Robust data-driven inference in the regression-discontinuity design

Sebastian Calonico
University of Miami
Coral Gables, FL
scalonico@bus.miami.edu

Matias D. Cattaneo
University of Michigan
Ann Arbor, MI
cattaneo@umich.edu

Rocío Titiunik
University of Michigan
Ann Arbor, MI
titiunik@umich.edu

Abstract. In this article, we introduce three commands to conduct robust data-driven statistical inference in regression-discontinuity (RD) designs. First, we present `rdrobust`, a command that implements the robust bias-corrected confidence intervals proposed in Calonico, Cattaneo, and Titiunik (2014d, *Econometrica* 82: 2295–2326) for average treatment effects at the cutoff in sharp RD, sharp kink RD, fuzzy RD, and fuzzy kink RD designs. This command also implements other conventional nonparametric RD treatment-effect point estimators and confidence intervals. Second, we describe the companion command `rdbwselect`, which implements several bandwidth selectors proposed in the RD literature. Following the results in Calonico, Cattaneo, and Titiunik (2014a, Working paper, University of Michigan), we also introduce `rdplot`, a command that implements several data-driven choices of the number of bins in evenly spaced and quantile-spaced partitions that are used to construct the RD plots usually encountered in empirical applications. A companion R package is described in Calonico, Cattaneo, and Titiunik (2014b, Working paper, University of Michigan).

Keywords: st0366, rdrobust, rdbwselect, rdplot, regression discontinuity (RD), sharp RD, sharp kink RD, fuzzy RD, fuzzy kink RD, treatment effects, local polynomials, bias correction, bandwidth selection, RD plots

1 Introduction

The regression-discontinuity (RD) design is a well-established and widely used research design in empirical work. In this design, units receive treatment on the basis of whether their value of an observed covariate is above or below a known cutoff. The key feature of the design is that the probability of receiving treatment conditional on this covariate jumps discontinuously at the cutoff, inducing "variation" in treatment assignment that is assumed to be unrelated to potential confounders. Because of 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 (an exception being Cattaneo, Frandsen, and Titiunik [Forthcoming]).[1]

In this article, we discuss data-driven (that is, fully automatic) local-polynomial-based robust inference procedures in the RD design. We introduce three main commands

---

1. For review and further discussion, see Hahn, Todd, and Van Der Klaauw (2001), Porter (2003), Lee (2008), Imbens and Lemieux (2008), Lee and Lemieux (2010), Dinardo and Lee (2011), and Imbens and Kalyanaraman (2012).

that together offer an array of data-driven nonparametric inference procedures useful for RD empirical applications, including point estimators and confidence intervals (CIs). First, our main command, `rdrobust`, implements the bias-corrected inference procedure proposed by Calonico, Cattaneo, and Titiunik (2014d), which is robust to "large" bandwidth choices. This command also implements other classical inference procedures using local polynomial regression, as suggested in the recent econometrics literature (see references in footnote 1). This implementation offers robust bias-corrected CIs for average treatment effects at the cutoff for sharp RD, sharp kink RD, fuzzy RD, and fuzzy kink RD designs, among other possibilities.

To construct these nonparametric estimators and CIs, all of which are based on estimating a regression function in a neighborhood of the cutoff, we need one or more choices of bandwidth. Our second command, `rdbwselect`, which is called by `rdrobust`, provides different data-driven bandwidth selectors based on the recent work of Imbens and Kalyanaraman (2012) and Calonico, Cattaneo, and Titiunik (2014d). Although this command can 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 `rdrobust`.

Finally, our third command, `rdplot`, implements the results described in Calonico, Cattaneo, and Titiunik (2014a), offering several data-driven choices of the number of bins in evenly spaced and quantile-spaced RD plots. These plots are derived either to approximate the regression function with local sample averages of the outcome variable within bins of the running variable or to depict the overall variability of the data in a disciplined and objective way. These selectors are obtained by approximating the asymptotic integrated variance and bias of certain partitioning-type estimators (see Cattaneo and Farrell [2013] for references) and are implemented nonparametrically using spacings and series estimators. We show how the results implemented in `rdplot` can be used to construct the RD plots commonly found in empirical applications. This command offers a fully automatic way of constructing several useful RD plots that can be used for both presenting the data and falsifying the design.

The rest of this article is organized as follows. In section 2, we review the methods implemented in our commands. In sections 3, 4, and 5, we describe the syntax of `rdrobust`, `rdbwselect`, and `rdplot`, respectively. In section 6, we offer an empirical illustration of our commands by investigating party advantages in U.S. Senate elections using the data and research design in Cattaneo, Frandsen, and Titiunik (Forthcoming). We illustrate several of the main features of our commands, including their compatibility with `outreg2` (Wada 2005). Finally, in section 7, we conclude. A companion R package is described in Calonico, Cattaneo, and Titiunik (2014b).

2   Review of methods

In this section, we review the methods implemented in our commands `rdrobust`, `rdbwselect`, and `rdplot`. To avoid distractions and technicalities, we do not present the regularity conditions and technical discussions underlying these methods herein, but these can be found in the references below. Also, to simplify the discussion, we focus

on the special case of the sharp RD design, where the probability of treatment changes deterministically from zero to one at the cutoff. However, we also cover sharp kink RD, fuzzy RD, and fuzzy kink RD designs. We discuss the implementation for the latter cases briefly in section 2.6 (see also sections 3 and 4 for the corresponding syntax), but we refer the reader to Calonico, Cattaneo, and Titiunik (2014d; 2014c) for further details.

For recent reviews on classical inference approaches in the RD design and comprehensive lists of empirical examples, see Imbens and Lemieux (2008), Lee and Lemieux (2010), Dinardo and Lee (2011), and references therein. Here we focus on approaches using local polynomial nonparametric estimators with data-driven bandwidth selectors and bias-correction techniques, also following the recent results in Imbens and Kalyanarman (2012) and Calonico, Cattaneo, and Titiunik (2014d) and their supplemental appendix (Calonico, Cattaneo, and Titiunik 2014c).

2.1 Setup and notation

We focus on large-sample inference for the average treatment effect at the cutoff in the sharp RD design. For each unit i, the scalar random variable Y_i denotes the outcome of interest, while the scalar regressor X_i is the "running variable" or "score" that determines treatment assignment based on a known cutoff.

We adopt the potential-outcomes framework commonly used in the treatment-effects literature (for example, Heckman and Vytlacil [2007]; Imbens and Wooldridge [2009]). Let $[\{Y_i(0), Y_i(1), X_i\}' : i = 1, 2, \ldots, n]$ be a random sample from $\{Y(0), Y(1), X\}'$, where $Y(1)$ and $Y(0)$ denote the potential outcomes with and without treatment, respectively, and treatment assignment is determined by the following known rule: unit i is assigned to the treatment condition if $X_i \geqslant \overline{x}$ and is assigned to the control condition if $X_i < \overline{x}$ for some known fixed value $\overline{x}$. Thus the observed outcome is

$$Y_i = \left\{ \begin{array}{ll} Y_i(0) & \text{if } X_i < \overline{x} \\ Y_i(1) & \text{if } X_i \geqslant \overline{x} \end{array} \right.$$

and the observed random sample is $\{(Y_i, X_i)' : i = 1, 2, \ldots, n\}$.

We discuss and implement several data-driven inference procedures for the sharp average treatment effect at the threshold, which is given by

$$\tau = \mathbb{E}\{Y_i(1) - Y_i(0)|X_i = \overline{x}\}$$

This popular estimand in the RD literature is nonparametrically identifiable under mild continuity conditions (Hahn, Todd, and Van Der Klaauw 2001). Specifically,

$$\tau = \mu_+ - \mu_-$$

with

$$\mu_+ = \lim_{x \downarrow \overline{x}} \mu(x), \quad \mu_- = \lim_{x \uparrow \overline{x}} \mu(x), \quad \mu(x) = \mathbb{E}(Y_i|X_i = x)$$

where here, and elsewhere in the article, we drop the evaluation point of functions whenever possible to simplify notation.

Following Hahn, Todd, and Van Der Klaauw (2001) and Porter (2003), we construct a popular estimator of τ by using kernel-based local polynomials on either side of the threshold. These regression estimators are particularly well suited for inference in the RD design because of their good properties at the boundary of the support of the regression function—see Fan and Gijbels (1996) for more details. The local polynomial RD estimator of order p is

$$\widehat{\tau}_p(h_n) = \widehat{\mu}_{+,p}(h_n) - \widehat{\mu}_{-,p}(h_n)$$

where $\widehat{\mu}_{+,p}(h_n)$ and $\widehat{\mu}_{-,p}(h_n)$ denote the intercept (at $\overline{x}$) of a weighted pth-order polynomial regression for only treated and only control units, respectively. More precisely,

$$\widehat{\mu}_{+,p}(h_n) = \mathbf{e}_0'\widehat{\boldsymbol{\beta}}_{+,p}(h_n) \quad \text{and} \quad \widehat{\mu}_{-,p}(h_n) = \mathbf{e}_0'\widehat{\boldsymbol{\beta}}_{-,p}(h_n)$$

with

$$\widehat{\boldsymbol{\beta}}_{+,p}(h_n) = \arg\min_{\boldsymbol{\beta}\in\mathbb{R}^{p+1}} \sum_{i=1}^{n} \mathbb{1}(X_i \geqslant \overline{x})\{Y_i - \mathbf{r}_p(X_i - \overline{x})'\boldsymbol{\beta}\}^2 K_{h_n}(X_i - \overline{x})$$

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

where $\mathbf{r}_p(x) = (1, x, \ldots, x^p)'$, $\mathbf{e}_0 = (1, 0, \ldots, 0) \in \mathbb{R}^{p+1}$ is the first unit vector, $K_h(u) = K(u/h)/h$ with $K()$ a kernel function, h_n is a positive bandwidth sequence, and $\mathbb{1}()$ denotes the indicator function.

Under simple regularity conditions, and assuming the bandwidth h_n vanishes at an appropriate rate, local polynomial estimators are known to satisfy

$$\widehat{\boldsymbol{\beta}}_{+,p}(h_n) \to_p \boldsymbol{\beta}_{+,p} \quad \text{and} \quad \widehat{\boldsymbol{\beta}}_{-,p}(h_n) \to_p \boldsymbol{\beta}_{-,p}$$

with

$$\boldsymbol{\beta}_{+,p} = \left(\mu_+, \mu_+^{(1)}, \frac{\mu_+^{(2)}}{2}, \ldots, \frac{\mu_+^{(p)}}{p!}\right)'$$

$$\boldsymbol{\beta}_{-,p} = \left(\mu_-, \mu_-^{(1)}, \frac{\mu_-^{(2)}}{2}, \ldots, \frac{\mu_-^{(p)}}{p!}\right)'$$

$$\mu_+^{(s)} = \lim_{x\downarrow\overline{x}} \frac{\partial^s}{\partial x^s}\mu_+(x)$$

$$\mu_+(x) = \mathbb{E}\{Y(1)|X_i = x\}$$

$$\mu_-^{(s)} = \lim_{x\uparrow\overline{x}} \frac{\partial^s}{\partial x^s}\mu_-(x)$$

$$\mu_-(x) = \mathbb{E}\{Y(0)|X_i = x\}$$

$s = 1, 2, \ldots, p$, thereby offering a family of consistent estimators of τ. Among these possible estimators, the local linear RD estimator $\widehat{\tau}_1(h_n)$ is perhaps the preferred and most common choice in practice.

2.2 Overview of upcoming discussion

We now survey the results presented in the following sections to help the reader easily identify the conceptual differences between the estimators implemented by `rdrobust`. In particular, in sections 2.3 and 2.4, we review some of the salient asymptotic properties of RD treatment-effect estimators $\widehat{\tau}_p(h_n)$, which are based on local polynomial nonparametric estimators. We outline our discussion in these sections as follows:

1. We assess some of the main properties of $\widehat{\tau}_p(h_n)$ as point estimators in the first part of section 2.3. Specifically, we discuss a (conditional) mean-squared error (MSE) expansion of $\widehat{\tau}_p(h_n)$ that highlights its variance and bias properties. We also use this expansion to summarize some bandwidth-selection approaches tailored to minimize the leading terms in the asymptotic MSE expansion, including plug-in rules and a cross-validation (CV) approach.

2. In the rest of section 2.3, we discuss the construction of asymptotically valid CIs based on $\widehat{\tau}_p(h_n)$ for the sharp mean treatment effect τ. In particular, we discuss two distinct approaches: one based on "undersmoothing" and the other based on "bias correction".

