nu,p,q}^{\text{bc}}$, respectively.

- **Extensions to Fuzzy RD confidence intervals**. All the ideas and results presented above extend to the case of fuzzy RD designs, which we omit here to conserve space. See IK and CCT for further details. The package **rdrobust** also implements fuzzy RD estimators and confidence intervals.

The function rdrobust() may be used to conduct inference in all RD settings; by default, this function employs the function rdbwselect() for bandwidth selection. Specifically, assuming y is the output variable, t is the treatment status variable, x is the running variable and the cutoff is $c = 0$, we have the following generic cases (with default options and bandwidth selection procedure):

- Sharp RD:
  ```
  rdrobust(y = y,x = x)
  ```
- Sharp Kink RD:
  ```
  rdrobust(y = y,x = x,deriv = 1)
  ```
- Fuzzy RD:
  ```
  rdrobust(y = y,x = x,fuzzy = t)
  ```
- Fuzzy Kink RD:
  ```
  rdrobust(y = y,x = x,fuzzy = t,deriv = 1)
  ```

RD plots

Because of the simplicity of the RD design, it is customary (and advisable) to summarize the main features of the RD design in a graphical way. The package **rdrobust** also implements the results in Calonico, Cattaneo, and Titiunik (in press), which offer several optimal data-driven choices of tuning parameters useful to produce several versions of the popular RD plots. These plots present global and local estimates of the regression functions, $\mu_{Y-}(x)$ and $\mu_{Y+}(x)$, in an attempt to describe their shapes for control ($X_i < \bar{x}$) and treated ($X_i \geq \bar{x}$) units. The plot typically includes two smooth "global" polynomial regression curve estimates, for control and treatment units separately, as well as binned sample means of the outcome, which are included either to capture the local behavior of the underlying regression functions or to represent the overall variability of the raw data in a disciplined way. These binned sample means are used to either (i) investigate whether other discontinuities are present, and (ii) depict an informative "cloud of points" around the smooth global polynomial fits.

The first ingredient of the RD plot (two regression curves estimated for $X_i < \bar{x}$ and $X_i \geq \bar{x}$ separately) is easy to construct because it requires only estimating a polynomial regression on the data; typical choices are 4-th and 5-th order global polynomials. The second ingredient of the RD plot requires computing sample means over non-overlapping regions of the support of the running variable X_i for control and treatment units, separately. For implementation the researcher needs to choose the number of bins to be used, denoted by $J_{-,n}$ (control) and $J_{+,n}$ (treatment), and the length of each bin; evenly-spaced bins is the most common partitioning scheme used. Calonico, Cattaneo, and Titiunik (in press) study the problem of selecting an optimal number of bins $J_{-,n}$ and $J_{+,n}$ under two partitioning schemes, evenly-spaced (ES) and quantile-spaced (QS), and propose several consistent nonparametric selectors using spacings and polynomial regression estimators. In particular, they propose two approaches to select the number of bins: (i) IMSE-optimal (tailored to approximate the underlying regression functions), and (ii) Mimicking Variance (tailored to represent the overall variability of the data in a disciplined way). These two proposed choices take the form, respectively,

$$\text{IMSE-optimal}: \quad J^*_{-,n} = \left\lceil \mathscr{C}_- \, n^{1/3} \right\rceil \quad \text{and} \quad J^*_{+,n} = \left\lceil \mathscr{C}_+ \, n^{1/3} \right\rceil,$$

and

$$\text{Mimicking Variance}: \quad J^*_{-,n} = \left\lceil \mathscr{C}_- \, \frac{n}{\log(n)^2} \right\rceil \quad \text{and} \quad J^*_{+,n} = \left\lceil \mathscr{C}_+ \, \frac{n}{\log(n)^2} \right\rceil,$$

where $\lceil \cdot \rceil$ denotes the ceiling function, and the unknown constants take different values depending on the targeted method and partitioning scheme used. Once the partitioning scheme is selected, the optimal choices $J^*_{-,n}$ and $J^*_{+,n}$ can be estimated using preliminary plug-in estimators of $\mathscr{C}_-$ and $\mathscr{C}_+$.

