Fuzzy Regression Discontinuity Design

A Simulation-Based Examination of Estimator Performance

Mark Fulginiti

2025-12-05

Contents

Introduction

Regression discontinuity designs (RDDs) provide a powerful framework for causal inference in settings where treatment assignment is governed by a deterministic threshold rule. As emphasized in Cunningham (2021), RDDs are "considered one of the most credible research designs with observational data," because units arbitrarily close to the cutoff are comparable in expectation. When a policy, score, or administrative criterion assigns treatment based on whether a running variable crosses a cutoff, observations near that threshold allow causal effects to be recovered using local comparisons around the eligibility boundary. In the canonical sharp RDD, treatment assignment changes deterministically at the cutoff. In many real applications, however, assignment is not strictly deterministic: eligibility shifts discretely at the cutoff, but the probability of receiving treatment does not move from zero to one. These "fuzzy" designs arise routinely in education, social programs, lending, medical decision-making, and administrative processes where implementation is imperfect or compliance is incomplete.

In fuzzy RDDs, crossing the cutoff functions as a local instrumental variable (IV), generating exogenous variation in treatment near the threshold. The resulting estimand is a local average treatment effect for compliers at the cutoff, typically recovered using a local Wald ratio or an equivalent local IV formulation.

Although the theoretical properties of fuzzy RDD estimators are well understood, their empirical performance depends critically on practical features of the design. These include the strength of the first stage, curvature in the conditional expectation functions, bandwidth choice, and the use of auxiliary covariates. Recent work further highlights how high-dimensional covariates and machine-learning methods can improve precision in data-rich settings while preserving the causal estimand (Chernozhukov et al., 2025).

This paper presents a focused simulation study evaluating commonly used fuzzy RDD estimators under a range of empirically relevant conditions. Using controlled data-generating processes that vary compliance strength, functional-form curvature, confounding, and treatment-effect heterogeneity, the study examines the behavior of the Wald estimator, local two-stage least squares, and modern bias-corrected methods. The goal is to clarify when fuzzy RDD performs reliably, where it becomes fragile, and how design and implementation choices affect estimator accuracy and inference. The simulation framework is implemented primarily in Python, with bias-corrected inference using rdrobust conducted in R and results formatted for presentation.

Fuzzy RDD Framework

Setup

Let X denote the running variable and c the eligibility cutoff. Let $D \in \{0, 1\}$ represent treatment status and Y the observed outcome. In a fuzzy regression discontinuity design, crossing the cutoff does not deterministically assign treatment. Instead, treatment probability shifts discontinuously at the threshold:

$$P(D = 1 \mid X = x) \text{ has a discontinuity at } x = c, \quad \text{but } D \neq \mathbf{1}\{X \geq c\}.$$

Eligibility therefore serves as an instrumental variable, defined by

$$Z = \mathbf{1}(X \geq c).$$

Identification in fuzzy RDD is local and limit-based. Because units with $X < c$ and $X > c$ do not overlap in the running variable, identification does not rely on global overlap. Instead, causal effects are recovered from comparisons arbitrarily close to the cutoff, where left- and right-hand limits of conditional expectations coincide as $x \to c$.

Target Estimand

The causal effect in the fuzzy design is captured by the local Wald estimand:

$$\tau_{\text{FRD}} = \frac{\lim_{x \downarrow c} E[Y \mid X = x] - \lim_{x \uparrow c} E[Y \mid X = x]}{\lim_{x \downarrow c} E[D \mid X = x] - \lim_{x \uparrow c} E[D \mid X = x]}.$$

This estimand compares the discontinuity in the outcome to the discontinuity in treatment probability at the cutoff. Under the maintained continuity and no-manipulation conditions, τ_{FRD} identifies a local average treatment effect (LATE) for the subpopulation of compliers, that is, units whose treatment status is shifted by crossing the threshold.

In the fuzzy design, the reduced-form discontinuity in Y reflects the causal effect scaled by the fraction of compliers. Dividing by the first-stage discontinuity rescales this reduced-form effect and recovers the causal effect for compliers at the cutoff. When the first-stage jump is small, as in weak compliance settings, the Wald ratio mechanically amplifies sampling variability, increasing estimator variance.