   The first approach, which is arguably the most commonly used in practice, assumes away the bias of the estimator and constructs $100(1-\alpha)$ percent CIs of the form

   $$\mathrm{CI}_{1-\alpha,n} = \left\{ \widehat{\tau}_p(h_n) \pm \Phi_{1-\frac{\alpha}{2}}^{-1} \sqrt{\widehat{\mathsf{v}}_n} \right\}$$

   where Φ_a^{-1} denotes the appropriate quantile of the Gaussian distribution (for example, 1.96 for $a = 0.975$), and $\widehat{\mathsf{v}}_n$ denotes an appropriate choice of variance estimator. This approach is theoretically justified only if the (smoothing) leading bias of the RD estimator is "small", which requires some form of "undersmoothing"; that is, it requires choosing a "smaller" bandwidth than the MSE-optimal one. In practice, researchers typically use the same bandwidth used to construct the RD point estimator $\widehat{\tau}_p(h_n)$, thereby ignoring the potential effects of the leading bias on the performance of these CIs.

   A second approach to constructing CIs is to use bias correction. This approach is conventional in the nonparametric literature, although it is not commonly used in empirical work, because it is regarded as having inferior finite-sample properties. The resulting CIs in this case take the form

   $$\mathrm{CI}_{1-\alpha,n}^{\mathrm{bc}} = \left[\left\{ \widehat{\tau}_p(h_n) - \widehat{b}_n \right\} \pm \Phi_{1-\frac{\alpha}{2}}^{-1} \sqrt{\widehat{\mathsf{v}}_n} \right]$$

   where here the only difference is the bias-estimate $\widehat{b}_n$, which is introduced with the explicit goal of removing the potentially large effects of the unknown leading bias of the RD estimator, $\widehat{\tau}_p(h_n)$. This second approach to constructing CIs justifies using MSE-optimal bandwidth choices when constructing the estimators.

3. The results in Calonico, Cattaneo, and Titiunik (2014d) offer alternative CIs based on bias-corrected local polynomials, which take the form

$$\text{CI}_{1-\alpha,n}^{\text{rbc}} = \left[\left\{ \widehat{\tau}_p(h_n) - \widehat{b}_n \right\} \pm \Phi_{1-\frac{\alpha}{2}}^{-1} \sqrt{\widehat{v}_n^{\text{bc}}} \right]$$

where the key difference between the "conventional" bias-corrected (CI^{bc}) and these alternative robust CIs (CI^{rbc}) is the presence of a different variance estimator, denoted here by $\widehat{v}_n^{\text{bc}}$. This new variance formula is theoretically derived by using an alternative asymptotic approximation to the bias-corrected RD estimator $\widehat{\tau}_p(h_n) - \widehat{b}_n$. The resulting CIs have some attractive theoretical properties and, as we discuss in more detail below, allow for the use of MSE-optimal bandwidth choices while offering excellent finite-sample performance. In section 2.4, we offer a heuristic discussion of these results, which are developed by Calonico, Cattaneo, and Titiunik (2014d) and are the main motivation for the development of our package.

4. In section 2.5, we summarize all the statistical procedures available in our commands for sharp RD designs. In section 2.6, we briefly explain how these results are extended to sharp kink, fuzzy, and fuzzy kink RD designs, among other possibilities.

5. Finally, in section 2.7, we discuss several fully automatic approaches for constructing the plots commonly shown in RD applications. Specifically, we implement the results in Calonico, Cattaneo, and Titiunik (2014a), which offer optimal choices of bin length for evenly spaced and quantile-spaced partitioning schemes for constructing local sample means for control and treatment units. These local sample means help to approximate the underlying regression functions and are usually plotted together with global polynomial estimates to summarize the empirical features of the RD design.

2.3   Conventional RD inference

Point estimators

Under appropriate regularity conditions, the treatment-effect estimator $\widehat{\tau}_p(h_n)$ admits the following MSE expansion. Let $\mathbf{X}_n = (X_1, X_2, \ldots, X_n)'$.

$$\text{MSE}_p(h_n) = \mathbb{E}\left[\left\{ \widehat{\tau}_p(h_n) - \tau \right\}^2 | \mathbf{X}_n \right] \approx h_n^{2(p+1)} B_{n,p}^2 + \frac{1}{nh_n} V_{n,p}$$

with

$$B_{n,p} \to_p B_p \quad \text{and} \quad V_{n,p} \to_p V_p$$

where $B_{n,p}$ and $V_{n,p}$ represent, respectively, the leading asymptotic bias and the asymptotic variance of $\widehat{\tau}_p(h_n)$. The exact form of $B_{n,p}$ and $V_{n,p}$, and their asymptotic counterparts, can be found in Calonico, Cattaneo, and Titiunik (2014d). This treatment-effect

estimator will be consistent if $h_n \to 0$ and $nh_n \to \infty$. Moreover, the point estimator $\widehat{\tau}_p(h_n)$ will be optimal in an asymptotic MSE sense if the bandwidth h_n is chosen so that

$$h_{\mathrm{MSE},n,p} = \left\{ \frac{V_p}{2(p+1)B_p^2} \right\}^{\frac{1}{(2p+3)}} n^{\frac{-1}{(2p+3)}}$$

whenever $B_p \neq 0$. This last assumption may be restrictive because $B_p \propto \mu_+^{(p+1)} - \mu_-^{(p+1)}$ may be (close to) 0 in some applications (see Imbens and Kalyanaraman [2012]; see them also for a recent review on bandwidth selection in the RD design).

Imbens and Kalyanaraman (2012) use this reasoning in providing a data-driven, asymptotically MSE-optimal, RD treatment-effect estimator. Specifically, they propose a more "robust" consistent bandwidth estimator of the form

$$\widehat{h}_{\mathrm{IK},n,p} = \left\{ \frac{\widehat{V}_{\mathrm{IK},p}}{2(p+1)\widehat{B}_{\mathrm{IK},p}^2 + \widehat{R}_{\mathrm{IK},p}} \right\}^{\frac{1}{(2p+3)}} n^{\frac{-1}{(2p+3)}}$$

where the additional (regularization) term $\widehat{R}_{\mathrm{IK},p}$ is introduced to avoid small denominators in moderate-size samples. Here $\widehat{B}_{\mathrm{IK},p}$ and $\widehat{V}_{\mathrm{IK},p}$ (and $\widehat{R}_{\mathrm{IK},p}$) are nonparametric consistent estimators of their respective population counterparts, which require the choice of preliminary bandwidths, generically denoted by b_n herein. Imbens and Kalyanaraman (2012) provide a direct implementation approach for $p = 1$, but the preliminary bandwidths used in their construction are not optimally chosen. Thus $\widehat{h}_{\mathrm{IK},n,p}$ can be viewed as a nonparametric first-generation plug-in rule (for example, Wand and Jones [1995]), sometimes denoted by a direct plug-in rule of order 1.

Motivated by the work of Imbens and Kalyanaraman (2012), Calonico, Cattaneo, and Titiunik (2014d) propose an alternative second-generation plug-in bandwidth selection approach. Specifically, they propose the following second-order direct plug-in rule:

$$\widehat{h}_{\mathrm{CCT},n,p} = \left\{ \frac{\widehat{V}_{\mathrm{CCT},p}}{2(p+1)\widehat{B}_{\mathrm{CCT},p}^2 + \widehat{R}_{\mathrm{CCT},p}} \right\}^{\frac{1}{(2p+3)}} n^{\frac{-1}{(2p+3)}}$$

This alternative bandwidth estimator has two distinct features relative to $\widehat{h}_{\mathrm{IK},n,p}$. First, not only are the estimators $\widehat{V}_{\mathrm{CCT},p}$ and $\widehat{B}_{\mathrm{CCT},p}$ (and $\widehat{R}_{\mathrm{CCT},p}$) consistent for their population counterparts, but the preliminary bandwidths used in their constructions are consistent estimators of the corresponding population MSE-optimal bandwidths. In this sense, $\widehat{h}_{\mathrm{CCT},n,p}$ is a direct plug-in rule of order 2. Second—motivated by finite-sample performance considerations—Calonico, Cattaneo, and Titiunik (2014d) construct an alternative estimator of V_p (denoted by $\widehat{V}_{\mathrm{CCT},p}$ above) that does not require an additional choice of bandwidth for its construction but, as in the work of Abadie and Imbens (2006), relies instead on a fixed-matches nearest-neighbor-based "estimate" of the residuals. This construction, as well as other more traditional approaches, is discussed further below because the term V_p plays a crucial role when forming CIs for τ.

The main bandwidth, $h_{n,p}$, can be chosen in other ways. A popular alternative is to use cross-validation, as done by Ludwig and Miller (2007). As discussed in Imbens and Kalyanaraman (2012), one such bandwidth-selection approach can be described as follows,

$$\widehat{h}_{\mathrm{CV},n,p} = \arg \min_{h>0} \mathrm{CV}_\delta(h), \quad \mathrm{CV}_\delta(h) = \sum_{i=1}^n \mathbb{1}(X_{-,[\delta]} \leqslant X_i \leqslant X_{+,[\delta]}) \{Y_i - \widehat{\mu}_p(X_i; h)\}^2$$

where

$$\widehat{\mu}_p(x; h) = \begin{cases} \mathbf{e}_0' \widehat{\boldsymbol{\beta}}_{+,p}(x,h) & \text{if } x \geqslant \overline{x} \\ \mathbf{e}_0' \widehat{\boldsymbol{\beta}}_{-,p}(x,h) & \text{if } x < \overline{x} \end{cases}$$

$$\widehat{\boldsymbol{\beta}}_{+,p}(x,h) = \arg \min_{\boldsymbol{\beta} \in \mathbb{R}^{p+1}} \sum_{i=1}^n \mathbb{1}(X_i \geqslant x) \{Y_i - \mathbf{r}_p(X_i - x)' \boldsymbol{\beta}\}^2 K_h(X_i - x)$$

$$\widehat{\boldsymbol{\beta}}_{-,p}(x,h) = \arg \min_{\boldsymbol{\beta} \in \mathbb{R}^{p+1}} \sum_{i=1}^n \mathbb{1}(X_i < x) \{Y_i - \mathbf{r}_p(X_i - x)' \boldsymbol{\beta}\}^2 K_h(X_i - x)$$

and, for $\delta \in (0,1)$, $X_{-,[\delta]}$ and $X_{+,[\delta]}$ denote the δth quantile of $\{X_i : X_i < \overline{x}\}$ and $\{X_i : X_i \geqslant \overline{x}\}$, respectively. Our bandwidth-selection command also implements this approach for completeness.

To summarize, the results discussed so far justify the following three data-driven RD treatment-effect point estimators:

$$\widehat{\tau}_p\left(\widehat{h}_{\mathrm{IK},n,p}\right), \qquad \widehat{\tau}_p\left(\widehat{h}_{\mathrm{CCT},n,p}\right), \qquad \text{and} \qquad \widehat{\tau}_p\left(\widehat{h}_{\mathrm{CV},n,p}\right)$$

Under appropriate conditions, these estimators may be interpreted as consistent and (asymptotically) MSE-optimal point estimators of τ.

CIs: Asymptotic distribution

Under appropriate regularity conditions and rate restrictions on the bandwidth sequence $h_n \to 0$, conventional CIs accompanying the point estimators discussed above rely on the following distributional approximation,

$$\sqrt{nh_n}\left\{\widehat{\tau}_p(h_n) - \tau - h_n^{p+1} B_{n,p}\right\} \to_d N(0, V_p), \quad V_p \propto \frac{\sigma_+^2 + \sigma_-^2}{f}$$

where $\sigma_+^2 = \lim_{x \to \overline{x}^+} \sigma^2(x)$ and $\sigma_-^2 = \lim_{x \to \overline{x}^-} \sigma^2(x)$ with $\sigma^2(x) = \mathbb{V}(Y_i | X_i = x)$, and $f = f(\overline{x})$ with $f(x)$ the density of X. Therefore, an infeasible asymptotic $100(1 - \alpha)$-percent CI for τ is given by

$$\mathrm{CI}_{1-\alpha}^*(h_n) = \left[\left\{\widehat{\tau}_p(h_n) - h_n^{p+1} B_{n,p}\right\} \pm \Phi_{1-\alpha/2}^{-1} \sqrt{\frac{V_p}{nh_n}}\right]$$

To implement this CI, we need to handle the leading bias $(B_{n,p})$ and the variance (V_p) of the RD estimator because they involve unknown quantities. We discuss these problems and related practical issues in the following subsections.

