Honest inference in Regression Discontinuity Designs

Michal Kolesár

November 07, 2026

Contents

1 Introduction **1**

2 Sharp RD **2**
2.1 Model . . . 2
2.2 Plots . . . 2
2.3 Inference based on local polynomial estimates . . . 3
2.4 Automatic bandwidth choice . . . 6
2.5 Choice of curvature parameter . . . 8
2.6 Inference when running variable is discrete . . . 9

3 Fuzzy RD **12**
3.1 Model . . . 12
3.2 Inference based on local polynomial estimates . . . 12
3.3 Choice of curvature parameters . . . 13

4 Extensions **14**
4.1 Covariates . . . 14
4.2 Aggregated data and weighted regression . . . 19
4.3 Clustering . . . 21
4.4 Specification testing . . . 22
4.5 Optimal weights under Taylor smoothness class . . . 23

5 Inference at a point **23**

6 Diagnostics: leverage and effective observations **24**

1 Introduction

The package `RDHonest` implements bias-aware inference methods in regression discontinuity (RD) designs developed in Armstrong and Kolesár [2018], Armstrong and Kolesár [2020], and Kolesár and Rothe [2018]. In this vignette, we demonstrate the implementation of these methods using datasets from Lee [2008], Oreopoulos [2006], and Battistin et al. [2009] and Ludwig and Miller [2007], which are included in the package as a data frame `lee08`, `cghs`, `rcp` and `headst`. The dataset

from Lalive [2008] used in Kolesár and Rothe [2018] is also included in the package as data frame `rebp`.

2 Sharp RD

2.1 Model

We observe units $i = 1, \ldots, n$, with the outcome Y_i for the ith unit given by

$$Y_i = f_Y(x_i) + u_{Y,i} \tag{1}$$

where $f_Y(x_i)$ is the expectation of Y_i conditional on the running variable x_i and $u_{Y,i}$ is the regression error that is conditionally mean zero by definition.

A unit is assigned to treatment if and only if the running variable x_i lies weakly above a known cutoff. We denote the assignment indicator by $Z_i = I\{x_i \geq c_0\}$. In a sharp RD design, all units comply with the assigned treatment, so that the observed treatment coincides with treatment assignment, $D_i = Z_i$. The parameter of interest is given by the jump of f at the cutoff,

$$\tau_Y = \lim_{x \downarrow c_0} f_Y(x) - \lim_{x \uparrow c_0} f_Y(x).$$

Under mild continuity conditions, τ_Y can be interpreted as the effect of the treatment for units at the threshold (Hahn et al. [2001]). Let $\sigma^2_Y(x_i)$ denote the conditional variance of Y_i.

In the Lee [2008] dataset `lee08`, the running variable corresponds to the margin of victory of a Democratic candidate in a US House election, and the treatment corresponds to winning the election. Therefore, the cutoff is zero. The outcome of interest is the Democratic vote share in the following election.

The `cghs` dataset from Oreopoulos [2006] consists of a subsample of British workers. The RD design exploits a change in minimum school-leaving age in the UK from 14 to 15, which occurred in 1947. The running variable is the year in which the individual turned 14, with the cutoff equal to 1947 so that the "treatment" is being subject to a higher minimum school-leaving age. The outcome is log earnings in 1998.

2.2 Plots

The package provides a function `RDScatter` to plot the raw data. To remove some noise, the function plots averages over `avg` number of observations.

```
library("RDHonest")
## plot 50-bin averages in for observations 50 at most
## points away from the cutoff. See Figure 1.
RDScatter(voteshare ~ margin, data = lee08, subset = abs(lee08$margin) <=
    50, avg = 50, propdotsize = FALSE, xlab = "Margin of victory",
    ylab = "Vote share in next election")
```

The running variable in the Oreopoulos dataset is discrete. It is therefore natural to plot the average outcome by each value of the running variable, which is achieved using by setting `avg=Inf`. The option `propdotsize=TRUE` makes the size of the points proportional to the number of observations that the point averages over.

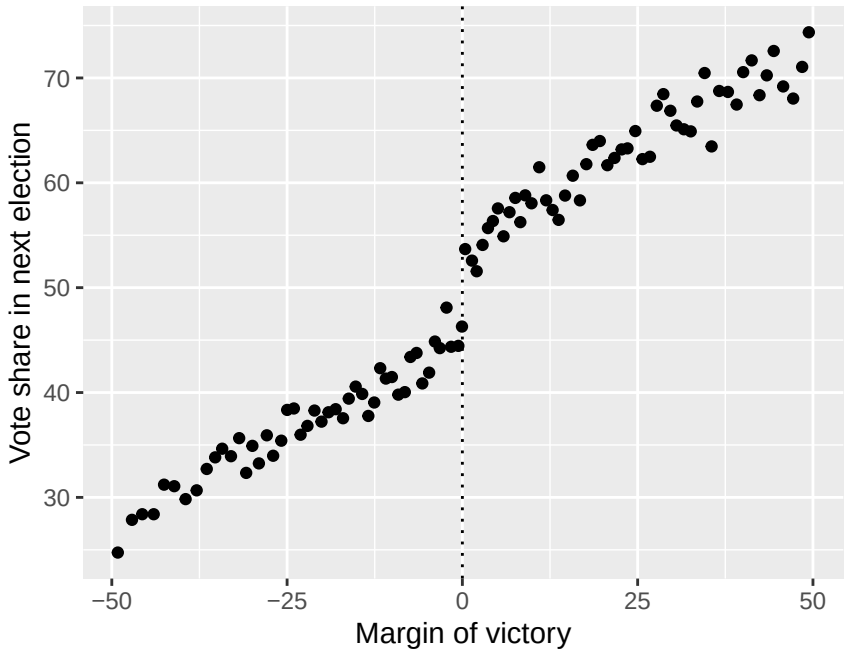


Figure 1: Lee (2008) data

```
## see Figure 2
f2 <- RDScatter(log(earnings) ~ yearat14, data = cghs, cutoff = 1947,
    avg = Inf, xlab = "Year aged 14", ylab = "Log earnings",
    propdotsize = TRUE)
## Adjust size of dots if they are too big
f2 + ggplot2::scale_size_area(max_size = 4)
```

2.3 Inference based on local polynomial estimates

The function `RDHonest` constructs one- and two-sided confidence intervals (CIs) around local linear estimators using either a user-supplied bandwidth, or bandwidth that is optimized for a given performance criterion. The sense of honesty is that, if the regression errors are normally distributed with known variance, the CIs are guaranteed to achieve correct coverage *in finite samples*, and achieve correct coverage asymptotically uniformly over the parameter space otherwise. Furthermore, because the CIs explicitly take into account the possible bias of the estimators, the asymptotic approximation does not rely on the bandwidth to shrinking to zero at a particular rate—in fact, the CIs are valid even if the bandwidth is fixed as $n \to \infty$.

We first estimate τ_Y using local linear regression: instead of using all available observations, we only use observations within a narrow estimation window around the cutoff, determined by a bandwidth h. Within this estimation window, we may choose to give more weight to observations closer to the cutoff—this weighing is determined a kernel K. The local linear regression is just a weighted least squares (WLS) regression of Y_i onto the treatment indicator, the running variable, and their interaction, weighting each observation using kernel weights $K(x_i/h)$. The local linear regression estimator $\hat{\tau}_{Y,h}$ or τ_Y is given by the first element of the vector of regression coefficients

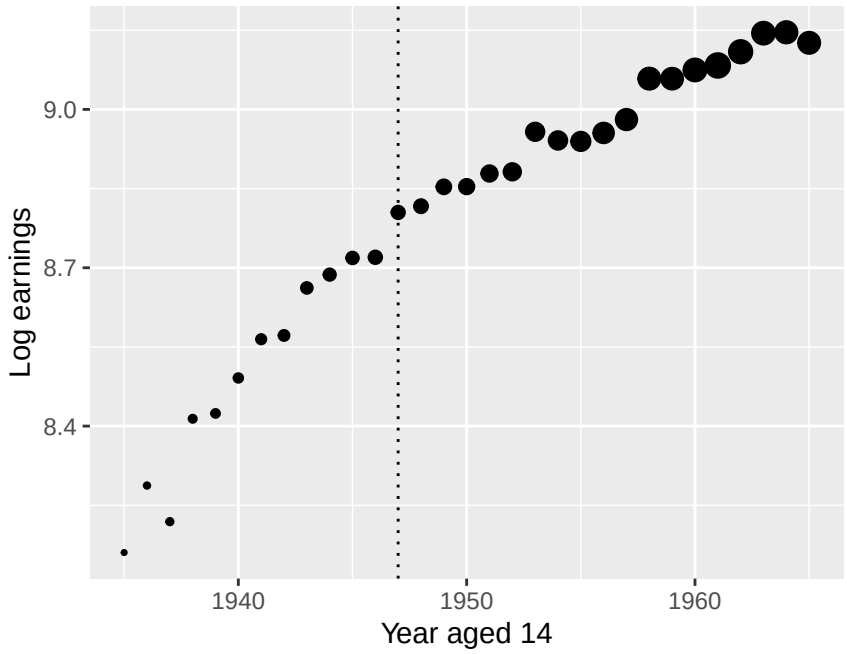


Figure 2: Oreopoulos (2006) data

from this regression,

$$\hat{\beta}_{Y,h} = \left(\sum_{i=1}^{n} K(x_i/h) m(x_i) m(x_i)' \right)^{-1} \sum_{i=1}^{n} K(x_i/h) m(x_i) Y_i,$$

where $m(x) = (I\{x \geq 0\}, I\{x \geq 0\}x, 1, x)'$ collects all the regressors, and we normalize the cutoff to zero. When the kernel is uniform, $K(u) = I\{|u| \leq 1\}$, $\hat{\tau}_{Y,h}$ is simply the treatment coefficient from an OLS regression of Y_i onto $m(x_i)$ for observations that are within distance h of the cutoff. Other kernel choices may weight observations within the estimation window $[-h, h]$ differently, giving more weight to observations that are relatively closer to the cutoff.