Although the Wald estimand is defined as a limit, empirical implementation proceeds by estimating the left- and right-hand limits of the conditional expectations of Y and D and taking their ratio to form $\widehat{\tau}_{\text{FRD}}$.

Local 2SLS (Local IV) Formulation

Fuzzy RDD can be implemented as a localized instrumental variables design. In practice, researchers estimate the reduced form and first stage using local polynomial regressions with separate functional forms on either side of the cutoff. A standard fully interacted specification is:

First stage: effect of instrument Z on treatment D

$$D_i = \alpha_0 + \alpha_1 Z_i + f_-(X_i - c) \cdot (1 - Z_i) + f_+(X_i - c) \cdot Z_i + u_i,$$

Reduced form: effect of instrument Z on outcome Y

$$Y_i = \beta_0 + \beta_1 Z_i + g_-(X_i - c) \cdot (1 - Z_i) + g_+(X_i - c) \cdot Z_i + \varepsilon_i.$$

The polynomial functions f_-, f_+, g_-, and g_+, each of degree p, absorb smooth variation in the running variable on each side of the cutoff, leaving $\widehat{\alpha}_1$ and $\widehat{\beta}_1$ to capture only the discontinuities in treatment and outcome, respectively. The treatment effect is then recovered via local 2SLS:

$$\tau = \frac{\widehat{\beta}_1}{\widehat{\alpha}_1},$$

which coincides with the Wald estimand. This equivalence highlights that fuzzy RDD is fundamentally a local IV design, with the cutoff indicator providing the exogenous variation required for causal identification, that is, variation in treatment assignment that is independent of individuals' potential outcomes and arises solely from the institutional rule at the threshold.

Identification Assumptions

Fuzzy regression discontinuity designs rely on a set of local identification assumptions under which discontinuities in treatment and outcome at the cutoff can be attributed to the causal effect of treatment.

These assumptions parallel those used in sharp regression discontinuity designs, local instrumental variables frameworks, and the LATE interpretation of IV estimands, as emphasized by Cunningham (2021). The distinction is that, in fuzzy RDD, these conditions are required only in a neighborhood around the cutoff rather than over the full support of the running variable.

Continuity of Potential Outcomes

The central assumption is continuity of expected potential outcomes at the threshold:

$$E[Y(t) \mid X = x] \text{ is continuous at } x = c.$$

Continuity requires that, holding treatment status fixed, expected outcomes evolve smoothly in the running variable at the cutoff. This rules out any omitted factor, institutional change, or competing intervention that would itself induce a discrete jump in outcomes at exactly the same point. If such a jump were present, it would be observationally indistinguishable from a treatment effect and identification would fail.

Continuity is the regression discontinuity analogue of conditional exogeneity, but it is strictly weaker. Conditional exogeneity requires treatment assignment to be independent of potential outcomes conditional on observed covariates. In contrast, RDD allows treatment to depend arbitrarily on observed and unobserved characteristics away from the cutoff, provided that those characteristics do not change discontinuously at c. As Cunningham notes, continuity should be viewed as the natural baseline: in the absence of a treatment-induced break, conditional expectations should pass smoothly through the threshold because "nature does not make jumps." Violations of this assumption arise when agents, institutions, or policies change discretely at the cutoff in ways unrelated to treatment, producing spurious outcome discontinuities.

First Stage Discontinuity

Identification further requires a discontinuity in the probability of treatment at the cutoff:

$$\lim_{x \downarrow c} P(D = 1 \mid X = x) \quad \neq \quad \lim_{x \uparrow c} P(D = 1 \mid X = x).$$

This condition ensures that crossing the threshold generates exogenous variation in treatment. If treatment probability were continuous at the cutoff, the local Wald estimand would be undefined because the denominator would vanish. The magnitude of the discontinuity determines the relevance of the instrument and directly parallels the relevance condition in instrumental variables models. As in IV settings, small first-stage discontinuities correspond to weak instruments, implying limited identifying variation even when other assumptions hold.