CIs: Variance estimators

The asymptotic variance is handled by replacing V_p with a consistent estimator. A natural approach uses the conditional (on $\mathbf{X}_n$) variance of $\widehat{\tau}_p(h_n)$ as a starting point because, as mentioned above, $V_{n,p} \to_{\mathrm{p}} V_p$. Here we have

$$V_{n,p} = nh_n\mathbb{V}\{\widehat{\tau}_p(h_n)|\mathbf{X}_n\} = V_{+,n,p} + V_{-,n,p}$$

with

$$V_{+,n,p} = h_n\mathbf{e}_0'\boldsymbol{\Gamma}_{+,n,p}^{-1}\mathbf{X}_{+,n,p}\mathbf{W}_{+,n,p}\boldsymbol{\Sigma}\mathbf{W}_{+,n,p}\mathbf{X}_{+,n,p}\boldsymbol{\Gamma}_{+,n,p}^{-1}\mathbf{e}_0$$

$$V_{-,n,p} = h_n\mathbf{e}_0'\boldsymbol{\Gamma}_{-,n,p}^{-1}\mathbf{X}_{-,n,p}\mathbf{W}_{-,n,p}\boldsymbol{\Sigma}\mathbf{W}_{-,n,p}\mathbf{X}_{-,n,p}\boldsymbol{\Gamma}_{-,n,p}^{-1}\mathbf{e}_0$$

where the exact form of these matrices is discussed in Calonico, Cattaneo, and Titiunik (2014d). Importantly, the only matrix including unknown quantities is

$$\boldsymbol{\Sigma} = \begin{bmatrix} \sigma^2(X_1) & 0 & \cdots & 0 \\ 0 & \sigma^2(X_2) & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \sigma^2(X_n) \end{bmatrix} = \mathbb{E}(\boldsymbol{\varepsilon}\boldsymbol{\varepsilon}'|\mathbf{X}_n)$$

where $\boldsymbol{\varepsilon} = (\varepsilon_1, \varepsilon_2, \ldots, \varepsilon_n)'$ with $\varepsilon_i = Y_i - \mu(X_i)$.

The "sandwich" structure of $V_{n,p}$ arises naturally from the weighted least-squares form of local polynomials, resembling the usual heteroskedasticity-robust standard-error formula in linear regression models. Therefore, and just as in linear models, implementing these standard-error estimators requires only an estimator of $\boldsymbol{\Sigma}$, which reduces the problem to plugging in an estimator of $\sigma^2(X_i) = \mathbb{E}(\varepsilon_i^2|X_i)$ for control and treatment units separately. We consider two approaches to construct such estimators: "plug-in estimated residuals" and "fixed-matches estimated residuals". Both approaches construct an estimator of $V_{n,p}$ by removing the conditional expectation in $\boldsymbol{\Sigma}$ and replacing ε_i by some estimator of it.

Plug-in Estimated Residuals. This approach follows the standard linear models logic underlying local polynomial estimators and thus replaces ε_i by

$$\breve{\varepsilon}_{+,i} = Y_i - \widehat{\mu}_{+,p}(X_i; c_n) \qquad \text{and} \qquad \breve{\varepsilon}_{-,i} = Y_i - \widehat{\mu}_{-,p}(X_i; c_n)$$

for treated and control units, respectively, where $\widehat{\mu}_{+,p}(X_i; c_n) = \mathbf{e}_0'\widehat{\boldsymbol{\beta}}_{+,p}(x, c_n)$ and $\widehat{\mu}_{-,p}(X_i; c_n) = \mathbf{e}_0'\widehat{\boldsymbol{\beta}}_{-,p}(x, c_n)$, and c_n denotes the bandwidth used. In practice, c_n is usually chosen to be $c_n = h_n$, even though this choice may not be optimal and could lead to poor finite-sample performance of the estimators. The resulting variance formula becomes the familiar Huber–Eicker–White standard-error estimator, which is robust to heteroskedasticity of unknown form. We denote this estimator by

$$\breve{V}_{n,p} = \breve{V}_{+,n,p} + \breve{V}_{-,n,p}$$

where $\breve{V}_{+,n,p}$ and $\breve{V}_{-,n,p}$ use, respectively, $\breve{\varepsilon}_{+,i}$ and $\breve{\varepsilon}_{-,i}$ in their construction.

Fixed-matches Estimated Residuals. The previous construction is intuitive and easy to implement, but it requires the additional choice of the bandwidth c_n to construct the estimates of the residuals. Employing the same bandwidth choice used to construct the RD treatment-effect estimator may not lead to a variance estimator with good finite-sample properties. Calonico, Cattaneo, and Titiunik (2014d) propose a variance estimator using a different construction for the residuals as an alternative motivated by the recent work of Abadie and Imbens (2006). This estimator is constructed using a simple fixed-matches estimator for the residuals, denoted by $\widehat{\varepsilon}_{+,i}$ and $\widehat{\varepsilon}_{-,i}$ for treatment and control, respectively, which are unbiased but inconsistent. Nonetheless, the resulting variance estimators are consistent under appropriate regularity conditions (see Calonico, Cattaneo, and Titiunik [2014d] for more details). We denote this estimator by

$$\widehat{V}_{n,p} = \widehat{V}_{+,n,p} + \widehat{V}_{-,n,p}$$

where $\widehat{V}_{+,n,p}$ and $\widehat{V}_{-,n,p}$ use, respectively, the fixed-matches estimators $\widehat{\varepsilon}_{+,i}$ and $\widehat{\varepsilon}_{-,i}$ in their construction.

In summary, two alternative variance estimators are i) the plug-in estimated residuals estimator $\check{V}_{n,p}$ and ii) the fixed-matches estimator $\widehat{V}_{n,p}$, both satisfying

$$\check{V}_{n,p} \to_p V_p \quad \text{and} \quad \widehat{V}_{n,p} \to_p V_p$$

For implementation, we use the same estimated bandwidth used in the treatment-effect estimator $\widehat{\tau}_p(h_n)$ whenever needed.

CIs: Asymptotic bias

There are two main approaches to handle the leading bias $(B_{n,p})$ in the infeasible CIs, $\text{CI}^*_{1-\alpha}(h_n)$: "undersmoothing" (alternatively, assuming the bias is "small") or "bias correction". We briefly discuss each of these ideas below.

Undersmoothing. The first approach is to "undersmooth" the estimator, that is, to choose a "small" enough bandwidth so that the bias is negligible. Theoretically, this approach simply requires selecting a bandwidth sequence $h_n \to 0$ such that

$$\sqrt{nh_n}\left\{\widehat{\tau}_p(h_n) - \tau - h_n^{p+1}\,B_{n,p}\right\} = \sqrt{nh_n}\{\widehat{\tau}_p(h_n) - \tau\} + o_p(1) \to_d N(0, V_p)$$

which is justified whenever the bandwidth h_n vanishes fast enough. In practice, however, this procedure may be difficult to implement because most bandwidth selectors—such as $h_{\text{MSE},n,p}$—will not satisfy the conditions required for undersmoothing. This fact implies that most empirical bandwidth selectors could lead to a nonnegligible leading bias in the distributional approximation, which will bias the associated CIs. Simulation evidence highlighting this potential drawback of the undersmoothing (small-bias) approach is provided in Calonico, Cattaneo, and Titiunik (2014d).

Nonetheless, in applications, it is common for researchers to simply ignore the leading bias and proceed as if $B_{n,p} \approx 0$. This approach is justified by either assuming the bias is "small" or by shrinking the bandwidth choice by some ad hoc factor (that

is, undersmoothing). We give the exact formula of the resulting CIs justified by this approach in the next subsection.

Bias Correction. The second approach to deal with the leading bias in the distributional approximation is to bias correct the estimator—that is, to construct an estimator of $B_{n,p}$, which is then subtracted from the point estimate to eliminate this leading bias.

A simple way to implement this idea is to use a higher-order local polynomial to estimate the unknown derivatives in the leading bias term. Recall that $B_p \propto \mu_+^{(p+1)} - \mu_-^{(p+1)}$. For example, $\mu_+^{(p+1)}$ can be estimated by using a qth-order local polynomial $(q \geqslant p+1)$ with pilot bandwidth b_n, leading to the estimator $\hat{\mu}_+^{(p+1)} = \mathbf{e}_{p+1}' \hat{\boldsymbol{\beta}}_{+,q}(b_n)$, where $\mathbf{e}_j$ is the conformable $(j+1)$th-unit vector—for example, $\mathbf{e}_1 = (0, 1, 0, 0)'$ if $q = 3$. This is the approach we use herein to construct a bias estimate $\hat{B}_{n,p,q}$, which depends on a preliminary bandwidth choice b_n. The resulting bias-corrected estimator is

$$\hat{\tau}_{p,q}^{\mathrm{bc}}(h_n, b_n) = \hat{\tau}_p(h_n) - h_n^{p+1}\,\hat{B}_{n,p,q}, \qquad \hat{B}_{n,p,q} = \hat{B}_{n,p,q}(h_n, b_n), \qquad p < q$$

where the exact formula of $\hat{B}_{n,p,q}(h_n, b_n)$ is given in Calonico, Cattaneo, and Titiunik (2014d).

Using this bias-corrected estimator and imposing appropriate regularity conditions and bandwidth restrictions, we obtain

$$\sqrt{nh_n}\left\{\hat{\tau}_{p,q}^{\mathrm{bc}}(h_n, b_n) - \tau\right\} = \underbrace{\sqrt{nh_n}\left\{\hat{\tau}_p(h_n) - \tau - h_n^{p+1}\,B_{n,p}\right\}}_{\to_d\,\mathcal{N}(0, \mathrm{V}_p)}$$
$$- \underbrace{\sqrt{nh_n}h_n^{p+1}\left(\hat{B}_{n,p,q} - B_{n,p}\right)}_{\to_p\,0}$$

which is valid if

$$\frac{h_n}{b_n} \to 0$$

This result immediately justifies bias-corrected CIs, where the unknown bias in $\mathrm{CI}_{1-\alpha}^*(h_n)$ is replaced by the bias estimate $\hat{B}_{n,p,q}$. We give the exact formula of the resulting CIs in the upcoming subsection.

To implement this approach, we must make a choice of pilot bandwidth b_n. As discussed above, bandwidth choices can be constructed using asymptotic MSE expansions for the appropriate estimators. This is the approach followed by Calonico, Cattaneo, and Titiunik (2014d), who propose the following MSE-optimal choice of pilot bandwidth b_n for the bias-correction estimator $\hat{B}_{n,p,q}$. This optimal pilot bandwidth is given by

$$b_{\mathrm{MSE},n,p,q} = \left\{\frac{(2p+3)\,V_{p+1,q}}{2(q-p)B_{p+1,q}^2}\right\}^{\frac{1}{(2q+3)}} n^{\frac{-1}{(2q+3)}}$$

where $V_{p,q}$ and $B_{p,q}$ are the corresponding leading variance and bias terms arising from the MSE expansion used. [Note that this choice is not necessarily optimal for

$\widehat{\tau}_{p,q}^{\mathrm{bc}}(h_n, b_n)$.] Calonico, Cattaneo, and Titiunik (2014d) discuss a relatively simple implementation procedure of $b_{\mathrm{MSE},n,p,q}$, which leads to the following data-driven estimator:

$$\widehat{b}_{\mathrm{CCT},n,p,q} = \left\{ \frac{(2p+3)\widehat{V}_{\mathrm{CCT},p+1,q}}{2(q-p)\widehat{B}_{\mathrm{CCT},p+1,q}^2 + \widehat{R}_{\mathrm{CCT},p+1,q}} \right\}^{\frac{1}{(2q+3)}} n^{\frac{-1}{(2q+3)}}$$

They also provide the exact form of the estimators $\widehat{V}_{\mathrm{CCT},p,q}$, $\widehat{B}_{\mathrm{CCT},p,q}$, and $\widehat{R}_{\mathrm{CCT},p,q}$. We use this pilot bandwidth estimator in our default implementation, but we also implement similar estimators constructed following the underlying logic in Imbens and Kalyanaraman (2012), denoted by $\widehat{b}_{\mathrm{IK},n,p,q}$.

CIs: Summary of classical approaches

The results discussed so far suggest the following data-driven RD treatment-effect CIs:

- *Undersmoothing (Small Bias)*:
  - *Plug-in estimated errors*: $\breve{\mathrm{CI}}_{1-\alpha}\left(\widehat{h}_{\mathrm{IK},n,p}\right)$, $\breve{\mathrm{CI}}_{1-\alpha}(\widehat{h}_{\mathrm{CCT},n,p})$, and $\breve{\mathrm{CI}}_{1-\alpha}(\widehat{h}_{\mathrm{CV},n,p})$, where

    $$\breve{\mathrm{CI}}_{1-\alpha}(h_n) = \left\{ \widehat{\tau}_p(h_n) \pm \Phi_{1-\frac{\alpha}{2}}^{-1} \sqrt{\frac{\breve{V}_p}{n h_n}} \right\}$$

  - *Fixed-matches estimated errors*: $\widehat{\mathrm{CI}}_{1-\alpha}(\widehat{h}_{\mathrm{IK},n,p})$, $\widehat{\mathrm{CI}}_{1-\alpha}(\widehat{h}_{\mathrm{CCT},n,p})$, and $\widehat{\mathrm{CI}}_{1-\alpha}(\widehat{h}_{\mathrm{CV},n,p})$, where

    $$\widehat{\mathrm{CI}}_{1-\alpha}(h_n) = \left\{ \widehat{\tau}_p(h_n) \pm \Phi_{1-\frac{\alpha}{2}}^{-1} \sqrt{\frac{\widehat{V}_p}{n h_n}} \right\}$$

- *Bias Correction*:
  - *Plug-in estimated errors*: $\breve{\mathrm{CI}}_{1-\alpha}^{\mathrm{bc}}(\widehat{h}_n, \widehat{b}_n)$, where

    $$\breve{\mathrm{CI}}_{1-\alpha}^{\mathrm{bc}}(h_n, b_n) = \left[\left\{ \widehat{\tau}_p(h_n) - h_n^{p+1} \, \widehat{B}_{n,p,q} \right\} \pm \Phi_{1-\frac{\alpha}{2}}^{-1} \sqrt{\frac{\breve{V}_p}{n h_n}} \right]$$

  - *Fixed-matches estimated errors*: $\widehat{\mathrm{CI}}_{1-\alpha}^{\mathrm{bc}}(\widehat{h}_n, \widehat{b}_n)$, where

    $$\widehat{\mathrm{CI}}_{1-\alpha}^{\mathrm{bc}}(h_n, b_n) = \left[\left\{ \widehat{\tau}_p(h_n) - h_n^{p+1} \, \widehat{B}_{n,p,q} \right\} \pm \Phi_{1-\frac{\alpha}{2}}^{-1} \sqrt{\frac{\widehat{V}_p}{n h_n}} \right]$$

Here $\widehat{h}_n \in (\widehat{h}_{\mathrm{IK},n,p}, \widehat{h}_{\mathrm{CCT},n,p}, \widehat{h}_{\mathrm{CV},n,p})$ and $\widehat{b}_n \in (\widehat{b}_{\mathrm{IK},n,p+1,q}, \widehat{b}_{\mathrm{CCT},n,p+1,q})$, for example.