Equivalently, using the Frisch–Waugh–Lovell theorem, $\hat{\tau}_{Y,h}$ may also be computed by first running an auxiliary WLS regression of the treatment indicator D_i onto the remaining regressors, $(I\{x_i \geq 0\}x_i, 1, x_i)$ and then running a WLS regression of the outcome Y_i onto the residuals $\tilde{D}_i$ from this auxiliary regression,

$$\hat{\tau}_{Y,h} = \sum_{i=1}^{n} k_{i,h} Y_i, \quad k_{i,h} = \frac{K(x_i/h)\tilde{D}_i}{\sum_{i=1}^{n} K(x_i/h)\tilde{D}_i^2}.$$

This representation makes it clear that the estimator simply a weighted average of the outcomes. By definition of the residual $\tilde{D}_i$, the weights sum to zero, and satisfy $\sum_i k_{i,h} x_i = \sum_i k_{i,h} x_i I\{x_i \geq 0\} = 0$: this ensures that our estimate of the jump at 0 is unbiased when the regression function is piecewise linear inside the estimation window.

2.3.1 Bias-aware confidence intervals

The estimator $\hat{\tau}_{Y,h}$ is a regression estimator, so it will be asymptotically normal under mild regularity conditions. In particular, if the residuals u_i are well-behaved, a sufficient condition is that none of

the weights $k_{i,h}$ are too influential in the sense that the maximal leverage goes to zero, as we discuss in the diagnostics section.

Due to the asymptotic normality, the simplest approach to inference is to use the usual CI, $\hat{\tau}_{Y,h} \pm z_{1-\alpha/2}\widehat{\text{se}}(\hat{\tau}_{Y,h})$, where z_α is the α quantile of the standard normal distribution. However, this CI will typically undercover relative to its nominal confidence level $1-\alpha$ because it's not correctly centered: unless the regression function f_Y is exactly linear inside the estimation window, the estimator $\hat{\tau}_{Y,h}$ will be biased. If f_Y is "close" to linear, then this bias will be small, but if it is wiggly, the bias may be substantial, leading to severe coverage distortions.

The idea behind bias-aware inference methods is to bound the potential bias of the estimator by making an explicit assumption on the smoothness of f_Y. A convenient way of doing this is to bound the curvature of f_Y by restricting its second derivative. To allow f_Y to be discontinuous at zero, we assume that it's twice differentiable on either side of the cutoff, with a second derivative bounded by a known constant M. The choice of the exact curvature parameter M is key to implementing bias-aware methods, and we discuss it below. Once M is selected, we can work out the potential finite-sample bias of the estimator and account for it in the CI construction. In particular, it turns out that the absolute value of the bias of $\hat{\tau}_{Y,h}$ is maximized at the piecewise quadratic function $x \mapsto Mx^2(I\{x<0\} - I\{x \geq 0\})/2$, so that

$$|E[\hat{\tau}_{Y,h} - \tau_Y]| \leq B_{M,h} := -\frac{M}{2}\sum_{i=1}^{n} k_{i,h} x_i^2 \,\text{sign}(x_i).$$

Hence, a simple way to ensure that we achieve correct coverage regardless of the true shape of the regression function f_Y (so long as $|f_Y''(x)| \leq M$) is to simply enlarge the usual CI by this bias bound, leading to the CI $\hat{\tau}_{Y,h} \pm (B_{M,h} + z_{1-\alpha/2}\widehat{\text{se}}(\hat{\tau}_{Y,h}))$. We can actually to slightly better than this, since the bias bound can't simultaneously be binding on both endpoints of the CI. In particular, observe that in large samples, the t-statistic $(\hat{\tau}_{Y,h} - \tau_Y)/\widehat{\text{se}}(\hat{\tau}_{Y,h})$ is normally distributed with variance one, and mean that is bounded by $t = B_{M,h}/\widehat{\text{se}}(\hat{\tau}_{Y,h})$ (ignoring sampling variability in the standard error, which is negligible in large samples). To ensure correct coverage, we therefore replace the usual critical value $z_{1-\alpha/2}$, with the $1-\alpha$ quantile of folded normal distribution $|\mathcal{N}(t,1)|$, $\text{cv}_\alpha(t)$ (note $\text{cv}_\alpha(0) = z_{1-\alpha/2}$). This leads to the bias-aware CI

$$\hat{\tau}_{Y,h} \pm \text{cv}_\alpha(B_{M,h}/\widehat{\text{se}}(\hat{\tau}_{Y,h}))\widehat{\text{se}}(\hat{\tau}_{Y,h}) \tag{2}$$

Notice the bias bound $B_{M,h}$ accounts for the exact *finite-sample bias* of the estimator. The only asymptotic approximation we have used in its construction is the approximate normality of the estimator $\hat{\tau}_{Y,h}$, which obtains without any restrictions on f_Y—we only need the maximal leverage to be close to zero, mirroring a standard leverage condition from parametric regression settings.

The function `CVb` gives the critical values $\text{cv}_\alpha(t)$:

```
CVb(0, alpha = 0.05)  ## Usual critical value
#> [1] 1.959964
CVb(1/2, alpha = 0.05)
#> [1] 2.181477
## Tabulate critical values for different bias levels
CVb(0:5, alpha = 0.1)
#> [1] 1.644854 2.284468 3.281552 4.281552 5.281552 6.281552
```

The function `RDHonest` puts all these steps together. Specifying curvature parameter $M = 0.1$, bandwidth $h = 8$, and a triangular kernel yields:

```
r0 <- RDHonest(voteshare ~ margin, data = lee08, kern = "triangular",
    M = 0.1, h = 8)
print(r0)
#>
#> Call:
#> RDHonest(formula = voteshare ~ margin, data = lee08, M = 0.1,
#>     kern = "triangular", h = 8)
#>
#> Inference for Sharp RD parameter (using Holder class), confidence level 95%:
#>             Estimate Std. Error Maximum Bias  Confidence Interval
#> I(margin>0) 5.873853   1.348925    0.6706413 (2.934244, 8.813462)
#>
#> Onesided CIs:  (-Inf, 8.763279), (2.984427, Inf)
#> Number of effective observations: 793.5835
#> Maximal leverage for sharp RD parameter: 0.009168907
#> Smoothness constant M:      0.1
#> P-value: 5.793498e-05
#>
#> Based on local regression with bandwidth:        8, kernel: triangular
#> Regression coefficients:
#>        I(margin>0)  I(margin>0):margin         (Intercept)              margin
#>             5.8739               0.1448             46.2830              0.6062
```

- The default for calculating the standard errors is to use the nearest neighbor method. Specifying `se.method="EHW"` changes them to the regression-based heteroskedasticity-robust Eicker-Huber-White standard errors. It can be shown that unless the regression function f_Y is linear inside the estimation window, the `EHW` standard errors generally overestimate the conditional variance.
- The default option `sclass="H"` specifies the parameter space as second-order Hölder smoothness class, which formalizes our assumption above that the second derivative of f_Y is bounded by M on either side of the cutoff. The package also allows the user to use a Taylor smoothness class by setting `sclass="T"`. This changes the computation of the worst-case bias, and allows f_Y to correspond to any function such that the approximation error from a second-order Taylor expansion around the cutoff is bounded by $Mx^2/2$. For more discussion, see Section 2 in Armstrong and Kolesár [2018] (note the constant C in that paper equals $C = M/2$ here).
- Other options for `kern` are `"uniform"` and `"epanechnikov"`, or the user can also supply their own kernel function.
- `RDHonest` reports two-sided as well one-sided CIs. One-sided CIs simply subtract off the worst-case bias bound $B_{M,h}$ in addition to subtracting the standard error times the $z_{1-\alpha}$ critical value from the estimate. It also reports the p-value for the hypothesis that $\tau_Y = 0$.
- `RDHonest` also reports the fitted regression coefficients $\hat{\beta}_{Y,h}$, and returns the `lm` object under `r0$lm`. We see from the above that the fitted slopes below and above the cutoff differ by 0.14, for instance.