The function rdplot() offers eight distinct automatic implementations for RD plots depending on (i) the choice of partitioning (ES or QS), (ii) the goal of the plot (IMSE-optimal or Mimicking Variance), and (iii) the estimation approach used (spacings or polynomial regression). Specifically, the function rdplot() covers the following.

- Population quantities:
  $J_{\text{ES-}\mu,\cdot,n} = $ IMSE-optimal choice with evenly-spaced (ES) bins.
  $J_{\text{ES-}\vartheta,\cdot,n} = $ Mimicking Variance choice with evenly-spaced (ES) bins.
  $J_{\text{QS-}\mu,\cdot,n} = $ IMSE-optimal choice with quantile-spaced (QS) bins.
  $J_{\text{QS-}\vartheta,\cdot,n} = $ Mimicking Variance choice with quantile-spaced (QS) bins.

- Estimators:
  $\hat{J}_{\text{ES-}\mu,\cdot,n}$ and $\hat{J}_{\text{QS-}\mu,\cdot,n} = $ spacings implementations of $J_{\text{ES-}\mu,\cdot,n}$ and $J_{\text{QS-}\mu,\cdot,n}$, respectively.
  $\check{J}_{\text{ES-}\mu,\cdot,n}$ and $\check{J}_{\text{QS-}\mu,\cdot,n} = $ polynomial regression implementations of $J_{\text{ES-}\mu,\cdot,n}$ and $J_{\text{QS-}\mu,\cdot,n}$, respectively.
  $\hat{J}_{\text{ES-}\vartheta,\cdot,n}$ and $\hat{J}_{\text{QS-}\vartheta,\cdot,n} = $ spacings implementations of $J_{\text{ES-}\vartheta,\cdot,n}$ and $J_{\text{QS-}\vartheta,\cdot,n}$, respectively.
  $\check{J}_{\text{ES-}\vartheta,\cdot,n}$ and $\check{J}_{\text{QS-}\vartheta,\cdot,n} = $ polynomial regression implementations of $J_{\text{ES-}\vartheta,\cdot,n}$ and $J_{\text{QS-}\vartheta,\cdot,n}$, respectively.

Further details on implementation and syntax are given in the help file of the function `rdplot()`. For other technical and methodological details see Calonico, Cattaneo, and Titiunik (in press).

The rdrobust package: Empirical illustration

We employ an extract of the dataset constructed by Cattaneo, Frandsen, and Titiunik (2015) to illustrate some of the features of our R package **rdrobust**. This dataset contains information on elections for the U.S. Senate during the period 1914–2010. We focus on the RD effect of the Democratic party winning a U.S. Senate seat on the vote share obtained in the following election for that same seat, mimicking the analysis conducted in Lee (2008) for the U.S. House. The dataset `rdrobust_RDsenate` contains two variables: *vote* and *margin*. The variable *vote* records the state-level vote share of the Democratic party in a given statewide election for a Senate seat, while the variable *margin* records the margin of victory of the Democratic party in the previous election for the same Senate seat (i.e., six years prior).

First, we load the database and present basic summary statistics. The functions included in the R package **rdrobust** allow for missing values, which are automatically excluded for estimation purposes.

```
> library(rdrobust)
> data(rdrobust_RDsenate)
> vote <- rdrobust_RDsenate$vote
> margin <- rdrobust_RDsenate$margin
> summary(vote)
   Min. 1st Qu.  Median   Mean 3rd Qu.   Max.    NA's
   0.00   42.67   50.55  52.67   61.35 100.00      93
> summary(margin)
    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
-100.000 -12.210   2.166   7.171  22.770 100.000
```

This data set has a total of 1,297 complete observations. The variable *margin* ranges from −100 to 100, and records the margin of victory in a given election for a given U.S. Senate seat, defined as the vote share of the Democratic party minus the vote share of the strongest opponent. When *margin* is above zero, the Democratic party wins the election for that seat, otherwise it loses. The variable *vote* ranges from 0 to 100 because it records the outcome of the (two-periods ahead) election for that given seat. Thus, observations for years 2008 and 2010 have missing *vote*. As it is usual in the literature, we exploit the discontinuity in incumbency status that occurs at 0 on *margin* to employ an RD design.