2.4 Robust RD inference

The classical CIs discussed above may have some unappealing properties that could seriously affect their performance in empirical work. The CIs $\check{\mathrm{CI}}_{1-\alpha}(h_n)$ and $\widehat{\mathrm{CI}}_{1-\alpha}(h_n)$ require undersmoothing (or small bias), which leads to potentially large-coverage distortions otherwise. The bias-corrected CIs $\check{\mathrm{CI}}^{\mathrm{bc}}_{1-\alpha}(h_n, b_n)$ and $\widehat{\mathrm{CI}}^{\mathrm{bc}}_{1-\alpha}(h_n, b_n)$, while theoretically justified for a larger range of bandwidths, are usually regarded as performing poorly in empirical settings, which also leads to potentially large-coverage distortions in applications. Monte Carlo evidence showing some of these potential pitfalls is reported in Calonico, Cattaneo, and Titiunik (2014d).

Calonico, Cattaneo, and Titiunik (2014d) propose alternative, more robust CIs constructed using bias-corrected RD treatment-effect estimators as a starting point. Intuitively, these estimators do not perform well in finite samples because the bias estimate introduces additional variability in $\widehat{\tau}^{\mathrm{bc}}_{p,q}(h_n, b_n) = \widehat{\tau}_p(h_n) - h_n^{p+1}\,\widehat{B}_{n,p,q}$. This variability is not accounted for when forming the associated CIs, for example, $\check{\mathrm{CI}}^{\mathrm{bc}}_{1-\alpha}(h_n, b_n)$ and $\widehat{\mathrm{CI}}^{\mathrm{bc}}_{1-\alpha}(h_n, b_n)$.

Following this reasoning, Calonico, Cattaneo, and Titiunik (2014d) propose an alternative asymptotic approximation for $\widehat{\tau}^{\mathrm{bc}}_{p,q}(h_n, b_n)$ that is heuristically given by the observation that

$$\sqrt{nh_n}\left\{\widehat{\tau}^{\mathrm{bc}}_{p,q}(h_n, b_n) - \tau\right\} = \underbrace{\sqrt{nh_n}\left\{\widehat{\tau}_p(h_n) - \tau - h_n^{p+1}\,B_{n,p}\right\}}_{\to_d\, N(0,V_p)}$$
$$-\underbrace{\sqrt{nh_n}h_n^{p+1}\left(\widehat{B}_{n,p,q} - B_{n,p}\right)}_{\to_d\, N\{0,V_{p+1,q}(\rho)\}}$$

provided that appropriate regularity conditions hold and

$$\frac{h_n}{b_n} \to \rho \in (0, \infty)$$

Here $V_{p,q}(\rho)$ could be interpreted as the contribution of the bias correction to the variability of the bias-corrected estimator. [It can be shown that $V_{p,q}(0) = 0$.]

Under weaker conditions than those typically imposed in the results in the previous subsections, Calonico, Cattaneo, and Titiunik (2014d) show that

$$\sqrt{nh_n}\left\{\widehat{\tau}^{\mathrm{bc}}_{p,q}(h_n, b_n) - \tau\right\} \to_d N\{0, V^{\mathrm{bc}}_{p,q}(\rho)\}$$

where $V^{\mathrm{bc}}_{p,q}(\rho)$ is the asymptotic variance for the bias-corrected estimator, which is different from the usual one, V_p. Indeed, it can be shown that $V^{\mathrm{bc}}_{p,q}(\rho) \to V_p$ if $\rho \downarrow 0$; but, in general, $V^{\mathrm{bc}}_{p,q}(\rho) > V_p$ under standard conditions.

It can also be shown that, under appropriate conditions,

$$\frac{\widehat{\tau}^{\mathrm{bc}}_{p,q}(h_n, b_n) - \tau}{\sqrt{V^{\mathrm{bc}}_{n,p,q}}} \to_d N(0, 1), \qquad V^{\mathrm{bc}}_{n,p,q} = V^{\mathrm{bc}}_{n,p,q}(h_n, b_n)$$

where the exact formula for $V_{n,p,q}^{\mathrm{bc}}(h_n, b_n)$ is given in Calonico, Cattaneo, and Titiunik (2014d). Intuitively, this variance formula is constructed to account for the variability of both the original RD treatment-effect estimator $\widehat{\tau}_p(h_n)$ and the bias-correction term, $\widehat{B}_{n,p,q}$, in the distributional approximation of the Studentized statistic. Further theoretical implications of this alternative approach to nonparametric bias correction are discussed in Calonico, Cattaneo, and Farrell (2014).

This more general distributional approximation leads to the following data-driven robust CIs:

- *Robust Bias Correction*:
    - *Plug-in estimated errors*: $\breve{\mathrm{CI}}_{1-\alpha}^{\mathrm{rbc}}(\widehat{h}_n, \widehat{b}_n)$, where

$$\breve{\mathrm{CI}}_{1-\alpha}^{\mathrm{rbc}}(h_n, b_n) = \left[\left\{ \widehat{\tau}_p(h_n) - h_n^{p+1} \, \widehat{B}_{n,p,q} \right\} \pm \Phi_{1-\frac{\alpha}{2}}^{-1} \, \sqrt{\breve{V}_{n,p,q}^{\mathrm{bc}}} \right]$$

    - *Fixed-matches estimated errors*: $\widehat{\mathrm{CI}}_{1-\alpha}^{\mathrm{rbc}}(\widehat{h}_n, \widehat{b}_n)$, where

$$\widehat{\mathrm{CI}}_{1-\alpha}^{\mathrm{rbc}}(h_n, b_n) = \left[\left\{ \widehat{\tau}_p(h_n) - h_n^{p+1} \, \widehat{B}_{n,p,q} \right\} \pm \Phi_{1-\frac{\alpha}{2}}^{-1} \, \sqrt{\widehat{V}_{n,p,q}^{\mathrm{bc}}} \right]$$

    As above, for example, $\widehat{h}_n \in (\widehat{h}_{\mathrm{IK},n,p}, \widehat{h}_{\mathrm{CCT},n,p}, \widehat{h}_{\mathrm{CV},n,p})$ and $\widehat{b}_n \in (\widehat{b}_{\mathrm{IK},n,p+1,q}, \widehat{b}_{\mathrm{CCT},n,p+1,q}, \widehat{h}_{\mathrm{CV},n,p})$.

The exact formulas for $\breve{V}_{n,p,q}^{\mathrm{bc}} = \breve{V}_{n,p,q}^{\mathrm{bc}}(h_n, b_n)$ and $\widehat{V}_{n,p,q}^{\mathrm{bc}} = \widehat{V}_{n,p,q}^{\mathrm{bc}}(h_n, b_n)$ are given in Calonico, Cattaneo, and Titiunik (2014d).

2.5   Procedures implemented for sharp RD inference

The commands `rdbwselect` and `rdrobust` implement the following procedures:

- `rdbwselect` implements three bandwidth selectors for $h_{\mathrm{MSE},n,p}$:

    - $\widehat{h}_{\mathrm{IK},n,p}$: IK implementation for pth-order local polynomial estimator.
    - $\widehat{h}_{\mathrm{CCT}}, n, p$: CCT implementation for pth-order local polynomial estimator. This is the default in `rdbwselect`.
    - $\widehat{h}_{\mathrm{CV},n,p}$: CV implementation for pth-order local polynomial estimator.

- `rdbwselect` implements two bandwidth selectors for $b_{\mathrm{MSE},n,p,q}$: $\widehat{b}_{\mathrm{IK},n,p,q}$ and $\widehat{b}_{\mathrm{CCT},n,p,q}$.

    - $\widehat{b}_{\mathrm{IK},n,p,q}$: IK-analog implementation for pth derivative of qth-order local polynomial estimator.
    - $\widehat{b}_{\mathrm{CCT},n,p,q}$: CCT implementation for pth derivative of qth-order local polynomial estimator. This is the default in `rdbwselect`.

- `rdrobust` implements two point estimators for τ:

  - $\widehat{\tau}_p(h_n)$: pth-order local polynomial estimator. This is the default in `rdrobust`.

  - $\widehat{\tau}_{p,q}^{\mathrm{bc}}(h_n, b_n)$: pth-order local polynomial estimator with qth-order local polynomial bias correction.

- `rdrobust` implements six CIs for τ:

  - $\breve{\mathrm{CI}}_{1-\alpha}(h_n)$: no bias correction, conventional variance, plug-in residuals.
  - $\widehat{\mathrm{CI}}_{1-\alpha}(h_n)$: no bias correction, conventional variance, fixed-matches residuals.
  - $\breve{\mathrm{CI}}_{1-\alpha}^{\mathrm{bc}}(h_n, b_n)$: bias correction, conventional variance, plug-in residuals.
  - $\widehat{\mathrm{CI}}_{1-\alpha}^{\mathrm{bc}}(h_n, b_n)$: bias correction, conventional variance, fixed-matches residuals.
  - $\breve{\mathrm{CI}}_{1-\alpha}^{\mathrm{rbc}}(h_n, b_n)$: bias correction, robust variance, plug-in residuals.
  - $\widehat{\mathrm{CI}}_{1-\alpha}^{\mathrm{rbc}}(h_n, b_n)$: bias correction, robust variance, fixed-matches residuals. This is the default in `rdrobust`.

We give details on the syntax of `rdrobust` and `rdbwselect` in section 3 and section 4, respectively. We include a complete empirical example that illustrates these methods and commands using real data in section 6. Further details on implementation and other technical issues are discussed in Imbens and Kalyanaraman (2012) and Calonico, Cattaneo, and Titiunik (2014d).

2.6   Extensions to other RD designs

As already mentioned, Calonico, Cattaneo, and Titiunik (2014d) also show how to construct analog robust CIs for average treatment effects (at the cutoff) in other RD contexts, including sharp kink RD, fuzzy RD, and fuzzy kink RD designs. For further discussion on these RD contexts, see, for example, Card et al. (2014) and Dong (2011).

Our implementation also includes the following empirically relevant extensions, among other possibilities:

- *Sharp Kink RD Design.* Here the estimand involves the derivative of the underlying regression functions at the cutoff (to a known scale) as opposed to their level. Using the notation introduced above, we write the generic estimand as

$$\tau^{(s)} = \mu_+^{(s)} - \mu_-^{(s)}$$

  where usually $s = 1$, and the population parameter of interest is $\tau^{(1)}/\kappa$ with κ being a known scaling factor (that is, sharp kink RD estimand). The corresponding conventional local polynomial RD estimator is, up to the known scale κ,

$$\widehat{\tau}_p^{(s)}(h_n) = \widehat{\mu}_{+,p}^{(s)}(h_n) - \widehat{\mu}_{-,p}^{(s)}(h_n)$$

where $\widehat{\mu}^{(s)}_{+,p}(h_n) = \mathbf{e}'_s\widehat{\boldsymbol{\beta}}_{+,p}(h_n)$ and $\widehat{\mu}^{(s)}_{-,p}(h_n) = \mathbf{e}'_s\widehat{\boldsymbol{\beta}}_{-,p}(h_n)$, with $s \leqslant p$. All the results discussed herein extend naturally to this case, and our implementations allow for this possibility using the option `deriv()` to set the derivative order and the option `scalepar()` to set the value of κ. See sections 3 and 4 for further details.