2.4 Automatic bandwidth choice

Instead of specifying a bandwidth, one can just specify the curvature parameter M, and the bandwidth will be chosen optimally for a given optimality criterion—minimizing the worst-case

MSE of the estimator, or minimizing the length the resulting confidence interval. Typically, this makes little difference:

```
RDHonest(voteshare ~ margin, data = lee08, kern = "triangular",
    M = 0.1, opt.criterion = "MSE")
#>
#> Call:
#> RDHonest(formula = voteshare ~ margin, data = lee08, M = 0.1,
#>     kern = "triangular", opt.criterion = "MSE")
#>
#> Inference for Sharp RD parameter (using Holder class), confidence level 95%:
#>             Estimate Std. Error Maximum Bias  Confidence Interval
#> I(margin>0) 5.936649   1.294421    0.8322588 (2.954829, 8.918469)
#>
#> Onesided CIs:  (-Inf,  8.89804), (2.975258, Inf)
#> Number of effective observations: 889.0468
#> Maximal leverage for sharp RD parameter: 0.008236381
#> Smoothness constant M:      0.1
#> P-value: 4.025594e-05
#>
#> Based on local regression with bandwidth: 8.848512, kernel: triangular
#> Regression coefficients:
#>        I(margin>0)  I(margin>0):margin         (Intercept)              margin
#>             5.9366              0.1175             46.2826              0.6058
## Choose bws optimal for length of CI
RDHonest(voteshare ~ margin, data = lee08, kern = "triangular",
    M = 0.1, opt.criterion = "FLCI")
#>
#> Call:
#> RDHonest(formula = voteshare ~ margin, data = lee08, M = 0.1,
#>     kern = "triangular", opt.criterion = "FLCI")
#>
#> Inference for Sharp RD parameter (using Holder class), confidence level 95%:
#>             Estimate Std. Error Maximum Bias  Confidence Interval
#> I(margin>0) 5.954455   1.278777    0.8833915 (2.952762, 8.956147)
#>
#> Onesided CIs:  (-Inf, 8.941247), (2.967662, Inf)
#> Number of effective observations: 917.7729
#> Maximal leverage for sharp RD parameter: 0.007988637
#> Smoothness constant M:      0.1
#> P-value: 3.665694e-05
#>
#> Based on local regression with bandwidth:  9.11113, kernel: triangular
#> Regression coefficients:
#>        I(margin>0)  I(margin>0):margin         (Intercept)              margin
#>             5.9545              0.1099             46.2825              0.6058
```

To compute the optimal bandwidth, the package assumes homoskedastic variance on either side of

the cutoff, which it estimates based on a preliminary local linear regression using the Imbens and Kalyanaraman [2012] bandwidth selector. This homoskedasticity assumption is dropped when the final standard errors are computed.

Notice that, when the variance function $\sigma_Y^2(x)$ is known, neither the conditional variance of the estimator, $\text{sd}(\hat{\tau}_{Y,h})^2 = \sum_{i=1}^n k_{i,h}^2 \sigma_Y^2(x_i)$, nor the bias bound $B_{M,h}$ depend on the outcome data. Therefore, the MSE and the length of the infeasible CI, $2\,\text{cv}_\alpha(B_{M,h}/\,\text{sd}(\hat{\tau}_{Y,h}))\,\text{sd}(\hat{\tau}_{Y,h})$, do not depend on the outcome data. To stress this property, we refer to this infeasible CI (and, with some abuse of terminology, also the feasible version in eq. (2)) as a fixed-length confidence interval (FLCI). As a consequence of this property, optimizing the bandwidth for CI length does not impact the coverage of the resulting CI.

2.5 Choice of curvature parameter

The curvature parameter M is the most important implementation choice. It would be convenient if one could use data-driven methods to automate its selection. Unfortunately, if one only assume that the second derivative of f_Y is bounded by some constant M, it is not possible to do that: one cannot use data to select M without distorting coverage (Low [1997], Armstrong and Kolesár [2018]). This result is essentially an instance of the general issue with using pre-testing or using model selection rules, such as using cross-validation or information criteria like AIC or BIC to pick which controls to include in a regression: doing so leads to distorted confidence intervals. Here the curvature parameter M indexes the size of the model: a large M is the analog of saying that all available covariates need to be included in the model to purge omitted variables bias; a small M is the analog of saying that a small subset of them will do. Just like one needs to use institutional knowledge of the problem at hand to decide which covariates to include in a regression, ideally one uses problem-specific knowledge to select M. Analogous to reporting results based on different subsets of controls in columns of a table with regression results, one can vary the choice of M by way of sensitivity analysis.

Depending on the problem at hand, it may be difficult to translate problem-specific intuition about how close we think the regression function is to a linear function into a statement about the curvature parameter M. In such cases, it is convenient to have a rule of thumb for selecting M using the data. To do this, we need to impose additional restrictions on f_Y besides assuming that its second derivative is bounded in order to get around the results on impossibility of post-model selection inference discussed above. An appealing way of doing this is to relate the assumption about the local smoothness of f_Y at the cutoff point, which drives the bias of the estimator but is difficult to measure in the data, to the global smoothness of f_Y, which is much easier to measure. In our implementation, we measure global smoothness by fitting a global quartic polynomial $\check{f}$, separately on either side of the cutoff, following Armstrong and Kolesár [2020]. We assume the local second derivative of f_Y, M, is no larger than the maximum second derivative of the global polynomial approximation $\check{f}$. Under this assumption, we can calibrate M by setting

$$\hat{M}_{ROT} = \sup_{x \in [x_{\min}, x_{\max}]} |\check{f}''(x)|.$$

There are different ways of relating local and global smoothness, which lead to different calibrations of M. For instance, Imbens and Wager [2019] propose fitting a global quadratic polynomial instead, and then multiplying the maximal curvature of the fitted model by a constant such as 2 or 4. An important question for future research is to figure out whether there is a way of relating local and

global smoothness that is empirically appealing across a wide range of scenarios. See also Noack and Rothe [2021] for a discussion how to visualize the choice of M to aid with its interpretation.

When the user doesn't supply `M`, the package uses the rule of thumb $\hat{M}_{ROT}$, and prints a message to inform the user:

```
## Data-driven choice of M
RDHonest(voteshare ~ margin, data = lee08)
#> Using Armstrong & Kolesar (2020) ROT for smoothness constant M.
#>
#> Call:
#> RDHonest(formula = voteshare ~ margin, data = lee08)
#>
#> Inference for Sharp RD parameter (using Holder class), confidence level 95%:
#>             Estimate Std. Error Maximum Bias  Confidence Interval
#> I(margin>0) 5.849736   1.365882    0.8880143 (2.694435, 9.005036)
#>
#> Onesided CIs:  (-Inf, 8.984425), (2.715046, Inf)
#> Number of effective observations: 764.5629
#> Maximal leverage for sharp RD parameter: 0.009560827
#> Smoothness constant M: 0.1428108
#> P-value: 0.0001406869
#>
#> Based on local regression with bandwidth: 7.715099, kernel: triangular
#> Regression coefficients:
#>        I(margin>0)  I(margin>0):margin         (Intercept)              margin
#>             5.8497              0.1218             46.3188              0.6235
```

2.6 Inference when running variable is discrete

The confidence intervals described above can also be used when the running variable is discrete, with G support points: their construction makes no assumptions on the nature of the running variable (see Section 5.1 in Kolesár and Rothe [2018] for more detailed discussion).

Units that lie exactly at the cutoff are considered treated, since the definition of treatment assignment is that the running variable lies weakly above the cutoff, $x_i \geq c_0$.

As an example, consider the Oreopoulos [2006] data, in which the running variable is age in years:

```
## Replicate Table 2, column (10) in Kolesar and Rothe
## (2018)
RDHonest(log(earnings) ~ yearat14, cutoff = 1947, data = cghs,
    kern = "uniform", M = 0.04, opt.criterion = "FLCI",
    sclass = "H")
#>
#> Call:
#> RDHonest(formula = log(earnings) ~ yearat14, data = cghs, cutoff = 1947,
#>     M = 0.04, kern = "uniform", opt.criterion = "FLCI", sclass = "H")
#>
#> Inference for Sharp RD parameter (using Holder class), confidence level 95%:
```

```
#>                 Estimate Std. Error Maximum Bias      Confidence Interval
#> I(yearat14>0) 0.07909463 0.06784089   0.04736585 (-0.08061322, 0.2388025)
#>
#> Onesided CIs:  (-Inf, 0.2380488), (-0.07985957, Inf)
#> Number of effective observations:     7424
#> Maximal leverage for sharp RD parameter: 0.0004652917
#> Smoothness constant M:     0.04
#> P-value: 0.3511574
#>
#> Based on local regression with bandwidth:        2, kernel: uniform
#> Regression coefficients:
#>          I(yearat14>0)  I(yearat14>0):yearat14             (Intercept)
#>               0.079095                0.023387                8.721185
#>               yearat14
#>               0.001302
```

In addition, the package provides function `RDHonestBME` that calculates honest confidence intervals under the assumption that the specification bias at zero is no worse at the cutoff than away from the cutoff as in Section 5.2 in Kolesár and Rothe [2018].

```
## Replicate Table 2, column (6), run local linear
## regression (order=1) with a uniform kernel (other
## kernels are not implemented for RDHonestBME)
RDHonestBME(log(earnings) ~ yearat14, cutoff = 1947, data = cghs,
    h = 3, order = 1)
#>
#> Call:
#> RDHonestBME(formula = log(earnings) ~ yearat14, data = cghs,
#>     cutoff = 1947, h = 3, order = 1)
#>
#> Inference for Sharp RD parameter (using BME class), confidence level 95%:
#>                  Estimate Std. Error Maximum Bias      Confidence Interval
#> I(x >= 0)TRUE 0.06488857 0.04902804   0.02229399 (-0.06965587, 0.2019889)
#>
#> Onesided CIs:  (-Inf, 0.1835311), (-0.05160896, Inf)
#> Number of effective observations:    10533
#> Maximal leverage for sharp RD parameter: 0.0004201627
#> P-value: 0.1857787
#>
#> Based on local regression with bandwidth:        3, kernel: uniform
#> Regression coefficients:
#>          (Intercept)               I(x^1)         I(x >= 0)TRUE
#>             8.740207             0.015933                0.064889
#> I(x^1):I(x >= 0)TRUE
#>             0.002151
```