We use `rdplot()` to construct an automatic plot of the RD design.

```
> (rdplot(y = vote, x = margin, title = "RD Plot - Senate Elections Data",
+    y.label = "Vote Share in Election at time t+1",
+    x.label = "Vote Share in Election at time t"))
Call:
rdplot(y = vote, x = margin, title = "RD Plot - Senate Elections Data",
   x.label = "Vote Share in Election at time t",
   y.label = "Vote Share in Election at time t+1")

Method: mimicking variance evenly-spaced method using spacings estimators
```

	Left	Right
Number of Obs.	595	702
Polynomial Order	4	4
Scale	1	1
Selected Bins	15	35
Bin Length	6.6614	2.8561
IMSE-optimal bins	8	9
Mimicking Variance bins	15	35
Relative to IMSE-optimal:		
Implied scale	1.8750	3.8889
WIMSE variance weight	0.1317	0.0167
WIMSE bias weight	0.8683	0.9833

Figure 1 is constructed using the default options in the command `rdplot`, which produce an RD plot with evenly-spaced bins selected to mimic the underlying variability of the data and is
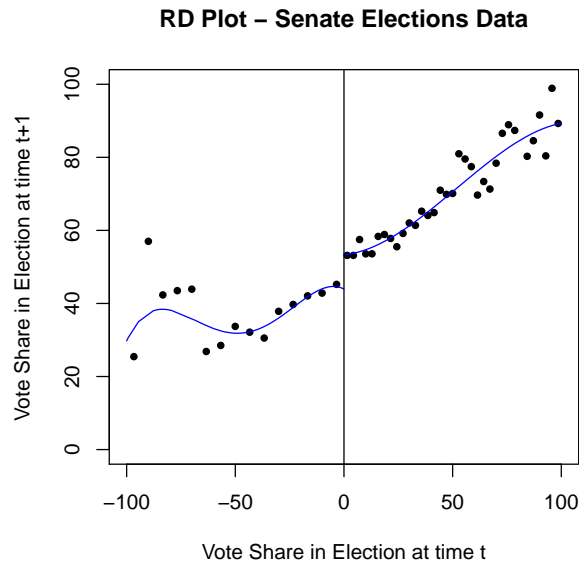
RD Plot – Senate Elections Data

Figure 1: Automatic RD plot with evenly spaced bins (Mimicking Variance).

implemented using spacings estimators. Using the notation introduced above, the number of optimal bins for control and treatment units are $\hat{J}_{-,n} = 15$ and $\hat{J}_{+,n} = 35$, respectively, implying bin lengths of 6.661 and 2.856 percentage points, respectively. The global polynomial is constructed using a 4*th* degree polynomial ($p = 4$ for $\hat{\mu}_{-\cdot p,1}(x)$ and $\hat{\mu}_{+\cdot p,1}(x)$).

Next, we construct an alternative RD plot using evenly-spaced bins selected to trace out the underlying regression function (i.e., using the IMSE-optimal selector), also implemented using spacings estimators. The resulting plot is given in Figure 2.

```
> (rdplot(y = vote, x = margin, binselect = "es",
+   title = "RD Plot - Senate Elections Data",
+   y.label = "Vote Share in Election at time t+1",
+   x.label = "Vote Share in Election at time t"))
Call:
rdplot(y = vote, x = margin, binselect = "es", x
   title = "RD Plot - Senate Elections Data",
   x.label = "Vote Share in Election at time t",
   y.label = "Vote Share in Election at time t+1")

Method: IMSE-optimal evenly-spaced method using spacings estimators


                         Left    Right
Number of Obs.           595     702
Polynomial Order         4       4
Scale                    1       1

Selected Bins            8       9
Bin Length               12.4901 11.1071

IMSE-optimal bins        8       9
Mimicking Variance bins  15      35

Relative to IMSE-optimal:
Implied scale            1.0000  1.0000
WIMSE variance weight    0.5000  0.5000
WIMSE bias weight        0.5000  0.5000
```