Local Exclusion Restriction

In the fuzzy design, the cutoff indicator affects outcomes only through its effect on treatment for units arbitrarily close to the threshold. Formally, conditional on treatment status, the cutoff indicator has no direct effect on potential outcomes in a neighborhood of c. This exclusion restriction is local rather than global: it need only hold in the limit as $x \to c$.

In regression discontinuity designs, the exclusion restriction follows from continuity of potential outcomes combined with the presence of a first-stage discontinuity. If potential outcomes are smooth at the cutoff and the only feature of the data-generating process that changes discretely at c is treatment probability, then any observed jump in outcomes must be attributable to the causal effect of treatment among compliers. This mirrors the logic of instrumental variables, with the important distinction that exclusion is justified locally by design rather than imposed globally.

No Precise Manipulation of the Running Variable

Finally, units must not precisely manipulate the running variable to sort just above or below the cutoff. Precise manipulation would invalidate local comparability and undermine the continuity assumption by inducing systematic differences between units on either side of the threshold.

This assumption is closely related to continuity of the running-variable density at the cutoff. Strategic sorting, heaping, or bunching at c may indicate that agents can finely control their position relative to the threshold, compromising identification. Institutional context is therefore critical: many running variables are administratively assigned, measured with noise, or determined by processes that make precise manipulation implausible. When fine control over the running variable is infeasible, the no-manipulation assumption is credible.

Practical Issues and Extensions

Covariate Adjustment

Covariates are often incorporated into fuzzy RDD implementations to improve precision, even though they are not required for identification. In applied work, covariate adjustment is typically used to reduce residual variance within the local neighborhood around the cutoff, thereby tightening standard errors without altering the target estimand.

Several approaches are commonly used. Low-dimensional linear adjustments augment local regressions with a small set of predetermined covariates that explain outcome variation near the cutoff. High-dimensional approaches, such as post-Lasso or related selection procedures, draw from a large pool of candidate predictors and include the selected variables in subsequent local regressions, trading variance reduction against additional tuning choices. More recent machine-learning approaches often rely on residualization, using flexible prediction methods to partial out covariate effects from the outcome and treatment prior to estimating the local effect. In all cases, covariate adjustment is best viewed as a precision-enhancing device rather than a substitute for design-based identification.

Heterogeneous Treatment Effects

Empirical applications often seek to assess whether treatment effects vary across subpopulations. In fuzzy regression discontinuity designs, heterogeneity is typically examined either by estimating subgroup-specific designs or by allowing treatment effects to vary with observable subgroup indicators in a pooled specification.

Subgroup-specific analyses estimate the fuzzy RD effect separately within each stratum, targeting complier-specific local average treatment effects for each group. Alternatively, interacted specifications incorporate subgroup indicators directly into the local regression framework, allowing the reduced form and first stage to differ across groups while maintaining a common estimation window.

Both approaches extend the standard fuzzy RD framework in a straightforward manner and preserve the local interpretation of the estimand. The choice between subgroup-specific estimation and interacted specifications is primarily driven by the structure of the heterogeneity of interest and by empirical design considerations rather than by differences in identification logic.

Bandwidth Selection and Bias Correction

Bandwidth choice is a central practical decision in fuzzy RDD applications. Narrow bandwidths restrict attention to observations very close to the cutoff, reducing sensitivity to functional-form misspecification but increasing sampling variability. Wider bandwidths increase effective sample size and stabilize estimation, at the cost of potentially incorporating curvature that local polynomial approximations do not fully capture.

Modern practice relies on data-driven bandwidth selectors combined with analytical or robust bias correction. The Calonico–Cattaneo–Titiunik framework emphasizes that conventional standard errors can be poorly calibrated in local polynomial settings. Bias-corrected point estimates paired with robust variance estimators provide more reliable inference and mitigate the undercoverage associated with classical local linear methods.