- *Fuzzy RD Design.* Here the estimand takes the form of a ratio of two sharp RD estimands: one for the main reduced-form equation (that is, the regression of Y_i on X_i) and the other for the first-stage equation (that is, the regression of T_i on X_i, where T_i denotes actual treatment status). As discussed in Calonico, Cattaneo, and Titiunik (2014d), robust bias-corrected CIs can be constructed in this case as well. In our command, CIs for the fuzzy RD estimand are implemented with the option `fuzzy()`, as discussed in sections 3 and 4.

- *Fuzzy Kink RD Design.* Finally, the results also provide robust bias-corrected CIs in the context of a fuzzy kink RD design, where the estimand of interest is the ratio of two sharp kink RD estimands: one for the main reduced-form equation and the other for the first-stage equation. In our command, CIs for the fuzzy kink RD estimand are implemented when both the `deriv()` and `fuzzy()` options are specified jointly, as discussed in sections 3 and 4.

Note that `rdrobust` conducts sharp RD inference by default. See section 3 and section 4 for details on the syntax of `rdrobust` and `rdbwselect`, respectively, to construct CIs in the other cases.

2.7 Data-driven RD plots

The main aspects of the RD design can be summarized in an easy-to-interpret figure, which shows how an estimated regression function behaves for control ($X_i < \overline{x}$) and treated ($X_i \geqslant \overline{x}$) units relative to some summary of the actual data. This common RD plot gives an idea of overall fit while also exhibiting graphically the sharp RD estimate. In most empirical applications, this figure is constructed using "dots" for local sample means over nonoverlapping bins partitioning a restricted support of X_i, together with two smooth "global" polynomial regression curve estimates for control and treatment units separately. The binned means are usually included to capture the behavior of the "cloud of points" and to show whether there are other discontinuities in the data away from the cutoff; the two global polynomial estimates are meant to give a flexible global approximation of $\mu_-(x)$ and $\mu_+(x)$. An example of this kind of plot using the data analyzed by Lee (2008) is shown in figure 1.
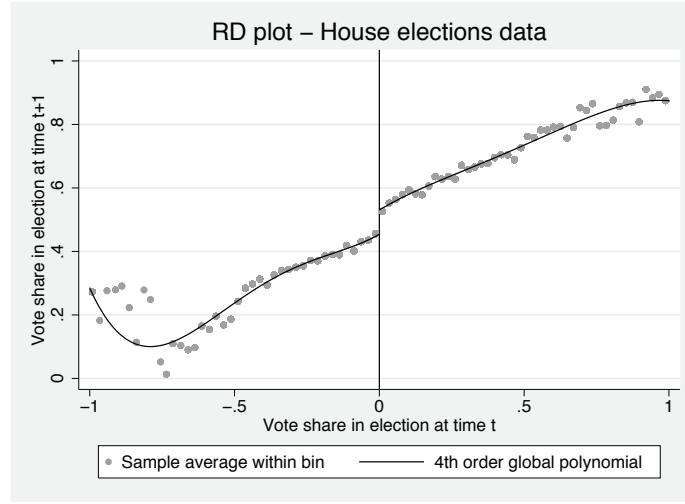
Figure 1. Ad hoc RD Plot with evenly spaced bins for U.S. House elections dataset

Calonico, Cattaneo, and Titiunik (2014a) study these RD plots and develop several (optimal) choices of the number of bins under two partitioning schemes: evenly spaced and quantile-spaced partitions. Here we briefly summarize the main aspects of this approach.

RD plots use two main ingredients. First, polynomial regression curves are estimated to flexibly approximate the population conditional mean functions for control and treated units, usually over a large but restricted support of the running variable. Formally, these estimates are the pth-order global polynomials given by

$$\widehat{\mu}_{-,p,1}(x) = \mathbf{r}_p(x)'\widehat{\boldsymbol{\gamma}}_{-,p,1}(x_l) \qquad \text{and} \qquad \widehat{\mu}_{+,p,1}(x) = \mathbf{r}_p(x)'\widehat{\boldsymbol{\gamma}}_{+,p,1}(x_u)$$

with

$$\widehat{\boldsymbol{\gamma}}_{-,p,k}(x_l) = \arg\min_{\boldsymbol{\gamma}\in\mathbb{R}^{p+1}} \sum_{i=1}^{n} \mathbb{1}(x_l < X_i \leqslant \overline{x})\{Y_i^k - \mathbf{r}_p(X_i)'\boldsymbol{\gamma}\}^2$$

$$\widehat{\boldsymbol{\gamma}}_{+,p,k}(x_u) = \arg\min_{\boldsymbol{\gamma}\in\mathbb{R}^{p+1}} \sum_{i=1}^{n} \mathbb{1}(\overline{x} \leqslant X_i < x_u)\{Y_i^k - \mathbf{r}_p(X_i)'\boldsymbol{\gamma}\}^2$$

where $\mathbf{r}_p(x) = (1, x, \dots, x^p)'$ and $k = 1, 2$. Here x_l and x_u are the lower and upper limits on the support of the running variable, and they satisfy $x_l < \overline{x} \leqslant x_u$. In other words, $\widehat{\mu}_{-,p,1}(x)$ and $\widehat{\mu}_{+,p,1}(x)$ are pth-order global polynomials over the supports $[x_l, \overline{x}]$ and $[\overline{x}, x_u]$, respectively. This ingredient of the figure is easily implemented in practice, with common choices being $p = 4$ and $p = 5$.

Second, sample means are constructed over nonoverlapping regions of the support of the running variable X_i, for control and treatment units separately. These sample means provide an approximation of the population regression functions, but they also help visualize the dispersion of the data, which could be used to detect other potential

discontinuities away from the cutoff (as a form of a falsification test). To implement these estimators, we construct two evenly spaced partitions for control and treatment units separately, as follows,

$$P_{-,n} = \bigcup_{j=1}^{J_{-,n}} P_{-,j,n} = [x_l, \overline{x}) \qquad \text{and} \qquad P_{+,n} = \bigcup_{j=1}^{J_{+,n}} P_{+,j,n} = [\overline{x}, x_u]$$

$$P_{-,j,n} = \begin{cases} [x_l \ , \ p_{-,1}) & j = 1 \\ [p_{-,j-1} \ , \ p_{-,j}) & j = 2, \ldots, J_{-,n} - 1 \\ [p_{-,J_{-,n}-1} \ , \ \overline{x}) & j = J_{-,n} \end{cases}$$

$$P_{+,j,n} = \begin{cases} [\overline{x} \ , \ p_{+,1}) & j = 1 \\ [p_{+,j-1} \ , \ p_{+,j}) & j = 2, \ldots, J_{+,n} - 1 \\ [p_{+,J_{+,n}-1} \ , \ x_u] & j = J_{+,n} \end{cases}$$

with $p_{-,0} < p_{-,1} < \cdots < p_{-,J_{-,n}}$ and $p_{+,0} < p_{+,1} < \cdots < p_{+,J_{+,n}}$. Thus $P_n = P_{-,n} \cup P_{+,n}$ forms a (disjoint) partition of (x_l, x_u) centered at $\overline{x}$, which is assumed to become finer as the sample size grows (that is, $J_{-,n} \to \infty$ and $J_{+,n} \to \infty$).

We consider the following two partitioning schemes:

$$\text{evenly spaced (ES):} \quad p_{-,j} = x_l + \frac{j \ (\overline{x} - x_l)}{J_{-,n}} \quad \text{and} \quad p_{+,j} = \overline{x} + \frac{j \ (x_u - \overline{x})}{J_{+,n}}$$

$$\text{quantile spaced (QS):} \quad p_{-,j} = X_{-,(\lfloor \frac{j}{J_{-,n}} \rfloor)} \quad \text{and} \quad p_{+,j} = X_{+,(\lfloor \frac{j}{J_{+,n}} \rfloor)}$$

$X_{-,(i)}$ and $X_{+,(i)}$ denote the ith quantile of the control and treatment subsamples, respectively, and $\lfloor \cdot \rfloor$ denotes the floor function. To implement the ES or QS binning in practice, we must select the lengths of the bins for control and treated units, which are determined by $J_{-,n}$ and $J_{+,n}$, respectively. The resulting binned means can be written as

$$\breve{\mu}_-(x; J_{-,n}) = \sum_{j=1}^{J_{-,n}} \mathbb{1}_{P_{-,j,n}}(x) \, \overline{Y}_{-,j}, \qquad \overline{Y}_{-,j} = \frac{1}{N_{-,j}} \sum_{i=1}^{n} \mathbb{1}_{P_{-,j,n}}(X_i) \, Y_i$$

$$\breve{\mu}_+(x; J_{+,n}) = \sum_{j=1}^{J_{+,n}} \mathbb{1}_{P_{+,j,n}}(x) \, \overline{Y}_{+,j}, \qquad \overline{Y}_{+,j} = \frac{1}{N_{+,j}} \sum_{i=1}^{n} \mathbb{1}_{P_{+,j,n}}(X_i) \, Y_i$$

with $N_{-,j} = \sum_{i=1}^{n} \mathbb{1}_{P_{-,j,n}}(X_i)$, $N_{+,j} = \sum_{i=1}^{n} \mathbb{1}_{P_{+,j,n}}(X_i)$, and $\mathbb{1}_P(x) = \mathbb{1}(x \in P)$. These estimators are a simple version of the "nonparametric partitioning estimators"; see, for example, Cattaneo and Farrell (2013) and references therein.

Calonico, Cattaneo, and Titiunik (2014a) develop asymptotic expansions of the integrated variance and square-bias of the partitioning estimators under ES and QS partitioning, and they obtain two distinct (optimal) choices for $J_{-,n}$ and $J_{+,n}$ depending on the explicit goal chosen. They derive an integrated mean squared-error (IMSE)-optimal choice, which is explicitly tailored to produce binned sample means that "trace out" the underlying regression function. This choice is useful for falsification purposes (for

example, identifying potential discontinuities at values of the running variable different from the RD cutoff). The selectors of the number of bins for control and treatment units, respectively, take the form

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

where the exact form of the constants depends on the partitioning scheme used, ES or QS, and $\lceil \cdot \rceil$ denotes the ceiling operator.

Calonico, Cattaneo, and Titiunik (2014a) also propose a different rule for selecting the number of bins used in the ES and QS RD plots that is explicitly tailored to approximate the underlying variability of the raw data. This approach, which leads to a substantially larger number of bins than the IMSE-optimal choice, is useful to depict the data in a disciplined and objective way. The corresponding selectors for control and treatment units, respectively, are given by

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

where here the specific constants are different from those in the IMSE-optimal choices, and they also vary depending on the partitioning scheme employed.

For either method, feasible plug-in rules can be constructed using preliminary estimators for the unknown constants $\mathscr{C}_-$ and $\mathscr{C}_+$ that enter each of the number-of-bins selectors. Specifically, Calonico, Cattaneo, and Titiunik (2014a) propose implementations combining spacings and series estimation techniques, with the implementation varying depending on whether the outcome variable is continuously distributed or not. All details regarding these selectors, and the resulting data-driven RD plots constructed using them, can be found in Calonico, Cattaneo, and Titiunik (2014a).

Our command `rdplot` implements eight distinct data-driven RD plots that vary in the partitioning scheme used (ES or QS), the objective desired (approximating the regression function or depicting the variability of the data), and the nonparametric procedure used for implementation (spacings estimators or polynomial series estimators). We discuss the syntax details of `rdplot` in section 5, and we give an empirical example of this command in section 6.

3   The rdrobust command

In this section, we describe the syntax of the command `rdrobust` to conduct point estimation and robust inference for mean treatment effects in the RD design.

3.1   Description

`rdrobust` provides an array of local-polynomial-based inference procedures for mean treatment effects in the RD design. The user must specify the dependent and running

variable. This command permits fully data-driven inference by using the companion command `rdbwselect`, which can also be used as a stand-alone command. We describe `rdbwselect` below.

3.2 Syntax

`rdrobust` *depvar* *runvar* $\big[$ *if* $\big]$ $\big[$ *in* $\big]$ $\big[$, `c(`*cutoff*`)` `p(`*pvalue*`)` `q(`*qvalue*`)`
   `deriv(`*dvalue*`)` `fuzzy(`*fuzzyvar*`)` `kernel(`*kernelfn*`)` `h(`*hvalue*`)` `b(`*bvalue*`)`
   `rho(`*rhovalue*`)` `scalepar(`*scaleparvalue*`)` `bwselect(`*bwmethod*`)`
   `scaleregul(`*scaleregulvalue*`)` `delta(`*deltavalue*`)` `cvgrid_min(`*cvgrid_minvalue*`)`
   `cvgrid_max(`*cvgrid_maxvalue*`)` `cvgrid_length(`*cvgrid_lengthvalue*`)` `cvplot`
   `vce(`*vcemethod*`)` `matches(`*nummatches*`)` `level(`*level*`)` `all` $\big]$

where *depvar* is the dependent variable, and *runvar* is the running variable (also known as the score or forcing variable).

3.3 Options

`c(`*cutoff*`)` specifies the RD cutoff. The default is `c(0)`.