While providing a good approximation to the underlying regression function (taking the global polynomial fit as benchmark), the IMSE-optimal number of bins will usually be too small in applications. This happens because the optimal formulas seek to balance squared bias and variance in order to approximate the underlying regression function globally. To obtain a visual "cloud of points" we need to increase the number of bins, that is, undersmooth the estimator. In other words, in order to

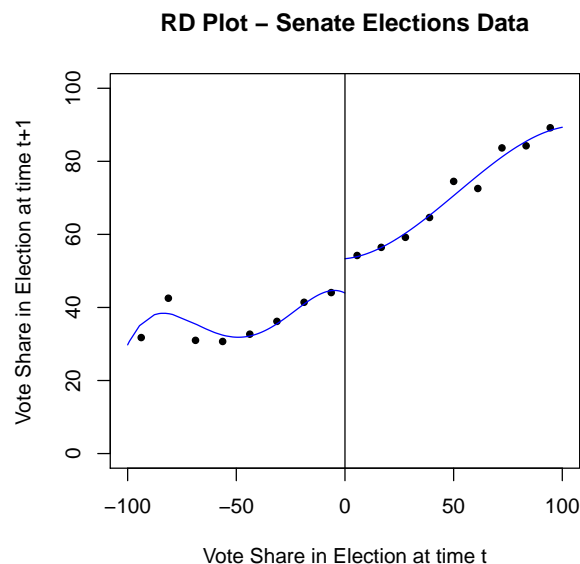**RD Plot – Senate Elections Data**

Figure 2: Automatic RD plot with evenly spaced bins (IMSE-optimal).

increase the overall variability of the plotted points, we may reduce the bin-length – which is done by increasing the total number of bins used. This may be easily done using the option `scale` as follows:

```
> (rdplot(y = vote, x = margin, binselect = "es", scale = 5,
+   title = "RD Plot - Senate Elections Data",
+   y.label = "Vote Share in Election at time t+1",
+   x.label = "Vote Share in Election at time t"))
Call:
rdplot(y = vote, x = margin, binselect = "es", scale = 5,
   title = "RD Plot - Senate Elections Data",
   x.label = "Vote Share in Election at time t",
   y.label = "Vote Share in Election at time t+1")

Method: IMSE-optimal evenly-spaced method using spacings estimators


                          Left    Right
Number of Obs.            595     702
Polynomial Order          4       4
Scale                     5       5

Selected Bins             40      45
Bin Length                2.4980  2.2214

IMSE-optimal bins         8       9
Mimicking Variance bins   15      35

Relative to IMSE-optimal:
Implied scale             5.0000 5.0000
WIMSE variance weight     0.0079 0.0079
WIMSE bias weight         0.9921 0.9921
```

Figure 3 shows the resulting (undersmoothed) RD plot, where now the number of bins used is five times larger than the optimal choice in an integrated mean squared error sense. The resulting estimator is naturally more variable than before.

Next, we conduct fully data-driven RD treatment effect estimation and inference. The function `rdrobust()` using its default options leads to the following output:

```
> rdrobust(y = vote, x = margin)

Call:
rdrobust(y = vote, x = margin)
```
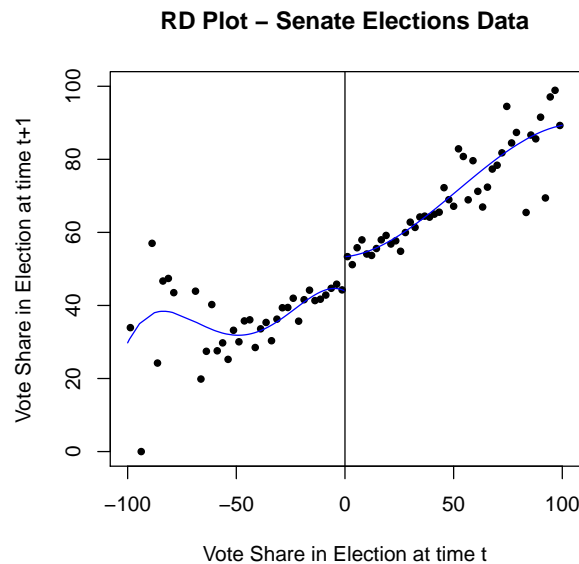
RD Plot – Senate Elections Data