Let us describe the implementation of the variance estimator $\hat{V}(W)$ used to construct the CI following Section 5.2 in Kolesár and Rothe [2018]. Suppose the point estimate is given by the first

element of the regression of the outcome y_i on $m(x_i)$. For instance, local linear regression with uniform kernel and bandwidth h corresponds to $m(x) = I(|x| \leq h) \cdot (I(x > c_0), 1, x, x \cdot I(x > c_0))'$. Let $\theta = Q^{-1}E[m(x_i)y_i]$, where $Q = E[m(x_i)m(x_i)']$, denote the estimand for this regression (treating the bandwidth as fixed), and let $\delta(x) = f(x) - m(x)'\theta$ denote the specification error at x. The RD estimate is given by first element of the least squares estimator $\hat{\theta} = \hat{Q}^{-1}\sum_i m(x_i)y_i$, where $\hat{Q} = \sum_i m(x_i)m(x_i)'$.

Let $w(x_i)$ denote a vector of indicator (dummy) variables for all support points of x_i within distance h of the cutoff, so that $\mu(x_g)$, where x_g is the gth support point of x_i, is given by the gth element of the regression estimand $S^{-1}E[w(x_i)y_i]$, where $S = E[w(x_i)w(x_i)']$. Let $\hat{\mu} = \hat{S}^{-1}\sum_i w(x_i)y_i$, where $\hat{S} = \sum_i w(x_i)w(x_i)'$ denote the least squares estimator. Then an estimate of $(\delta(x_1), \ldots, \delta(x_G))'$ is given by $\hat{\delta}$, the vector with elements $\hat{\mu}_g - x_g\hat{\theta}$.

By standard regression results, the asymptotic distribution of $\hat{\theta}$ and $\hat{\mu}$ is given by

$$\sqrt{n}\begin{pmatrix}\hat{\theta}-\theta\\ \hat{\mu}-\mu\end{pmatrix} \xrightarrow{d} \mathcal{N}(0,\Omega),$$

where

$$\Omega = \begin{pmatrix} Q^{-1}E[(\epsilon_i^2 + \delta(x_i)^2)m(x_i)m(x_i)']Q^{-1} & Q^{-1}E[\epsilon_i^2 m(x_i)w(x_i)']S^{-1} \\ S^{-1}E[\epsilon_i^2 w(x_i)m(x_i)']Q^{-1} & S^{-1}E[\epsilon_i^2 w(x_i)w(x_i)']S^{-1}\end{pmatrix}.$$

Let $\hat{u}_i$ denote the regression residual from the regression of y_i on $m(x_i)$, and let $\hat{e}_i$ denote the regression residuals from the regression of y_i on $w(x_i)$. Then a consistent estimator of the asymptotic variance Ω is given by

$$\hat{\Omega} = n\sum_i T_i T_i', \qquad T_i' = \begin{pmatrix}\hat{u}_i m(x_i)'\hat{Q}^{-1} & \hat{e}_i w(x_i)'\hat{S}^{-1}\end{pmatrix}.$$

Note that the upper left block and lower right block correspond simply to the Eicker-Huber-White estimators of the asymptotic variance of $\hat{\theta}$ and $\hat{\mu}$. By the delta method, a consistent estimator of the asymptotic variance of $(\hat{\delta}, \hat{\theta}_1)$ is given by

$$\hat{\Sigma} = \begin{pmatrix}-X & I \\ e_1' & 0\end{pmatrix}\hat{\Omega}\begin{pmatrix}-X & I \\ e_1' & 0\end{pmatrix}',$$

where X is a matrix with gth row equal to x_g', and e_1 is the first unit vector.

Recall that in the notation of Kolesár and Rothe [2018], $W = (g^-, g^+, s^-, s^+)$, and g^+ and g^- are such that $x_{g^-} < c_0 \leq x_{g^+}$, and $s^+, s^- \in \{-1, 1\}$. An upper limit for a right-sided CI for $\theta_1 + b(W)$ is then given by

$$\hat{\theta}_1 + s^+\hat{\delta}(x_{g^+}) + s^-\hat{\delta}(x_{g^-}) + z_{1-\alpha}\hat{V}(W),$$

where $\hat{V}(W) = a(W)'\hat{\Sigma}a(W)$, and $a(W) \in \mathbb{R}^{G_h+1}$ denotes a vector with the g_-th element equal to s^-, $(G_h^- + g_+)$th element equal to s^+, the last element equal to one, and the remaining elements equal to zero. The rest of the construction then follows the description in Section 5.2 in Kolesár and Rothe [2018].

3 Fuzzy RD

3.1 Model

In a fuzzy RD design, the treatment status D_i of a unit does not necessarily equate the treatment assignment $Z_i = I\{x_i \geq c_0\}$. Instead, the treatment assignment induces a jump in the treatment probability at the cutoff. Correspondingly, we augment the outcome model with a first stage that measures the effect of the running variable on the treatment:

$$Y_i = f_Y(x_i) + u_{Y,i}, \qquad D_i = f_D(x_i) + u_{Y,i}, \tag{3}$$

where f_D, f_Y are the conditional mean functions.

To account for imperfect compliance the fuzzy RD parameter scales the jump in the outcome equation τ_Y by the jump in the treatment probability at the cutoff, $\tau_D = \lim_{x \downarrow c_0} f_D(x) - \lim_{x \uparrow c_0} f_D(x)$. This fuzzy RD parameter, $\theta = \tau_Y / \tau_D$, measures the local average treatment effect for individuals at the threshold who comply with the treatment assignment, provided mild continuity conditions and a monotonicity condition hold (Hahn et al. [2001]). Under perfect compliance, the treatment probability jumps all the way from zero to one at the threshold so that $\tau_D = 1$, and the two parameters coincide.

For example, in the Battistin et al. [2009] dataset, the treatment variable is an indicator for retirement, and the running variable is the number of years since being eligible to retire. The cutoff is 0. Individuals exactly at the cutoff are dropped from the dataset. If there were individuals exactly at the cutoff, they are assumed to receive the treatment assignment (i.e. be eligible for retirement).

3.2 Inference based on local polynomial estimates

A natural estimator for the fuzzy RD parameter θ is the sample analog based on local linear regression,

$$\hat{\theta}_h = \frac{\hat{\tau}_{Y,h}}{\hat{\tau}_{D,h}}.$$

Unlike in the sharp case, our bias-aware CIs do rely on the consistency of the estimator, which generally requires the bandwidth to shrink with the sample size. Since this estimator is a ratio of regression coefficients, it follows by the delta method that so long as $\tau_D \neq 0$, the estimator will be asymptotically normal in large samples. In fact, the estimator is equivalent to a weighted IV regression of Y_i onto D_i, using Z_i as an instrument, and x_i and its interaction with Z_i as controls, so the variance formula is analogous to the IV variance formula:

$$\operatorname{sd}(\hat{\theta}_h)^2 = \frac{\operatorname{sd}(\tau_{Y,h})^2 + \theta^2 \operatorname{sd}(\tau_{D,h})^2 - 2\operatorname{cov}(\tau_{D,h}, \tau_{Y,h})\theta}{\tau_D^2},$$

where $\operatorname{cov}(\tau_{D,h}, \tau_{Y,h}) = \sum_i k_{i,h}^2 \operatorname{cov}(Y_i, D_i \mid x_i)$ is the covariance of the estimators.