`p(`*pvalue*`)` specifies the order of the local polynomial to be used to construct the point estimator. The default is `p(1)` (local linear regression).

`q(`*qvalue*`)` specifies the order of the local polynomial to be used to construct the bias correction. The default is `q(2)` (local quadratic regression).

`deriv(`*dvalue*`)` specifies the order of the derivative of the regression functions to be estimated. The default is `deriv(0)` (sharp RD, or fuzzy RD if `fuzzy()` is also specified). Setting `deriv(1)` results in estimation of a kink RD design (up to scale), or fuzzy kink RD if `fuzzy()` is also specified.

`fuzzy(`*fuzzyvar*`)` specifies the treatment status variable used to implement fuzzy RD estimation (or fuzzy kink RD if `deriv(1)` is also specified). The default is sharp RD design; hence, this option is not used. For fuzzy RD designs, bandwidths are estimated using sharp RD bandwidth selectors for the reduced-form outcome equation.

`kernel(`*kernelfn*`)` specifies the kernel function used to construct the local polynomial estimators. *kernelfn* may be <u>tri</u>angular, <u>epa</u>nechnikov, and <u>uni</u>form. The default is `kernel(triangular)`.

`h(`*hvalue*`)` specifies the main bandwidth, h_n, to be used to construct the RD point estimator. If not specified, this is computed by the companion command `rdbwselect`.

`b(`*bvalue*`)` specifies the pilot bandwidth, b_n, used to construct the bias-correction estimator. If not specified, this is computed by the companion command `rdbwselect`.

rho(*rhovalue*) specifies the value of ρ so that the pilot bandwidth, b_n, equals $b_n = h_n/\rho$. The default is rho(1) if h_n is specified but b_n is not.

scalepar(*scaleparvalue*) specifies the scaling factor for the RD parameter of interest. This option is useful when the population parameter of interest involves a known multiplicative factor (for example, sharp kink RD). The default is scalpar(1) (no scaling).

bwselect(*bwmethod*) specifies the bandwidth selection procedure to be used. By default, it computes both h_n and b_n, unless ρ is specified, in which case it computes only the h_n and sets $b_n = h_n/\rho$. *bwmethod* may be one of the following:

   CCT for the bandwidth selector proposed by Calonico, Cattaneo, and Titiunik (2014d). The default is bwselect(CCT).

   IK for the bandwidth selector proposed by Imbens and Kalyanaraman (2012) (available for only sharp RD design).

   CV for the cross-validation method proposed by Ludwig and Miller (2007) (available for only sharp RD design).

scaleregul(*scaleregulvalue*) specifies the scaling factor for the regularization terms of CCT and IK bandwidth selectors. Setting scaleregulvalue(0) removes the regularization term from the bandwidth selectors. See companion command rdbwselect for more details. The default is scaleregul(1).

delta(*deltavalue*) specifies the quantile that defines the sample used in the cross-validation procedure. This option is used only if bwselect(CV) is specified. See companion command rdbwselect for more details. The default is delta(0.5), that is, the median of the control and treated subsamples.

cvgrid_min(*cvgrid_minvalue*) specifies the minimum value of the bandwidth grid used in the cross-validation procedure. This option is used only if bwselect(CV) is specified. See companion command rdbwselect for more details.

cvgrid_max(*cvgrid_maxvalue*) specifies the maximum value of the bandwidth grid used in the cross-validation procedure. This option is used only if bwselect(CV) is specified. See companion command rdbwselect for more details.

cvgrid_length(*cvgrid_lengthvalue*) specifies the bin length of the (evenly spaced) bandwidth grid used in the cross-validation procedure. This option is used only if bwselect(CV) is specified. See companion command rdbwselect for more details.

cvplot generates a graph of the cross-validation objective function. This option is used only if bwselect(CV) is specified. See companion command rdbwselect for more details.

vce(*vcemethod*) specifies the procedure used to compute the variance–covariance matrix estimator. *vcemethod* may be one of the following:

nn for nearest-neighbor matches residuals using matches(). This is the default option (with matches(3), see below).

resid for estimated plug-in residuals using h_n bandwidth.

matches(*nummatches*) specifies the number of matches in the nearest-neighbor-based variance–covariance matrix estimator. This option is used only when nearest-neighbor matches residuals are used. The default is matches(3).

level(*level*) is the confidence level for CIs. The default is level(95).

all specifies that rdrobust report three different procedures:

  i) conventional RD estimates with a conventional variance estimator;
 ii) bias-corrected RD estimates with a conventional variance estimator; and
iii) bias-corrected RD estimates with a robust variance estimator.

4   The rdbwselect command

Here we describe the syntax for rdbwselect. This command implements the different bandwidth selection procedures for the local-polynomial regression-discontinuity estimators discussed above.

4.1   Description

rdbwselect implements several bandwidth selection procedures currently available for the RD design. The user must specify the dependent and running variable.

4.2   Syntax

rdbwselect *depvar* *runvar* $\big[$ *if* $\big]$ $\big[$ *in* $\big]$ $\big[$, c(*cutoff*) p(*pvalue*) q(*qvalue*)

   deriv(*dvalue*) rho(*rhovalue*) kernel(*kernelfn*) bwselect(*bwmethod*)

   scaleregul(*scaleregulvalue*) delta(*deltavalue*) cvgrid_min(*cvgrid_minvalue*)

   cvgrid_max(*cvgrid_maxvalue*) cvgrid_length(*cvgrid_lengthvalue*) cvplot

   vce(*vcemethod*) matches(*nummatches*) all $\big]$

where *depvar* is the dependent variable, and *runvar* is the running variable (also known as the score or forcing variable).

4.3   Options

c(*cutoff*) specifies the RD cutoff. The default is c(0).

p(*pvalue*) specifies the order of the local polynomial to be used to construct the point estimator. The default is p(1) (local linear regression).

q(*qvalue*) specifies the order of the local polynomial to be used to construct the bias correction. The default is q(2) (local quadratic regression).

deriv(*dvalue*) specifies the order of the derivative of the regression functions to be estimated. The default is deriv(0) (sharp RD, or fuzzy RD if fuzzy() is also specified). Setting deriv(1) results in estimation of a kink RD design (up to scale), or fuzzy kink RD if fuzzy() is also specified.

rho(*rhovalue*) sets the pilot bandwidth, b_n, equal to h_n/ρ, where h_n is computed using the method and options chosen below.

kernel(*kernelfn*) specifies the kernel function used to construct the local polynomial estimators. Options are <u>tri</u>angular, <u>epa</u>nechnikov, and <u>uni</u>form. The default is kernel(triangular).

bwselect(*bwmethod*) specifies the bandwidth selection procedure to be used. By default, it computes both h_n and b_n, unless ρ is specified, in which case it computes only h_n and sets $b_n = h_n/\rho$. *bwmethod* may be one of the following:

    CCT for the bandwidth selector proposed by Calonico, Cattaneo, and Titiunik (2014d). The default is bwselect(CCT).

    IK for the bandwidth selector proposed by Imbens and Kalyanaraman (2012) (available for only sharp RD design).

    CV for the cross-validation method proposed by Ludwig and Miller (2007) (available for only sharp RD design).

scaleregul(*scaleregulvalue*) specifies the scaling factor for the regularization terms of CCT and IK bandwidth selectors. Setting scaleregulvalue(0) removes the regularization term from the bandwidth selectors. The default is scalregul(1).

delta(*deltavalue*) specifies the quantile that defines the sample used in the cross-validation procedure. This option is used only if bwselect(CV) is specified. The default is delta(0.5), that is, the median of the control and treated subsamples.

cvgrid_min(*cvgrid_minvalue*) specifies the minimum value of the bandwidth grid used in the cross-validation procedure. This option is used only if bwselect(CV) is specified.

cvgrid_max(*cvgrid_maxvalue*) specifies the maximum value of the bandwidth grid used in the cross-validation procedure. This option is used only if bwselect(CV) is specified.

cvgrid_length(*cvgrid_lengthvalue*) specifies the bin length of the (evenly spaced) bandwidth grid used in the cross-validation procedure. This option is used only if bwselect(CV) is specified.

`cvplot` generates a graph of the cross-validation objective function. This option is used only if `bwselect(CV)` is specified.

`vce(`*vcemethod*`)` specifies the procedure used to compute the variance–covariance matrix estimator. This option is used only if the `bwselect(CCT)` or `bwselect(IK)` bandwidth procedure is used. *vcemethod* may be one of the following:

    `nn` for nearest-neighbor matches residuals using `matches()`. This is the default (with `matches(3)`, see below).

    `resid` for estimated plug-in residuals using h_n bandwidth.

`matches(`*nummatches*`)` specifies the number of matches in the nearest-neighbor-based variance–covariance matrix estimator. This option is used only when nearest-neighbor matches residuals are used. The default is `matches(3)`.

`all` implements all three bandwidth selection procedures; see `bwselect()` above.

5 The rdplot command

In this section, we describe the syntax of the `rdplot` command, which implements different data-driven RD plots using either evenly spaced or quantile-spaced binning. The number of bins on these plots is chosen either i) to approximate the underlying regression function (IMSE-optimal selectors) or ii) to mimic the underlying variability of the raw data. For further details on these approaches, see Calonico, Cattaneo, and Titiunik (2014a).

5.1 Description

`rdplot` implements several selectors of the number of bins used to construct RD plots using either an evenly spaced or a quantile-spaced partitioning of the support of the running variable X_i. Two distinct methods are implemented: one to approximate the regression function and one to display the variability of the data in an objective way. The user must specify the dependent and running variables.

5.2 Syntax

`rdplot` *depvar runvar* $\big[$ *if* $\big]$ $\big[$ *in* $\big]$ $\big[$ `,` `c(`*cutoff*`)` `p(`*pvalue*`)` `numbinl(`*numbinlvalue*`)`

    `numbinr(`*numbinrvalue*`)` `binselect(`*binmethod*`)` `lowerend(`*xlvalue*`)`

    `upperend(`*xuvalue*`)` `scale(`*scalevalue*`)` `scalel(`*scalelvalue*`)` `scaler(`*scalervalue*`)`

    `generate(`*idname meanxname meanyname*`)` `graph_options(`*gphopts*`)` `hide` $\big]$

where *depvar* is the dependent variable, and *runvar* is the running variable (also known as the score or forcing variable).

5.3   Options

c(*cutoff*) specifies the RD cutoff. The default is c(0).

p(*pvalue*) specifies the order of the global polynomial used to approximate the population conditional mean functions for control and treated units. The default is p(4).

numbinl(*numbinlvalue*) specifies the number of bins used to the left of the cutoff, denoted J_-. If not specified, J_- is estimated using the method and options chosen below.

numbinr(*numbinrvalue*) specifies the number of bins used to the right of the cutoff, denoted J_+. If not specified, J_+ is estimated using the method and options chosen below.

binselect(*binmethod*) specifies the procedure to select the number of bins. This option is available only if J_- and J_+ are not set manually. *binmethod* may be one of the following:

   es specifies the IMSE-optimal evenly spaced method using spacings estimators.

   espr specifies the IMSE-optimal evenly spaced method using polynomial regression.

   esmv specifies the mimicking variance evenly spaced method using spacings estimators; the default.

   esmvpr specifies the mimicking-variance evenly spaced method using polynomial regression.

   qs specifies the IMSE-optimal quantile-spaced method using spacings estimators.

   qspr specifies the IMSE-optimal quantile-spaced method using polynomial regression.

   qsmv specifies the mimicking-variance quantile-spaced method using spacings estimators.

   qsmvpr specifies the mimicking-variance quantile-spaced method using polynomial regression.

lowerend(*xlvalue*) specifies the lower bound for *indepvar* to the left of the cutoff. The default is the minimum value in sample.

upperend(*xuvalue*) specifies the upper bound for *indepvar* to the right of the cutoff. The default is the maximum value in sample.

scale(*scalevalue*) specifies a multiplicative factor to be used with the optimal number of bins selected. Specifically, the number of bins used for the treatment and control groups will be scale(*scalevalue*) $\times J_+$ and scale(*scalevalue*) $\times J_-$, where J_- and J_+ denote the optimal numbers of bins originally computed for each group. The default is scale(1).

scalel(*scalelvalue*) specifies a multiplicative factor to be used with the optimal number of bins selected to the left of the cutoff. The number of bins used will be scalel(*scalelvalue*) $\times \widehat{J}_{-,n}$. The default is scalel(1).

scaler(*scalervalue*) specifies a multiplicative factor to be used with the optimal number of bins selected to the right of the cutoff. The number of bins used will be scaler(*scalerval*) $\times \widehat{J}_{+,n}$. The default is scaler(1).

generate(*idname meanxname meanyname*) generates new variables storing the results:

idname specifies the name of a new generated variable with a unique bin id that identifies the chosen bins. This variable indicates the bin (between lowerend() and upperend()) to which each observation belongs. Negative natural numbers are assigned to observations to the left of the cutoff, and positive natural numbers are assigned to observations to the right of the cutoff.

meanxname specifies the name of a new generated variable (of the same length as *idname*) with the middle point of the running variable within each chosen bin.

meanyname specifies the name of a new generated variable (of the same length as *idname*) with the sample mean of the outcome variable within each chosen bin.

graph_options(*gphopts*) specifies graphical options to be passed on to the underlying graph command.

hide omits the RD plot.