Figure 3: Automatic RD plot with evenly spaced bins and scaled down bin length (IMSE-optimal).

```
Summary:

Number of Obs 1297
NN Matches     3
BW Type        CCT
Kernel Type    Triangular


                      Left    Right
Number of Obs         343     310
Order Loc Poly (p)    1       1
Order Bias (q)        2       2
BW Loc Poly (h)       16.7936 16.7936
BW Bias (b)           27.4372 27.4372
rho (h/b)             0.6121  0.6121


Estimates:
             Coef    Std. Err. z       P>|z|  CI Lower CI Upper
Conventional 7.4253  1.4954    4.9656  0.0000 4.4944   10.3562
Robust                                 0.0000 4.0697   10.9833
```

These results contain a variety of information, which is organized in three panels. The first two contain a summary of the main choices selected to construct the RD treatment effect estimators, while the lower panel includes the main estimation results. Specifically, using the notation introduced above, this table shows:

1. The total number of observations is $1,297$, with effective 343 control and 310 treated units (given the bandwidth h_n chosen; see below). The estimation is conducted using a local-linear ($p = 1$) estimator with a local-quadratic ($q = 2$) bias-correction estimate, with a triangular kernel. The standard error estimators are the ones proposed by CCT, computed using 3 nearest-neighbors.

2. The bandwidth selection procedure is the one proposed by CCT, leading to $\hat{h}_{\text{CCT},\nu,p} = 16.7936$ with $p = 1$, and $\hat{b}_{\text{CCT},\nu,p,q} = 27.4372$ with $q = 2$. Recall $\hat{h}_{\text{CCT},\nu,p}$ and $\hat{b}_{\text{CCT},\nu,p,q}$ are the estimated bandwidths used to construct the RD point estimator and the RD bias-correction, respectively, and $\nu = 0$ in this illustration.

3. The RD point estimator is $\hat{\tau}_p(\hat{h}_{\text{CCT},\nu,p}) = 7.4253$ and the RD robust confidence intervals is $\hat{\text{Cl}}_{1-\alpha}^{rbc}(\hat{h}_{\text{CCT},\nu,p}, \hat{b}_{\text{CCT},\nu,p,q}) = [\, 4.0697 \,,\, 10.9833 \,]$, with a default choice of $\alpha = 0.05$.

The function rdrobust() also offers a more detailed output, which includes all the point estimators, standard errors and confidence intervals discussed previously. These results are retrieved using the all = TRUE option. The corresponding output is as follows:

```
> rdrobust(y = vote, x = margin, all = TRUE)

Call:
rdrobust(y = vote, x = margin, all = TRUE)

Summary:

Number of Obs 1297
NN Matches    3
BW Type       CCT
Kernel Type   Triangular

                   Left    Right
Number of Obs      343     310
Order Loc Poly (p) 1       1
Order Bias (q)     2       2
BW Loc Poly (h)    16.7936 16.7936
BW Bias (b)        27.4372 27.4372
rho (h/b)          0.6121  0.6121

Estimates:
               Coef    Std. Err. z      P>|z|  CI Lower CI Upper
Conventional   7.4253  1.4954    4.9656 0.0000 4.4944   10.3562
Bias-Corrected 7.5265  1.4954    5.0333 0.0000 4.5957   10.4574
Robust         7.5265  1.7637    4.2675 0.0000 4.0697   10.9833
```

Finally, we illustrate all the bandwidth selection procedures contained in our package. To this end, we employ our companion function rdbwselect() to compare the CCT bandwidth selectors with the IK and CV approaches. We have:

```
> rdbwselect(y = vote, x = margin, all = TRUE)

Call:
rdbwselect(y = vote, x = margin, all = TRUE)

BW Selector   All
Number of Obs 1297
NN Matches    3
Kernel Type   Triangular

                   Left Right
Number of Obs      595  702
Order Loc Poly (p) 1    1
Order Bias (q)     2    2

        h        b
CCT 16.79357 27.43722
IK  15.66761 16.48524
CV  35.42113      NA
```