If the second derivative of f_Y is bounded by M_Y and the second derivative of f_D is bounded by M_D, a linearization argument from Section 3.2.3 in Armstrong and Kolesár [2020] that the bias can be bounded in large samples by $B_{M,h}$, with $M = (M_Y + |\theta| M_D)/|\tau_D|$, which now depends on θ itself. Therefore, optimal bandwidth calculations will require a preliminary estimate of $|\theta|$, which can be passed to `RDHonest` via the option `T0`. Like in the sharp case, the optimal bandwidth calculations assume homoskedastic covariance of $(u_{Y,i}, u_{D,i})$ on either side of the cutoff, which

are estimated based on a preliminary local linear regression for both the outcome and first stage equation, with bandwidth given by the Imbens and Kalyanaraman [2012] bandwidth selector applied to the outcome equation.

```
## Initial estimate of treatment effect for optimal
## bandwidth calculations
r <- RDHonest(log(cn) | retired ~ elig_year, data = rcp,
    kern = "triangular", M = c(0.001, 0.002), opt.criterion = "MSE",
    sclass = "H", T0 = 0)
## Use it to compute optimal bandwidth
RDHonest(log(cn) | retired ~ elig_year, data = rcp, kern = "triangular",
    M = c(0.001, 0.002), opt.criterion = "MSE", sclass = "H",
    T0 = r$coefficients$estimate)
#>
#> Call:
#> RDHonest(formula = log(cn) | retired ~ elig_year, data = rcp,
#>     M = c(0.001, 0.002), kern = "triangular", opt.criterion = "MSE",
#>     sclass = "H", T0 = r$coefficients$estimate)
#>
#> Inference for Fuzzy RD parameter (using Holder class), confidence level 95%:
#>            Estimate Std. Error Maximum Bias     Confidence Interval
#> retired -0.09062179 0.07162253   0.04374348 (-0.253552, 0.07230844)
#>
#> Onesided CIs:  (-Inf, 0.07093026), (-0.2521738, Inf)
#> Number of effective observations: 7465.396
#> Maximal leverage for fuzzy RD parameter: 0.0008176489
#> First stage estimate: 0.3479478
#> First stage smoothness constant M:    0.002
#> Reduced form smoothness constant M:    0.001
#> P-value: 0.286715
#>
#> Based on local regression with bandwidth: 9.551734, kernel: triangular
#> Regression coefficients:
#>                            log(cn)    retired
#> I(elig_year>0)            -0.031532   0.347948
#> I(elig_year>0):elig_year  -0.005692  -0.009782
#> (Intercept)                9.760643   0.246088
#> elig_year                 -0.005370   0.032019
```

3.3 Choice of curvature parameters

Like in the sharp RD case, without further restrictions, the curvature parameters M_Y and M_D cannot be data-driven: to maintain honesty over the whole function class, a researcher must choose them a priori, rather than attempting to use a data-driven method. Therefore, one should, whenever possible, use problem-specific knowledge to decide what choices of M_Y and M_D are reasonable a priori.

For cases in which this is difficult, the function `RDHonest` implements the rule of thumb Armstrong and Kolesár [2020] described earlier, based on computing the global smoothness of both f_Y and f_D

using a quartic polynomial. When the user doesn't supply the curvature bounds, the package uses the rule of thumb $\hat{M}_{ROT}$, and prints a message to inform the user:

```
## Data-driven choice of M
RDHonest(log(cn) | retired ~ elig_year, data = rcp, kern = "triangular",
    opt.criterion = "MSE", sclass = "H", T0 = r$coefficients$estimate)
#> Using Armstrong & Kolesar (2020) ROT for smoothness constant M.
#>
#> Call:
#> RDHonest(formula = log(cn) | retired ~ elig_year, data = rcp,
#>     kern = "triangular", opt.criterion = "MSE", sclass = "H",
#>     T0 = r$coefficients$estimate)
#>
#> Inference for Fuzzy RD parameter (using Holder class), confidence level 95%:
#>           Estimate Std. Error Maximum Bias      Confidence Interval
#> retired -0.1762098  0.1042236   0.08473369 (-0.4329103, 0.08049077)
#>
#> Onesided CIs:  (-Inf, 0.07995648), (-0.4323761, Inf)
#> Number of effective observations: 4543.727
#> Maximal leverage for fuzzy RD parameter: 0.001114888
#> First stage estimate: 0.3185837
#> First stage smoothness constant M: 0.008178929
#> Reduced form smoothness constant M: 0.002849525
#> P-value: 0.1962011
#>
#> Based on local regression with bandwidth: 6.127523, kernel: triangular
#> Regression coefficients:
#>                           log(cn)    retired
#> I(elig_year>0)           -0.056138   0.318584
#> I(elig_year>0):elig_year -0.009435  -0.019943
#> (Intercept)               9.777239   0.273343
#> elig_year                 0.001841   0.042820
```

See Armstrong and Kolesár [2020] for a discussion of the restrictions on the parameter space under which this method yields honest inference.

4 Extensions

4.1 Covariates

RD datasets often contain information on a vector of K pre-treatment covariates W_i, such as pre-intervention outcomes, demographic, or socioeconomic characteristics of the units. Similar to randomized controlled trials, while the presence of covariates doesn't help to weaken the fundamental identifying assumptions, augmenting the RD estimator with predetermined covariates can increase its precision.

Let us first describe covariate adjustment in the sharp RD case. We implement the covariate adjustment studied in Calonico et al. [2019], namely to include W_i as one of the regressors in the WLS regression, regressing Y_i onto $m(x_i)$ and W_i. As in the case with no covariates, we weight each

observation with the kernel weight $K(x_i/h)$. This leads to the estimator

$$\tilde{\tau}_{Y,h} = \tilde{\beta}_{Y,h,1}, \qquad \tilde{\beta}_{Y,h} = \left(\sum_{i=1}^{n} K(x_i/h)\tilde{m}(x_i, W_i)\tilde{m}(x_i, W_i)'\right)^{-1} \sum_{i=1}^{n} K(x_i/h)\tilde{m}(x_i, W_i)Y_i,$$

where $\tilde{m}(x_i, W_i) = (m(x_i)', W_i')'$. Denote the coefficient on W_i in this regression by $\tilde{\gamma}_{Y,h}$; this corresponds to the last K elements of $\tilde{\beta}_{Y,h}$. As in the case without covariates, we first take the bandwidth h as given, and defer bandwidth selection choice to the end of this subsection.

To motivate the estimator under our framework, and to derive bias-aware CIs that incorporate covariates, we need to formalize the assumption that the covariates are predetermined (without any assumptions on the covariates, it is optimal to ignore the covariates and use the unadjusted estimator $\hat{\tau}_{Y,h}$). Let $f_W(x) = E[W_i \mid X_i = x]$ denote the regression function from regressing the covariates on the running variable, and let

$$\Sigma_{WW}(x) = \text{var}(W_i \mid X_i = x), \qquad \Sigma_{WY}(x) = \text{cov}(W_i, Y_i \mid X_i = x).$$

We assume that the variance and covariance functions are continuous, except possibly at zero. Let $\gamma_Y = (\Sigma_{WW}(0_+) + \Sigma_{WW}(0_-))^{-1}(\Sigma_{WY}(0_+) + \Sigma_{WY}(0_-))$ denote the coefficient on W_i when we regress Y_i onto W_i for observations at the cutoff. Let $\tilde{Y}_i := Y_i - W_i'\gamma_Y$ denote the covariate-adjusted outcome. To formalize the assumption that the covariates are pre-determined, we assume that $\tau_W = \lim_{x\downarrow 0} f_W(0) - \lim_{x\uparrow 0} f_W(0) = 0$, which implies that τ_Y can be identified as the jump in the covariate-adjusted outcome $\tilde{Y}_i$ at 0. Following Appendix B.1 in Armstrong and Kolesár [2018], we also assume that the covariate-adjusted outcome varies smoothly with the running variable (except for a possible jump at the cutoff), in that the second derivative of

$$\tilde{f}(x) := f_Y(x) - f_W(x)'\gamma_Y$$

is bounded by a known constant $\tilde{M}$. In addition, we assume f_W has bounded second derivatives.

Under these assumptions, if γ_Y was known and hence $\tilde{Y}_i$ was directly observable, we could estimate τ as in the case without covariates, replacing M with $\tilde{M}$ and Y_i with $\tilde{Y}_i$. Furthermore, as discussed in Armstrong and Kolesár [2018], such approach would be optimal under homoskedasticity assumptions. Although γ_Y is unknown, it turns out that the estimator $\tilde{\tau}_{Y,h}$ has the same large sample behavior as the infeasible estimator $\hat{\tau}_{\tilde{Y},h}$. To show this, note that by standard regression algebra, $\tilde{\tau}_{Y,h}$ can equivalently be written as

$$\tilde{\tau}_{Y,h} = \hat{\tau}_{Y - W'\tilde{\gamma}_{Y,h},h} = \hat{\tau}_{\tilde{Y},h} - \sum_{k=1}^{K} \hat{\tau}_{W_k,h}(\tilde{\gamma}_{Y,h,k} - \gamma_{Y,k}).$$

The first equality says that covariate-adjusted estimate is the same as an unadjusted estimate that replaces the original outcome Y_i with the covariate-adjusted outcome $Y_i - W_i'\tilde{\gamma}_{Y,h}$. The second equality uses the decomposition $Y_i - W_i'\tilde{\gamma}_{Y,h} = \tilde{Y}_i - W_i'(\tilde{\gamma}_{Y,h} - \gamma_Y)$ to write the estimator as a sum of the infeasible estimator and a linear combination of "placebo RD estimators" $\hat{\tau}_{W_k,h}$, that replace Y_i in the outcome equation with the kth element of W_i, Since f_W has bounded second derivatives, these placebo estimators converge to zero, with rate that is at least as fast as the rate of convergence of the infeasible estimator $\hat{\tau}_{\tilde{Y},h}$: $\hat{\tau}_{W_k,h} = O_p(B_{\tilde{M},h} + \text{sd}(\hat{\tau}_{\tilde{Y},h}))$. Furthermore, under regularity conditions, $\tilde{\gamma}_{Y,h}$ converges to γ_Y, so that the second term in the previous display is asymptotically negligible

relative to the first. Consequently, we can form bias-aware CIs based on $\tilde{\tau}_{Y,h}$ as in the case without covariates, treating the covariate-adjusted outcome $Y_i - W_i'\tilde{\gamma}_Y$ as the outcome,

$$\tilde{\tau}_{Y,h} \pm \text{cv}_{1-\alpha}(B_{\tilde{M},h}/\,\text{sd}(\hat{\tau}_{\tilde{Y},h}))\,\text{sd}(\hat{\tau}_{\tilde{Y},h}), \qquad \text{sd}(\hat{\tau}_{\tilde{Y},h})^2 = \sum_{i=1}^{n} k_{i,h}^2 \sigma_{\tilde{Y}}^2(x_i),$$

where $\sigma_{\tilde{Y}}^2(x_i) = \sigma_Y^2(x_i) + \gamma_Y'\Sigma_{WW}(x_i)\gamma_Y - 2\gamma_Y'\Sigma_{WY}(x_i)$. If the covariates are effective at explaining variation in the outcomes, then the quantity $\sum_i k_{i,h}^2\,(\gamma_Y'\Sigma_{WW}(x_i)\gamma_Y - 2\gamma_Y'\Sigma_{WY}(x_i))$ will be negative, and $\text{sd}(\hat{\tau}_{\tilde{Y},h}) \leq \text{sd}(\hat{\tau}_{Y,h})$. If the smoothness of the covariate-adjusted conditional mean function $f_Y - f_W'\gamma_Y$ is greater than the smoothness of the unadjusted conditional mean function f_Y, so that $\tilde{M} \leq M$, then using the covariates will help tighten the confidence intervals.