6   Illustration of methods

We illustrate our commands using an extract of the dataset constructed by Cattaneo, Frandsen, and Titiunik (Forthcoming); the dataset comes from a study on party advantages in U.S. Senate elections for the period 1914–2010. In particular, we focus here 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. The unit of observation is the state. This empirical illustration is analogous to the one presented by Lee (2008) for U.S. House elections.

The dataset rdrobust_rdsenate contains two variables: vote and margin. The variable margin ranges from −100 to 100 and records the Democratic party's margin of victory in the statewide election for a given U.S. Senate seat, which is defined as the vote share of the Democratic party minus the vote share of its strongest opponent. The variable vote ranges from 0 to 100 and records the Democratic vote share in the following election for the same seat (6 years later). Thus observations for years 2008 and 2010 are missing vote. When margin is above zero, the Democratic party wins the election for that seat; otherwise, it loses. As is usual in the literature, we exploit this discontinuity in incumbency status that occurs at margin = 0 to estimate the incumbency advantage of parties with an RD design.

The dataset has a total of 1,297 complete observations. We load the database and present basic summary statistics of these two variables.

```
. use rdrobust_rdsenate.dta
. summarize vote margin
    Variable |      Obs       Mean    Std. Dev.      Min       Max
-------------+--------------------------------------------------------
        vote |     1297   52.66627   18.12219        0       100
      margin |     1390   7.171159   34.32488     -100       100
```

To further explore the available data, we use `rdplot` to construct an automatic plot of the RD design.

```
. rdplot vote margin,
>         graph_options(title(RD Plot - Senate elections data)
>                           ytitle(Vote share in election at time t+1)
>                           xtitle(Vote share in election at time t))
Number of bins for RD estimates.
Method: Mimicking Variance evenly spaced using spacings estimators.
```

Cutoff c = 0	Left of c	Right of c
Number of observations	595	702
Polynomial order	4	4
Chosen scale	1.000	1.000
Selected bins	15	35
Bin length	6.661	2.856
IMSE-optimal bins	8	9
Mimicking Variance bins	15	35
Relative to IMSE-optimal:		
Implied scale	1.875	3.889
WIMSE variance weight	0.132	0.017
WIMSE bias weight	0.868	0.983

Figure 2 is constructed using the default options in the command `rdplot`, which produce an RD plot that has evenly spaced bins that mimic the underlying variability of the data and is implemented using spacings estimators. Using the notation introduced above, we see that the number of optimal bins for control and treatment units is $\widehat{J}_{-,n} = 15$ and $\widehat{J}_{+,n} = 35$, respectively, implying bin lengths of 6.661 and 2.856 percentage points, respectively. The global polynomial is constructed using a 4th-degree polynomial [$p = 4$ for $\widehat{\mu}_{-,p,1}(x)$ and $\widehat{\mu}_{+,p,1}(x)$]. The output table also reports the IMSE-optimal number of bins (first row of the middle panel) and the multiplicative factor (scale) associated with the selected number of bins (taking the IMSE-optimal value as a reference). This is shown in the last row of the first panel. Finally, the bottom panel includes the IMSE weights that correspond to the selected choice of the number of bins—see Calonico, Cattaneo, and Titiunik (2014a, sec. 3.3.1) for additional details.
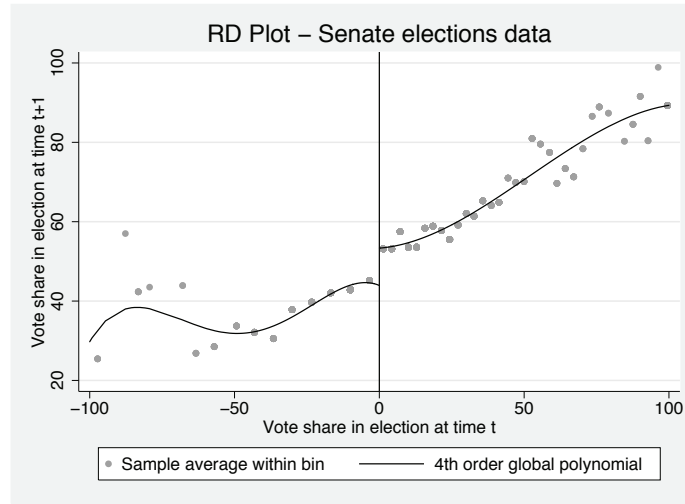
Figure 2. Automatic RD plot

Next, we construct an alternative RD plot using evenly spaced bins selected to trace out the underlying regression function (that is, IMSE-optimal selectors) and implemented using spacings estimators. The resulting plot is given in figure 3, which shows how the (local) binned sample means indeed seem to approximate the underlying regression function well (taking the global polynomial fit as a benchmark). This does not provide evidence of potential discontinuities away from the cutoff in the underlying regression functions.

```
. rdplot vote margin, binselect(es)
>     graph_options(title(RD plot - Senate elections data)
>                   ytitle(Vote share in election at time t+1)
>                   xtitle(Vote share in election at time t))
Number of bins for RD estimates.
Method: IMSE-optimal evenly spaced method using spacings estimators.
```

Cutoff c = 0	Left of c	Right of c
Number of observations	595	702
Polynomial order	4	4
Chosen scale	1.000	1.000
Selected bins	8	9
Bin length	12.490	11.107
IMSE-optimal bins	8	9
Mimicking Variance bins	15	35
Relative to IMSE-optimal:		
Implied scale	1.000	1.000
WIMSE variance weight	0.500	0.500
WIMSE bias weight	0.500	0.500

The last two figures show different features of the underlying data and research design. Figure 2 gives a disciplined scatterplot of the data, which seeks to represent the overall variability of the raw data. This figure uses more than the IMSE-optimal number of bins (that is, it undersmooths the nonparametric partitioning estimator), which induces more variability in the underlying binned sample means. This gives a visual "cloud of points" on top of the polynomial approximation to the regression function. On the other hand, figure 3 uses the IMSE-optimal number of bins specifically tailored to produce an estimator that approximates the underlying regression function well. In this approach, the optimal number of bins is selected to balance squared bias and variance to approximate the underlying conditional expectation globally.
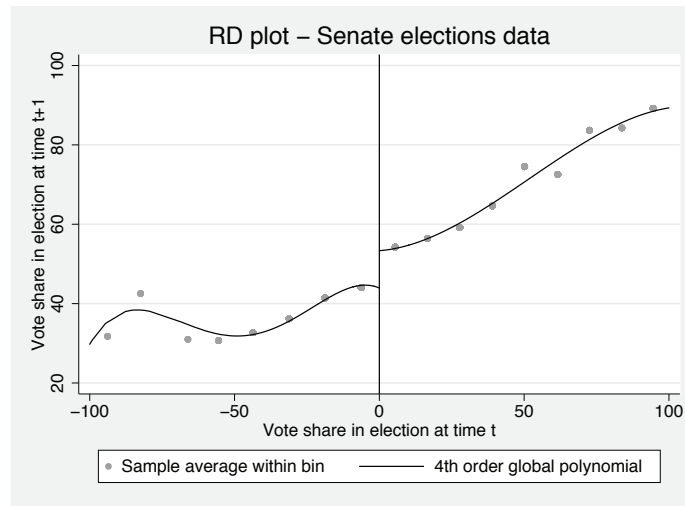

Figure 3. Automatic RD plot, with scaled-down optimal bin-length choice

Finally, to illustrate some of the other features of the command `rdplot`, we present figure 4. Here the RD plot is constructed using the quantile-spaced binning approach. In this case, the number of optimal bins for control and treatment units is $\widehat{J}_{-,n} = 28$ and $\widehat{J}_{+,n} = 49$, respectively.

```
. rdplot vote margin, binselect(qsmv)
>               graph_options(title(RD plot - Senate elections data)
>        ytitle(Vote share in election at time t+1)
>        xtitle(Vote share in election at time t))
```

Number of bins for RD estimates.
Method: Mimicking Variance quantile spaced using spacings estimators.

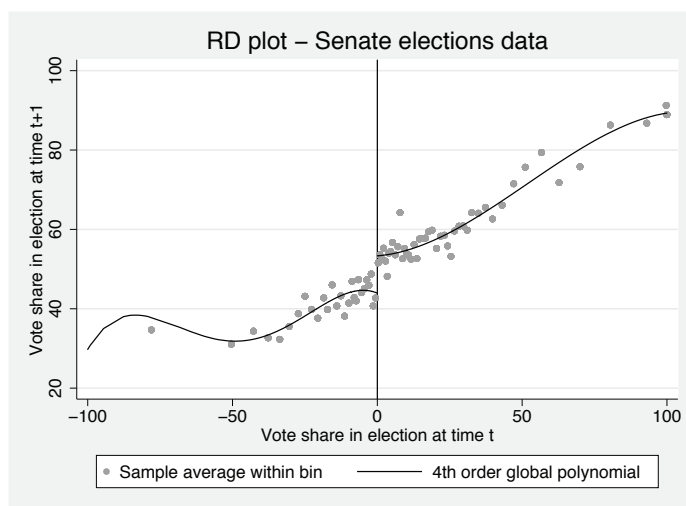| Cutoff c = 0	Left of c	Right of c
Number of observations	595	702
Polynomial order	4	4
Chosen scale	1.000	1.000
Selected bins	28	49
Bin length	3.569	2.040
IMSE-optimal bins	21	16
Mimicking Variance bins	28	49
Relative to IMSE-optimal:		
Implied scale	1.333	3.062
WIMSE variance weight	0.297	0.034
WIMSE bias weight	0.703	0.966

Figure 4. Automatic RD plot using a quantile-spaced approach

Next, we conduct fully data-driven RD treatment-effects estimation and inference. The command **rdrobust**, using its default options, gives the following output:

```
. rdrobust vote margin
Preparing data.
Computing bandwidth selectors.
Computing variance-covariance matrix.
Computing RD estimates.
Estimation completed.

Sharp RD estimates using local polynomial regression.
         Cutoff c = 0 | Left of c  Right of c       Number of obs =       1297
       ------------------+---------------------       NN matches    =          3
        Number of obs    |    343         310         BW type       =        CCT
   Order loc. poly. (p)  |      1           1         Kernel type   = Triangular
        Order bias (q)   |      2           2
      BW loc. poly. (h)  |  16.794      16.794
          BW bias (b)    |  27.437      27.437
          rho (h/b)      |   0.612       0.612

Outcome: vote. Running Variable: margin.
----------------------------------------------------------------------------------
          Method |   Coef.   Std. Err.     z     P>|z|     [95% Conf. Interval]
-----------------+----------------------------------------------------------------
    Conventional |  7.4253    1.4954    4.9656   0.000      4.49446    10.3561
          Robust |     -         -      4.2675   0.000      4.06975    10.9833
----------------------------------------------------------------------------------
```

These results contain a variety of information organized into two panels. The upper panel of the output table contains 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 the following:

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 variance estimators are the robust ones proposed by Calonico, Cattaneo, and Titiunik (2014d), computed using three nearest neighbors.

2. The bandwidth selection procedure is the one proposed by Calonico, Cattaneo, and Titiunik (2014d) leading to

$$\widehat{h}_{\text{CCT},n,p} = 16.794 \qquad \text{and} \qquad \widehat{b}_{\text{CCT},n,p+1,q} = 27.437$$

   with $p = 1$ and $q = 2$.

3. The point estimator and robust CI are

$$\widehat{\tau}_p\left(\widehat{h}_{\text{CCT},n,1}\right) = 7.4253, \qquad \widehat{\text{CI}}_{1-\alpha}^{\text{rbc}}\left(\widehat{h}_{\text{CCT},n,1}, \widehat{b}_{\text{CCT},n,2,2}\right) = [\, 4.06975, 10.9833 \,]$$

   with $\alpha = 0.05$.