In this case we employed the option all = TRUE, which computes the three bandwidth selectors briefly discussed above. Notice that the option CV is currently not available for derivative estimation. To further understand the performance of the CV approach, we include a graph of the CV objective function over the grid being considered. This is done using the option cvplot as shown next (in this example we also changed the grid features to obtain a better plot, and to show this additional functionality in action as well).

```
> rdbwselect(y = vote, x = margin, bwselect = "CV",
+   cvgrid_min = 10, cvgrid_max = 80, cvplot = TRUE)

Call:
rdbwselect(y = vote, x = margin, bwselect = "CV", cvgrid_min = 10,
    cvgrid_max = 80, cvplot = TRUE)

BW Selector   CV
Number of Obs 1297
```

Cross−Validation Objective Function
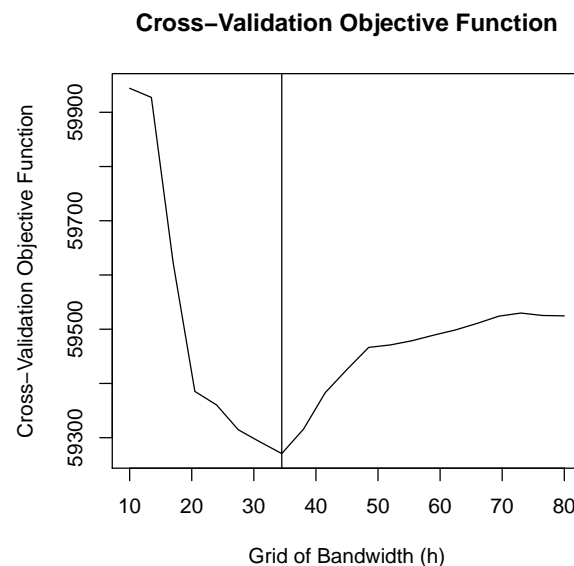

Figure 4: Values of the CV function over the selected grid.

```
NN Matches      3
Kernel Type    Triangular


                  Left Right
Number of Obs     595  702
Order Loc Poly (p) 1    1
Order Bias (q)     2    2

    h     b
  34.5  34.5
```

Figure 4 shows the CV objective function as a function of a grid of bandwidth. In this example, the cross validation approach delivers a minimum at $\hat{h}_{\text{CV},p} = 34.5$.

Our functions contained in the R package **rdrobust** have many other options. For instance, for the main function rdrobust() we have the following additional examples (output is not provided to conserve space).

- Estimation using uniform kernel:
  rdrobust(y = vote,x = margin,kernel = "uniform")

- Estimation using the IK bandwidth selector:
  rdrobust(y = vote,x = margin,bwselect = "IK")

- Estimation using the CV bandwidth selector:
  rdrobust(y = vote,x = margin,bwselect = "CV")

- Estimation using $h_n = 15$ and $\rho = h_n/b_n = 0.8$:
  rdrobust(y = vote,x = margin,h = 15,rho = 0.8)

- Estimation using $p = 2$ and $q = 4$:
  rdrobust(y = vote,x = margin,p = 2,q = 4)

- Estimation using plug-in residuals estimates:
  rdrobust(y = vote,x = margin,vce = "resid")

Conclusions

We introduced the main features of the R package **rdrobust**, which includes the functions rdrobust(), rdbwselect(), and rdplot() designed to conduct data-driven nonparametric robust inference in RD designs. This implementation covers average RD treatment effects at the cutoff in the sharp RD, sharp kink RD, fuzzy RD and fuzzy kink RD designs. A full description of this package may be found in its manual and help files. A companion Stata package offering the same structure and capabilities is described in Calonico, Cattaneo, and Titiunik (2014b).