Implementation of covariate-adjustment requires a choice of $\tilde{M}$, and computing the optimal bandwidth requires a preliminary estimate of the variance of the covariate-adjusted outcome. In our implementation, we first estimate the model without covariates (using a rule of thumb to calibrate M, the bound on the second derivative of f_Y), and compute the bandwidth $\check{h}$ that's MSE optimal without covariates. Based on this bandwidth, we compute a preliminary estimate $\tilde{\gamma}_{Y,\check{h}}$ of γ_Y, and use this preliminary estimate to compute a preliminary covariate-adjusted outcome $Y_i - W_i'\tilde{\gamma}_{Y,\check{h}}$. If $\tilde{M}$ is not supplied, we calibrate $\tilde{M}$ using the rule of thumb, using this preliminary covariate-adjusted outcome as the outcome. Similarly, we use this preliminary covariate-adjusted outcome as the outcome to compute a preliminary estimator of the conditional variance $\sigma_{\tilde{Y}}^2(x_i)$, for optimal bandwidth calculations, as in the case without covariates. With this choice of bandwidth h, in the second step, we estimate τ_Y using the estimator $\tilde{\tau}_{Y,h}$ defined above.

A demonstration using the `headst` data:

```
## No covariates
rn <- RDHonest(mortHS ~ povrate, data = headst)
#> Using Armstrong & Kolesar (2020) ROT for smoothness constant M.
## Use Percent attending school aged 14-17, urban,
## black, and their interaction as covariates.
rc <- RDHonest(mortHS ~ povrate | urban * black + sch1417,
    data = headst)
#> Using Armstrong & Kolesar (2020) ROT for smoothness constant M.
rn
#>
#> Call:
#> RDHonest(formula = mortHS ~ povrate, data = headst)
#>
#> Inference for Sharp RD parameter (using Holder class), confidence level 95%:
#>              Estimate Std. Error Maximum Bias     Confidence Interval
#> I(povrate>0) -3.15129   1.272338    0.7014739 (-5.980412, -0.3221684)
#>
#> Onesided CIs:  (-Inf, -0.357007), (-5.945573, Inf)
#> Number of effective observations: 256.5127
#> Maximal leverage for sharp RD parameter: 0.0279751
#> Smoothness constant M: 0.2993999
#> P-value: 0.02831734
#>
```

```
#> Based on local regression with bandwidth: 4.880646, kernel: triangular
#> Regression coefficients:
#>         I(povrate>0)  I(povrate>0):povrate           (Intercept)
#>              -3.1513                0.4805                3.7589
#>              povrate
#>               0.2800
#> 24 observations with missing values dropped
rc
#>
#> Call:
#> RDHonest(formula = mortHS ~ povrate | urban * black + sch1417,
#>     data = headst)
#>
#> Inference for Sharp RD parameter (using Holder class), confidence level 95%:
#>               Estimate Std. Error Maximum Bias     Confidence Interval
#> I(povrate>0) -2.678646   1.163838    0.6105652 (-5.240691, -0.1166008)
#>
#> Onesided CIs:  (-Inf, -0.1537367), (-5.203555, Inf)
#> Number of effective observations: 303.9467
#> Maximal leverage for sharp RD parameter: 0.02287772
#> Smoothness constant M: 0.185326
#> P-value: 0.04014376
#>
#> Based on local regression with bandwidth: 5.891514, kernel: triangular
#> Regression coefficients:
#>         I(povrate>0)  I(povrate>0):povrate           (Intercept)
#>           -2.6786458             0.3592431            15.2995726
#>              povrate                 urban                 black
#>            0.1117840             0.0111355             0.0401396
#>              sch1417          urban:black
#>           -0.1553701            -0.0005111
#> 29 observations with missing values dropped
```

We see that the inclusion of covariates leads to a reduction in the rule-of-thumb curvature and also smaller standard errors (this would be true even if the bandwidth was kept fixed). Correspondingly, the CIs are tighter by about 9 percentage points:

```
ci_len <- c(rc$coefficients$conf.high - rc$coefficients$conf.low,
    rn$coefficients$conf.high - rn$coefficients$conf.low)
100 * (1 - ci_len[1]/ci_len[2])
#> [1] 9.440274
```

In the fuzzy RD case, we need to covariate-adjust the treatment D_i as well as the outcome. The implementation mirrors the sharp case. Define γ_D analogously to γ_Y, and assume that the second derivative of $f_Y(x) - f_W(x)'\gamma_Y$ is bounded by a known constant $\tilde{M}_Y$, and that $f_D(x) - f_W(x)'\gamma_D$ is bounded by a known constant $\tilde{M}_D$. The covariate-adjusted estimator is given by $\tilde{\theta}_h = \tilde{\tau}_{Y,h}/\tilde{\tau}_{D,h}$, with variances and worst-case bias computed as in the case without covariates, replacing the treatment and outcome with their covariate-adjusted versions.

A demonstration using the `rcp` data, where we add education controls:

```
RDHonest(log(cn) | retired ~ elig_year | education, data = rcp,
    T0 = r$coefficients$estimate)
#> Using Armstrong & Kolesar (2020) ROT for smoothness constant M.
#>
#> Call:
#> RDHonest(formula = log(cn) | retired ~ elig_year | education,
#>     data = rcp, T0 = r$coefficients$estimate)
#>
#> Inference for Fuzzy RD parameter (using Holder class), confidence level 95%:
#>         Estimate Std. Error Maximum Bias      Confidence Interval
#> retired -0.26109   0.113415    0.1029379 (-0.5508736, 0.02869354)
#>
#> Onesided CIs:  (-Inf, 0.02839886), (-0.5505789, Inf)
#> Number of effective observations: 3259.425
#> Maximal leverage for fuzzy RD parameter: 0.001393073
#> First stage estimate: 0.316514
#> First stage smoothness constant M: 0.007191008
#> Reduced form smoothness constant M: 0.005370042
#> P-value: 0.08225438
#>
#> Based on local regression with bandwidth: 5.069605, kernel: triangular
#> Regression coefficients:
#>                                log(cn)    retired
#> I(elig_year>0)                -0.082639   0.316514
#> I(elig_year>0):elig_year      -0.031593  -0.008402
#> (Intercept)                    9.378002   0.515120
#> elig_year                      0.025771   0.035794
#> educationelementary school     0.275151  -0.201655
#> educationlower secondary       0.431725  -0.287265
#> educationvocational studies    0.580132  -0.360524
#> educationupper secondary       0.741646  -0.294614
#> educationcollege or higher     0.923659  -0.424387
```

Relative to the previous estimate without covariates, the point estimate is now much larger. This is in part due to slightly smaller bandwidth used, and the regression function for the reduced form appears noisy below the cutoff, potentially due to measurement error: see Figure 3. As a result, the estimates are quite sensitive to the bandwidth used. The noise is also responsible for the rather large data-driven estimates of the curvature parameters.

```
## see Figure 3
f3 <- RDScatter(log(cn) ~ elig_year, data = rcp, cutoff = 0,
    avg = Inf, xlab = "Years to eligibility", ylab = "Log consumption of non-durables",
    propdotsize = TRUE, subset = abs(elig_year) < 15)
## Adjust size of dots if they are too big
f3 + ggplot2::scale_size_area(max_size = 5)
```