The command `rdrobust` also offers a more detailed output, which includes all the point estimators, variance estimators, and CIs discussed in section 2. These results are retrieved by including the option `all`, as follows:

```
. rdrobust vote margin, all
Preparing data.
Computing bandwidth selectors.
Computing variance-covariance matrix.
Computing RD estimates.
Estimation completed.

Sharp RD estimates using local polynomial regression.
         Cutoff c = 0 | Left of c  Right of c         Number of obs =        1297
                       |                               NN matches    =           3
         Number of obs |      343         310          BW type       =         CCT
   Order loc. poly. (p)|        1           1          Kernel type   = Triangular
        Order bias (q) |        2           2
    BW loc. poly. (h)  |   16.794      16.794
          BW bias (b)  |   27.437      27.437
           rho (h/b)   |    0.612       0.612

Outcome: vote. Running variable: margin.

              Method |   Coef.   Std. Err.    z     P>|z|   [95% Conf. Interval]

        Conventional | 7.4253     1.4954   4.9656   0.000    4.49446    10.3561
              Robust |    -          -     4.2675   0.000    4.06975    10.9833

All estimates.

              Method |   Coef.   Std. Err.    z     P>|z|   [95% Conf. Interval]

        Conventional | 7.4253     1.4954   4.9656   0.000    4.49446    10.3561
      Bias-corrected | 7.5265     1.4954   5.0333   0.000    4.59569    10.4574
              Robust | 7.5265     1.7637   4.2675   0.000    4.06975    10.9833
```

This detailed output contains an additional table located at the bottom, relative to the default output. Using our notation and the options specified in the case above, this extra table follows the structure of table 1:

Table 1. Bottom table structure in output of `rdrobust ..., all`

Method	Coef.	Std. err.	z	P> |z|	[95% CI]
Conventional	$\widehat{\tau}_p(h_n)$	$\sqrt{\widehat{V}_p/nh_n}$	·	·	$\widehat{\text{CI}}_{1-\alpha}(h_n)$
Bias corrected	$\widehat{\tau}_{p,q}^{\text{bc}}(h_n, b_n)$	$\sqrt{\widehat{V}_p/nh_n}$	·	·	$\widehat{\text{CI}}_{1-\alpha}^{\text{bc}}(h_n, b_n)$
Robust	$\widehat{\tau}_{p,q}^{\text{bc}}(h_n, b_n)$	$\sqrt{\widehat{V}_{n,p,q}^{\text{bc}}}$	·	·	$\widehat{\text{CI}}_{1-\alpha}^{\text{rbc}}(h_n, b_n)$

Note: In the output above, $p = 1$, $q = 2$, $h_n = \widehat{h}_{\text{CCT},n,p}$, $b_n = \widehat{b}_{\text{CCT},n,p+1,q}$, and $\alpha = 0.05$.

Finally, we explore all the bandwidth selection procedures contained in our package. Specifically, we can use our companion `rdbwselect` to compare the CCT bandwidth selectors with the IK and CV bandwidth selectors:

```
. rdbwselect vote margin, all
Computing CCT bandwidth selector.
Computing IK bandwidth selector.
Computing CV bandwidth selector.
Bandwidth estimators for RD local polynomial regression
```

Cutoff c = 0	Left of c	Right of c
Number of obs	595	702
Order loc. poly. (p)	1	1
Order bias (q)	2	2
Range of margin	99.921	99.964

Number of obs	=	1297
NN matches	=	3
Kernel type	=	Triangular
Min BW grid	=	0.69039
Max BW grid	=	99.92107
Length BW grid	=	4.96153

Method	h	b	rho
CCT	16.79369	27.43745	.612072
IK	15.75038	16.47286	.956141
CV	35.42113	NA	NA

Here we used the option `all`, which computes the three bandwidth selectors briefly discussed above. In the case of h_n, these choices range from 16.79369 to 35.42113. In the case of b_n, we obtain $\widehat{b}_{\text{CCT},n,2,2} = 27.43745$ and $\widehat{b}_{\text{IK},n,2,2} = 16.47286$ for the CCT and IK methods, respectively. 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 in figure 5. This is done using the option `cvplot` as follows (in this example, we also changed the grid features to obtain a better plot and to show this additional functionality in action):

```
. rdbwselect vote margin, bwselect(CV) cvplot cvgrid_min(10) cvgrid_max(80)
Computing CV bandwidth selector.
Bandwidth estimators for RD local polynomial regression
```

Cutoff c = 0	Left of c	Right of c
Number of obs	595	702
Order loc. poly. (p)	1	1
Order bias (q)	2	2
Range of margin	99.921	99.964

Number of obs	=	1297
NN matches	=	3
Kernel type	=	Triangular
Min BW grid	=	10.00000
Max BW grid	=	80.00000
Length BW grid	=	3.50000

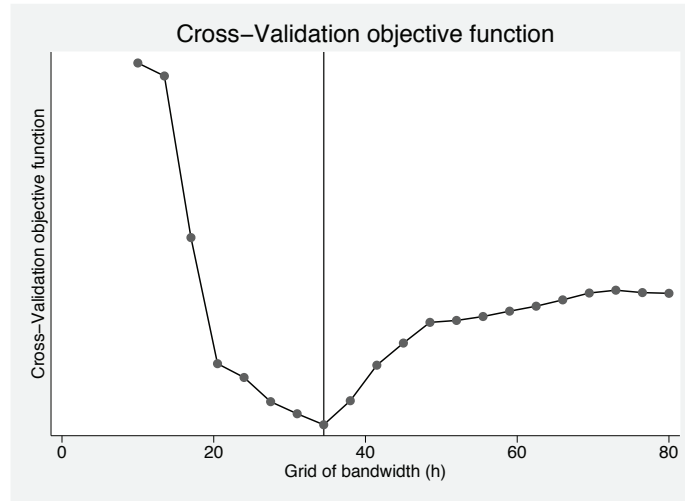| Method	h	b	rho
CV	34.5	NA	NA

Figure 5. Values of the CV function over the grid selected

As discussed above, our commands have many other options. For example, for the main command `rdrobust`, we have the following additional examples (output is not provided to conserve space):

1. `rdrobust vote margin, kernel(uniform)`
   Estimation using uniform kernel.

2. `rdrobust vote margin, bwselect(IK)`
   Estimation using the IK bandwidth selectors.

3. `rdrobust vote margin, bwselect(CV)`
   Estimation using the CV bandwidth selector (and $\rho = 1$).

4. `rdrobust vote margin, h(15) rho(0.8)`
   Estimation using $h_n = 15$ and $b_n = h_n/\rho = 15/0.8 = 18.75$.

5. `rdrobust vote margin, p(2) q(4)`
   Estimation using $p = 2$ and $q = 4$.

6. `rdrobust vote margin, vce(resid)`
   Estimation using plug-in residuals estimates in the variance–covariance estimator.

Finally, our commands can also be used to conduct inference in other RD design settings. For example, let's assume y is the output variable, t is the treatment status variable, and x is the running variable:

1. rdrobust y x, deriv(1)
   Estimation for sharp kink RD.

2. rdrobust y x, fuzzy(t)
   Estimation for fuzzy RD.

3. rdrobust y x, fuzzy(t) deriv(1)
   Estimation for fuzzy kink RD.

6.1 Generating tables with outreg2

The output generated by rdrobust can be used to construct tables with available Stata packages because the main estimates and chosen parameters are stored as e() results. This can facilitate the creation of tables and postestimation handling of numerical results, but it should not be used to perform joint or other postestimation hypothesis tests among the coefficients in the stored matrices.

To illustrate how to produce output tables with outreg2 (Wada 2005), we consider the following three estimation specifications:

```
. rdrobust vote margin, kernel(uniform)
. rdrobust vote margin, bwselect(CV)
. rdrobust vote margin, p(2) q(3)
```

Then we can easily obtain the following table in LATEX format:

| | (1) | (2) | (3) |
VARIABLES	vote	vote	vote
RD_Estimate	7.403	7.167	7.943
Observations	489	1013	828
Robust 95% CI	[4.16 ; 11.46]	[4.42 ; 10.59]	[4.03 ; 11.99]
Kernel Type	Uniform	Triangular	Triangular
BW Type	CCT	CV	CCT
Conventional Std. Error	1.640	1.103	1.818
Conventional p-value	0.000	0.000	0.000
Robust p-value	0.000	0.000	0.000
Order Loc. Poly. (p)	1.000	1.000	2.000
Order Bias (q)	2.000	2.000	3.000
BW Loc. Poly. (h)	11.095	35.421	24.328
BW Bias (b)	22.790	35.421	35.572

This table is constructed using the following command after each specification:

```
outreg2 using table1, addstat(Conventional Std. Err., e(se_cl),
Conventional p-value, e(pv_cl), Robust p-value, e(pv_rb), Order Loc.
Poly. (p), e(p), Order Bias (q), e(q), BW Loc. Poly. (h), e(h_bw), BW
Bias (b), e(b_bw)) addtext(Robust`e(level)´% CI, `e(ci_rb)´, Kernel
Type, `e(kernel)´, BW Type, `e(bwsel)´, Observations, `e(N)´)´ noobs nose
noaster nonotes adec(3)
```

The appropriate additional options (for example, `replace`, `append`, `tex`, etc.) can be used as needed. Additional statistics can also be included. The complete list of estimation results can be obtained by running `ereturn list` after `rdrobust`.

Finally, we can also construct tables in a more standard format if we use `outreg2` after running `rdrobust` with the `all` option. For example, for the same specifications as before, we obtain the following table in LaTeX format:

VARIABLES	(1) vote	(2) vote	(3) vote
Conventional	7.403***	7.167***	7.943***
	(1.640)	(1.103)	(1.818)
Bias-corrected	7.810***	7.502***	8.006***
	(1.640)	(1.103)	(1.818)
Robust	7.810***	7.502***	8.006***
	(1.861)	(1.575)	(2.031)
Observations	489	1,013	828

Standard errors in parentheses
*** p<0.01, ** p<0.05, * p<0.1

Our companion replication file (`rdrobust_illustration.do`) includes the exact syntax of all the examples discussed above.

7 Conclusion

In this article, we discussed the implementation of several data-driven local-polynomial-based (robust) inference procedures in the RD design. We introduced the commands `rdrobust`, `rdbwselect`, and `rdplot`, which together offer an array of data-driven nonparametric inference methods for performing empirical work in RD applications. Our implementations cover average treatment effects at the cutoff in the sharp RD, sharp kink RD, fuzzy RD, and fuzzy kink RD designs, among other possibilities.

A companion R package is also available—see Calonico, Cattaneo, and Titiunik (2014b) for a description.

8 Acknowledgments

9 References

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

Calonico, S., M. D. Cattaneo, and M. H. Farrell. 2014. On the effect of bias estimation on coverage accuracy in nonparametric estimation. Working Paper, University of Michigan. http://www-personal.umich.edu/~cattaneo/papers/Calonico-Cattaneo-Farrell_2013_wp.pdf.

Calonico, S., M. D. Cattaneo, and R. Titiunik. 2014a. Optimal data-driven regression discontinuity plots. Working Paper, University of Michigan. http://www-personal.umich.edu/~cattaneo/papers/RD-rdplot.pdf.

———. 2014b. rdrobust: An R package for robust inference in regression-discontinuity designs. Working Paper, University of Michigan. http://www-personal.umich.edu/~cattaneo/papers/Calonico-Cattaneo-Titiunik_2014_Rpkg.pdf.

———. 2014c. Supplement to Robust nonparametric confidence intervals for regression-discontinuity designs. *Econometica Supplementary Materials*.

———. 2014d. Robust nonparametric confidence intervals for regression-discontinuity designs. *Econometrica* 82: 2295–2326.

Card, D., D. S. Lee, Z. Pei, and A. Weber. 2014. Inference on causal effects in a generalized regression kink design. Working Paper, University of California–Berkeley.

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

Cattaneo, M. D., B. Frandsen, and R. Titiunik. Forthcoming. Randomization inference in the regression discontinuity design: An application to party advantages in the U.S. Senate. *Journal of Causal Inference*.

Dinardo, J., and D. S. Lee. 2011. Program evaluation and research designs. In *Handbook of Labor Economics*, ed. O. Ashenfelter and D. Card, vol. 4A, 463–536. Amsterdam: Elsevier.

Dong, Y. 2011. Jumpy or kinky? Regression discontinuity without the discontinuity. Working Paper, University of California–Irvine. http://www.economics.uci.edu/files/economics/docs/workingpapers/2011-2012/dong-07.pdf.

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

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

Heckman, J. J., and E. J. Vytlacil. 2007. Econometric evaluation of social programs, part I: Causal models, structural models and econometric policy evaluation. In *Handbook of Econometrics*, ed. J. J. Heckman and E. Leamer, vol. 6B, 4779–4874. Amsterdam: Elsevier.

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

Imbens, G. W., and T. Lemieux. 2008. Regression discontinuity designs: A guide to practice. *Journal of Econometrics* 142: 615–635.

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

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

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

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

Porter, J. 2003. Estimation in the regression discontinuity model. Working Paper, University of Wisconsin. http://www.ssc.wisc.edu/~jrporter/reg_discont_2003.pdf.

Wada, R. 2005. outreg2: Stata module to arrange regression outputs into an illustrative table. Statistical Software Components S456416, Department of Economics, Boston College. http://ideas.repec.org/c/boc/bocode/s456416.html.

Wand, M. P., and M. C. Jones. 1995. *Kernel Smoothing*. New York: Chapman & Hall/CRC.

About the authors

Sebastian Calonico is an assistant professor of economics at the University of Miami.

Matias D. Cattaneo is an associate professor of economics at the University of Michigan.

Rocío Titiunik is an assistant professor of political science at the University of Michigan.