Acknowledgments

Bibliography

A. Abadie and G. W. Imbens. Large sample properties of matching estimators for average treatment effects. *Econometrica*, 74(1):235–267, 2006. [p42]

S. Calonico, M. D. Cattaneo, and R. Titiunik. Robust nonparametric confidence intervals for regression-discontinuity designs. *Econometrica*, 82(6):2295–2326, 2014a. [p38]

S. Calonico, M. D. Cattaneo, and R. Titiunik. Robust data-driven inference in the regression-discontinuity design. *Stata Journal*, 14(4):909–946, 2014b. [p38, 49]

S. Calonico, M. D. Cattaneo, and M. H. Farrell. On the effect of bias estimation on coverage accuracy in nonparametric estimation. Working paper, University of Michigan, 2015a. [p42]

S. Calonico, M. D. Cattaneo, and R. Titiunik. *rdrobust: Robust Data-Driven Statistical Inference in Regression-Discontinuity Designs*, 2015b. URL http://CRAN.R-project.org/package=rdrobust. R package version 0.80. [p38]

S. Calonico, M. D. Cattaneo, and R. Titiunik. Optimal data-driven regression discontinuity plots. *Journal of the American Statistical Association*, in press. doi: 10.1080/01621459.2015.1017578. [p38, 43, 44]

D. Card, D. S. Lee, Z. Pei, and A. Weber. Inference on causal effects in a generalized regression kink design. Working paper, UC Berkley, 2014. [p39]

M. D. Cattaneo and M. H. Farrell. Optimal convergence rates, Bahadur representation, and asymptotic normality of partitioning estimators. *Journal of Econometrics*, 174(2):127–143, 2013. [p38]

M. D. Cattaneo, B. Frandsen, and R. Titiunik. Randomization inference in the regression discontinuity design: An application to party advantages in the U.S. Senate. *Journal of Causal Inference*, 3(1):1–24, 2015. [p44]

J. Fan and I. Gijbels. *Local Polynomial Modelling and Its Applications*. Chapman & Hall/CRC, New York, 1996. [p40]

J. Hahn, P. Todd, and W. van der Klaauw. Identification and estimation of treatment effects with a regression-discontinuity design. *Econometrica*, 69(1):201–209, 2001. [p39]

J. J. Heckman and E. J. Vytlacil. Econometric evaluation of social programs, part I: Causal models, structural models and econometric policy evaluation. In J. Heckman and E. Leamer, editors, *Handbook of Econometrics*, volume VI, pages 4780–4874. Elsevier Science B.V., 2007. [p39]

G. Imbens and T. Lemieux. Regression discontinuity designs: A guide to practice. *Journal of Econometrics*, 142(2):615–635, 2008. [p38]

G. W. Imbens and K. Kalyanaraman. Optimal bandwidth choice for the regression discontinuity estimator. *Review of Economic Studies*, 79(3):933–959, 2012. [p38]

G. W. Imbens and J. M. Wooldridge. Recent developments in the econometrics of program evaluation. *Journal of Economic Literature*, 47(1):5–86, 2009. [p39]

D. S. Lee. Randomized experiments from non-random selection in U.S. House elections. *Journal of Econometrics*, 142(2):675–697, 2008. [p39, 44]

D. S. Lee and T. Lemieux. Regression discontinuity designs in economics. *Journal of Economic Literature*, 48(2):281–355, 2010. [p38]

J. Porter. Estimation in the regression discontinuity model. Working paper, University of Wisconsin, 2003. [p39]

StataCorp. *Stata Data Analysis Statistical Software: Release 13*. StataCorp LP, College Station, TX, 2013. URL http://www.stata.com/. [p38]

M. Wand and M. Jones. *Kernel Smoothing*. Chapman & Hall/CRC, Florida, 1995. [p41]

Sebastian Calonico
Department of Economics
University of Miami
Coral Gables, Florida
USA
scalonico@bus.miami.edu

Matias D. Cattaneo
Department of Economics
University of Michigan
Ann Arbor, Michigan
USA
cattaneo@umich.edu

Rocío Titiunik
Department of Political Science
University of Michigan
Ann Arbor, Michigan
USA
titiunik@umich.edu

scalonico@bus.miami.edu