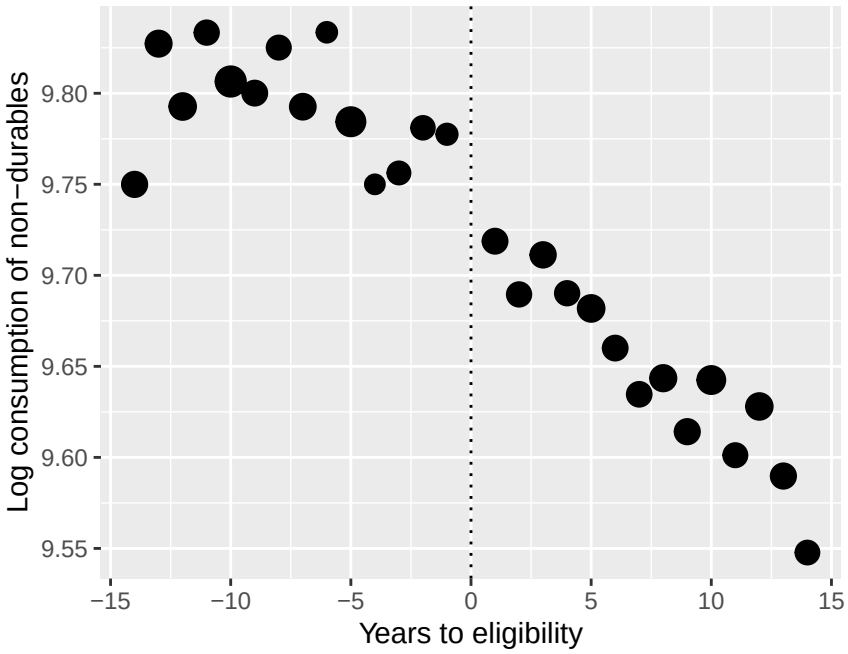


Figure 3: Battistin et al (2009) data

4.2 Aggregated data and weighted regression

In some cases, data is only observed as cell averages. For instance, suppose that instead of observing the original `cghs` data, we only observe averages for cells as follows:

```
dd <- data.frame()
## Collapse data by running variable
for (j in unique(cghs$yearat14)) {
    ix <- cghs$yearat14 == j
    df <- data.frame(y = mean(log(cghs$earnings[ix])), x = j,
        weights = sum(ix), sigma2 = var(log(cghs$earnings[ix]))/sum(ix))
    dd <- rbind(dd, df)
}
```

The column `weights` gives the number of observations that each cell averages over. In this case, if we weight the observations using `weights`, we can recover the original estimates (and the same worst-case bias). If we use the estimates of the conditional variance of the outcome, `dd$sigma2`, then we can also replicate the standard error calculations:

```
s0 <- RDHonest(log(earnings) ~ yearat14, cutoff = 1947,
    data = cghs)
#> Using Armstrong & Kolesar (2020) ROT for smoothness constant M.
## keep same bandwidth
s1 <- RDHonest(y ~ x, cutoff = 1947, data = dd, weights = weights,
    sigmaY2 = sigma2, se.method = "supplied.var", h = s0$coefficients$bandwidth)
#> Using Armstrong & Kolesar (2020) ROT for smoothness constant M.
## Results are identical:
s0
```

```
#>
#> Call:
#> RDHonest(formula = log(earnings) ~ yearat14, data = cghs, cutoff = 1947)
#>
#> Inference for Sharp RD parameter (using Holder class), confidence level 95%:
#>                  Estimate Std. Error Maximum Bias      Confidence Interval
#> I(yearat14>0) 0.07047433 0.05307276   0.03796328 (-0.05531319, 0.1962619)
#>
#> Onesided CIs:  (-Inf, 0.1957345), (-0.05478587, Inf)
#> Number of effective observations: 9074.452
#> Maximal leverage for sharp RD parameter: 0.0005055239
#> Smoothness constant M: 0.02296488
#> P-value: 0.2905956
#>
#> Based on local regression with bandwidth: 3.442187, kernel: triangular
#> Regression coefficients:
#>          I(yearat14>0)  I(yearat14>0):yearat14             (Intercept)
#>               0.070474                0.009301                8.732659
#>               yearat14
#>               0.010848
s1
#>
#> Call:
#> RDHonest(formula = y ~ x, data = dd, weights = weights, cutoff = 1947,
#>     h = s0$coefficients$bandwidth, se.method = "supplied.var",
#>     sigmaY2 = sigma2)
#>
#> Inference for Sharp RD parameter (using Holder class), confidence level 95%:
#>          Estimate Std. Error Maximum Bias      Confidence Interval
#> I(x>0) 0.07047433 0.05307276   0.03796328 (-0.05531319, 0.1962619)
#>
#> Onesided CIs:  (-Inf, 0.1957345), (-0.05478587, Inf)
#> Number of effective observations: 9074.452
#> Maximal leverage for sharp RD parameter: 0.0005055239
#> Smoothness constant M: 0.02296488
#> P-value: 0.2905956
#>
#> Based on local regression with bandwidth: 3.442187, kernel: triangular
#> Regression coefficients:
#>      I(x>0)     I(x>0):x  (Intercept)            x
#>    0.070474     0.009301     8.732659     0.010848
```

Without supplying the variance estimates and specifying `se.method="supplied.var"`, the variance estimates will not match, since the collapsed data is not generally not sufficient to learn about the true variability of the collapsed outcomes.

The same method works in fuzzy designs, but one has to also save the conditional variance of the treatment and its covariance with the outcome:

```
r0 <- RDHonest(log(cn) | retired ~ elig_year, data = rcp,
    h = 7)
#> Using Armstrong & Kolesar (2020) ROT for smoothness constant M.
dd <- data.frame(x = sort(unique(rcp$elig_year)), y = NA,
    d = NA, weights = NA, sig11 = NA, sig12 = NA, sig21 = NA,
    sig22 = NA)
for (j in seq_len(NROW(dd))) {
    ix <- rcp$elig_year == dd$x[j]
    Y <- cbind(log(rcp$cn[ix]), rcp$retired[ix])
    dd[j, -1] <- c(colMeans(Y), sum(ix), as.vector(var(Y))/sum(ix))
}
r1 <- RDHonest(y | d ~ x, data = dd, weights = weights,
    sigmaY2 = sig11, T0 = 0, sigmaYD = sig21, sigmaD2 = sig22,
    h = 7, se.method = "supplied.var")
#> Using Armstrong & Kolesar (2020) ROT for smoothness constant M.
## Outputs match up to numerical precision
max(abs(r0$coefficients[2:11] - r1$coefficients[2:11]))
#> [1] 2.728484e-11
```

4.3 Clustering

In some applications, the data are collected by clustered sampling. In such cases, the user can specify a vector `clusterid` signifying cluster membership. In this case, preliminary bandwidth calculations assume that the regression errors have a Moulton-type structure, with homoskedasticity on either side of the cutoff:

$$\operatorname{cov}(Y_i, Y_j) = \begin{cases} \sigma_+^2 & \text{if } i = j \text{ and } x_i \geq 0, \\ \sigma_-^2 & \text{if } i = j \text{ and } x_i < 0, \\ \rho & \text{if } i \neq j \text{ and } g(i) = g(j), \\ 0 & \text{otherwise,} \end{cases}$$

where $g(i) \in \{1, \ldots, G\}$ denotes cluster membership. Since it appears difficult to generalize the nearest neighbor variance estimator to clustering, we use regression-based cluster-robust variance formulas to compute estimator variances, so that option `se.method="EHW"` is required.

```
## make fake clusters
set.seed(42)
clusterid <- sample(1:50, NROW(lee08), replace = TRUE)
sc <- RDHonest(voteshare ~ margin, data = lee08, se.method = "EHW",
    clusterid = clusterid, M = 0.14, h = 7)
## Since clusters are unrelated to outcomes, not
## clustering should yield similar standard errors
sn <- RDHonest(voteshare ~ margin, data = lee08, se.method = "EHW",
    M = 0.14, h = 7)
sc
#>
#> Call:
#> RDHonest(formula = voteshare ~ margin, data = lee08, M = 0.14,
#>     h = 7, se.method = "EHW", clusterid = clusterid)
```

```
#>
#> Inference for Sharp RD parameter (using Holder class), confidence level 95%:
#>             Estimate Std. Error Maximum Bias  Confidence Interval
#> I(margin>0) 5.821591   1.528091    0.7131699 (2.527825, 9.115358)
#>
#> Onesided CIs:  (-Inf, 9.048247), (2.594936, Inf)
#> Number of effective observations: 695.1204
#> Maximal leverage for sharp RD parameter: 0.01063752
#> Smoothness constant M:     0.14
#> P-value: 0.0004238717
#>
#> Based on local regression with bandwidth:        7, kernel: triangular
#> Regression coefficients:
#>        I(margin>0)  I(margin>0):margin         (Intercept)              margin
#>            5.82159             0.07179            46.38139             0.65536
sn
#>
#> Call:
#> RDHonest(formula = voteshare ~ margin, data = lee08, M = 0.14,
#>     h = 7, se.method = "EHW")
#>
#> Inference for Sharp RD parameter (using Holder class), confidence level 95%:
#>             Estimate Std. Error Maximum Bias  Confidence Interval
#> I(margin>0) 5.821591   1.417699    0.7131699 (2.725463,  8.91772)
#>
#> Onesided CIs:  (-Inf, 8.866669), (2.776514, Inf)
#> Number of effective observations: 695.1204
#> Maximal leverage for sharp RD parameter: 0.01063752
#> Smoothness constant M:     0.14
#> P-value: 0.000159109
#>
#> Based on local regression with bandwidth:        7, kernel: triangular
#> Regression coefficients:
#>        I(margin>0)  I(margin>0):margin         (Intercept)              margin
#>            5.82159             0.07179            46.38139             0.65536
```

4.4 Specification testing

The package also implements lower-bound estimates for the smoothness constant M for the Taylor and Hölder smoothness class, as described in the supplements to Kolesár and Rothe [2018] and Armstrong and Kolesár [2018]

```
r1 <- RDHonest(voteshare ~ margin, data = lee08, M = 0.1,
    se.method = "nn")
### Only use three point-average for averages of a 100
### points closest to cutoff, and report results
### separately for points above and below cutoff
RDSmoothnessBound(r1, s = 100, separate = TRUE, multiple = FALSE,
```

```
    sclass = "T")
#>               estimate  conf.low
#> Below cutoff 0.3337537 0.1224842
#> Above cutoff 0.1738905 0.0000000
## Pool estimates based on observations below and
## above cutoff, and use three-point averages over the
## entire support of the running variable
RDSmoothnessBound(r1, s = 100, separate = FALSE, multiple = TRUE,
    sclass = "H")
#>         estimate   conf.low
#> Pooled 0.1959213 0.01728526
```

4.5 Optimal weights under Taylor smoothness class

For the second-order Taylor smoothness class, the function `RDHonest`, with `kernel="optimal"`, computes finite-sample optimal estimators and confidence intervals, as described in Section 2.2 in Armstrong and Kolesár [2018]. This typically yields tighter CIs:

```
r1 <- RDHonest(voteshare ~ margin, data = lee08, kern = "optimal",
    M = 0.1, opt.criterion = "FLCI", se.method = "nn")$coefficients
r2 <- RDHonest(voteshare ~ margin, data = lee08, kern = "triangular",
    M = 0.1, opt.criterion = "FLCI", se.method = "nn", sclass = "T")$coefficients
r1$conf.high - r1$conf.low
#> [1] 6.33639
r2$conf.high - r2$conf.low
#> [1] 6.685286
```

5 Inference at a point

The package can also perform inference at a point, and optimal bandwidth selection for inference at a point. Suppose, for example, one was interested in the vote share for candidates with margin of victory equal to 20 points:

```
## Specify we're interested in inference at x0=20, and
## drop observations below cutoff
RDHonest(voteshare ~ margin, data = lee08, subset = margin >
    0, cutoff = 20, kern = "uniform", opt.criterion = "MSE",
    sclass = "H", point.inference = TRUE)
#> Using Armstrong & Kolesar (2020) ROT for smoothness constant M.
#>
#> Call:
#> RDHonest(formula = voteshare ~ margin, data = lee08, subset = margin >
#>     0, cutoff = 20, kern = "uniform", opt.criterion = "MSE",
#>     sclass = "H", point.inference = TRUE)
#>
#> Inference for Value of conditional mean (using Holder class), confidence level 95%:
#>             Estimate Std. Error Maximum Bias  Confidence Interval
#> (Intercept) 61.66394   0.468336    0.2482525 (60.63086, 62.69703)
```

```
#>
#> Onesided CIs:  (-Inf, 62.68254), (60.64535, Inf)
#> Number of effective observations:      738
#> Maximal leverage for value of conditional mean: 0.001435988
#> Smoothness constant M: 0.0275703
#> P-value:        0
#>
#> Based on local regression with bandwidth: 7.311286, kernel: uniform
#> Regression coefficients:
#> (Intercept)       margin
#>     61.6639       0.4076
```

To compute the optimal bandwidth, the package assumes homoskedastic variance on either side of the cutoff, which it estimates based on a preliminary local linear regression using the Fan and Gijbels [1996] rule of thumb bandwidth selector. This homoskedasticity assumption is dropped when the final standard errors are computed.

6 Diagnostics: leverage and effective observations

The estimators in this package are just weighted regression estimators, or ratios of regression estimators in the fuzzy RD case. Regression estimators are linear in outcomes, taking the form $\sum_i k_{i,h} Y_i$, where $k_{i,h}$ are estimation weights, returned by `data$est_w` part of the `RDHonest` output (see expression for $\hat{\tau}_{Y,h}$ above).

For the sampling distribution of the estimator to be well-approximated by a normal distribution, it is important that these regression weights not be too large: asymptotic normality requires $L_{\max} = \max_j k_{j,h}^2 / \sum_i k_{i,h}^2 \to 0$. If uniform kernel is used, the weights $k_{i,h}$ are just the diagonal elements of the partial projection matrix. We therefore refer to $L_{\max}$ as maximal (partial) leverage, and it is reported in the `RDHonest` output. The package issues a warning if the maximal leverage exceeds 0.1—in such cases using a bigger bandwidth is advised.

In the fuzzy RD case, by Theorem B.2 in the appendix to Armstrong and Kolesár [2020], the estimator is asymptotically equivalent to $\sum_i k_{i,h}(Y_i - D_i\theta)/\tau_D$, where $k_{i,h}$ are the weights for $\hat{\tau}_{Y,h}$. The maximal leverage calculations are thus analogous to the sharp case.

With local regression methods, it is clear that observations outside the estimation window don't contribute to estimation, reducing the effective sample size. If the uniform kernel is used, the package therefore reports the number of observations inside the estimation window as the "number of effective observations".

To make this number comparable across different kernels, observe that, under homoskedasticity, the variance of a linear estimator $\sum_i k_i Y_i$ is $\sigma^2 \sum_i k_i^2$. We expect this to scale in inverse proportion to the sample size: with twice as many observations and the same bandwidth, we expect the variance to halve. Therefore, if the variance ratio relative to a uniform kernel estimator with weights $\sum_i k_{uniform,i} Y_i$ is $\sigma^2 \sum_i k_{uniform,i}^2 / \sigma^2 \sum_i k_i^2 = \sum_i k_{uniform,i}^2 / \sum_i k_i^2$, the precision of this estimator is the same as if we used a uniform kernel, but with $\sum_i k_{uniform,i}^2 / \sum_i k_i^2$ as many observations. Correspondingly, we define the number of effective observations for other kernels as the number of observations inside the estimation window times $\sum_i k_{uniform,i}^2 / \sum_i k_i^2$. With this definition, using

a triangular kernel typically yields effective samples sizes equal to about 80% of the number of observations inside the estimation window.

Finally, to assess which observations are important for pinning down the estimate, it can be useful to explicitly plot the estimation weights.

References

Timothy B. Armstrong and Michal Kolesár. Optimal inference in a class of regression models. *Econometrica*, 86(2):655–683, March 2018. doi: 10.3982/ECTA14434.

Timothy B. Armstrong and Michal Kolesár. Simple and honest confidence intervals in nonparametric regression. *Quantitative Economics*, 11(1):1–39, January 2020. doi: 10.3982/QE1199.

Erich Battistin, Agar Brugiavini, Enrico Rettore, and Guglielmo Weber. The retirement consumption puzzle: Evidence from a regression discontinuity approach. *American Economic Review*, 99(5): 2209–2226, December 2009. doi: 10.1257/aer.99.5.2209.

Sebastian Calonico, Matias D. Cattaneo, Max H. Farrell, and Rocío Titiunik. Regression discontinuity designs using covariates. *The Review of Economics and Statistics*, 101(3):442–451, July 2019. doi: 10.1162/rest_a_00760.

Jianqing Fan and Irène Gijbels. *Local Polynomial Modelling and Its Applications*. Number 66 in Monographs on Statistics and Applied Probability. Chapman & Hall/CRC, New York, NY, 1996. doi: 10.1201/9780203748725.

Jinyong Hahn, Petra E. Todd, and Wilbert van der Klaauw. Identification and estimation of treatment effects with a regression-discontinuity design. *Econometrica*, 69(1):201–209, January 2001. doi: 10.1111/1468-0262.00183.

Guido W. Imbens and Karthik Kalyanaraman. Optimal bandwidth choice for the regression discontinuity estimator. *The Review of Economic Studies*, 79(3):933–959, July 2012. doi: 10.1093/restud/rdr043.

Guido W. Imbens and Stefan Wager. Optimized regression discontinuity designs. 101(2):264–278, May 2019. doi: 10.1162/rest_a_00793.

Michal Kolesár and Christoph Rothe. Inference in regression discontinuity designs with a discrete running variable. *American Economic Review*, 108(8):2277—-2304, August 2018. doi: 10.1257/aer.20160945.

Rafael Lalive. How do extended benefits affect unemployment duration? A regression discontinuity approach. *Journal of Econometrics*, 142(2):785–806, February 2008. doi: 10.1016/j.jeconom.2007.05.013.

David S. Lee. Randomized experiments from non-random selection in U.S. House elections. *Journal of Econometrics*, 142(2):675–697, February 2008. doi: 10.1016/j.jeconom.2007.05.004.

Mark G. Low. On nonparametric confidence intervals. 25(6):2547–2554, December 1997. doi: 10.1214/aos/1030741084.

Jens Ludwig and Douglas L. Miller. Does head start improve children's life chances? Evidence from a regression discontinuity design. *Quarterly Journal of Economics*, 122(1):159–208, February 2007. doi: 10.1162/qjec.122.1.159.

Claudia Noack and Christoph Rothe. Bias-aware inference in fuzzy regression discontinuity designs, February 2021.

Philip Oreopoulos. Estimating average and local average treatment effects when compulsory education schooling laws really matter. *American Economic Review*, 96(1):152–175, March 2006. doi: 10.1257/000282806776157641.