Violations and Diagnostic Checks

Applied fuzzy RDD analyses routinely assess design credibility using diagnostic and falsification tools designed to probe the plausibility of the identifying assumptions. Because identification relies on local continuity and the absence of manipulation at the cutoff, diagnostics focus on detecting violations that would undermine local comparability rather than on model fit or predictive accuracy.

Density-based diagnostics examine whether the distribution of the running variable is smooth at the cutoff. A discontinuity in the density may indicate sorting or precise manipulation, which would invalidate the assumption that units just above and below the threshold are comparable. In practice, this is commonly assessed using tests such as McCrary's density test, which evaluates whether excess mass appears on either side of the cutoff relative to local trends in the running-variable distribution.

Covariate balance checks assess whether predetermined characteristics vary smoothly through the threshold. While covariate balance is not required for identification in RDD, large or systematic discontinuities in baseline covariates can signal violations of the continuity assumption or institutional changes coinciding with the cutoff. Balance checks therefore serve as indirect evidence on whether unobserved determinants of the outcome are also likely to be smooth at the threshold.

Placebo exercises provide an additional falsification tool by testing whether the estimation procedure detects discontinuities where none should exist. In fuzzy RDDs, placebo tests must be interpreted carefully. Shifting the cutoff generally eliminates the first-stage discontinuity, rendering the fuzzy Wald estimand undefined. As a result, placebo tests are typically implemented using sharp RDD specifications applied to the outcome at false cutoffs or to outcomes that should not respond to treatment. Evidence of systematic placebo effects would suggest that curvature, misspecification, or numerical artifacts are driving apparent discontinuities rather than the treatment mechanism.

Taken together, these diagnostics do not prove identification, but they help rule out prominent failure modes. When density is smooth, covariates are balanced, and placebo tests show no spurious jumps, the empirical implementation is consistent with the maintained assumptions, strengthening the interpretation that observed discontinuities at the true cutoff reflect the causal effect of treatment rather than unrelated features of the data.

Estimation Methods Compared

The simulation study compares a small set of estimators that span naïve, misspecified, classical, and modern approaches to fuzzy regression discontinuity. This section introduces the estimators included in the comparison; formal definitions and implementation details are provided in the Appendix.

We consider five estimators. (1) Naïve OLS, which ignores the cutoff and treats the running variable as a standard covariate, serves as a baseline for assessing the gains from design-aware methods. (2) Sharp RDD applied to a fuzzy design, which incorrectly assumes deterministic treatment at the cutoff and is included as a representative misspecification commonly encountered in practice. (3) The fuzzy Wald estimator, defined as the ratio of outcome and treatment discontinuities, provides the nonparametric benchmark that directly targets the fuzzy RD estimand. (4) Local two-stage least squares (2SLS) with fully interacted polynomial terms implements the fuzzy design within a regression-based local IV framework. (5) Bias-corrected local polynomial FRD via rdrobust, which combines data-driven bandwidth selection, bias correction, and robust variance estimation, serves as a best-practice inference benchmark.

Together, these estimators allow the simulations to isolate how design alignment, compliance strength, and smoothing choices affect finite-sample performance in fuzzy regression discontinuity settings.

Simulation Design

The simulation study uses Monte Carlo experiments to evaluate the fuzzy regression discontinuity estimators described above under a sequence of data-generating processes that vary compliance strength, curvature of the conditional expectation functions, smooth confounding, and treatment-effect heterogeneity. Each scenario isolates a distinct empirical challenge commonly encountered in applied fuzzy RDD settings. In each Monte Carlo replication, data are generated from the same core structure: a smooth running variable, a discontinuous treatment probability at the cutoff, and potential outcomes that evolve smoothly in the running variable except for the treatment-induced jump.

Table 1: Summary of simulation scenarios used in the Monte Carlo study.

Scenario	First-stage strength	Outcome function	Smooth confounding	Treatment-effect heterogeneity	Noise
S1: Baseline	Moderate	Mild curvature	No	No	Homoskedastic
S2: Curvature	Moderate	Strong curvature	No	No	Homoskedastic
S3: Weak Compliance	Weak	Mild curvature	No	No	Homoskedastic
S4: Confounding	Moderate	Mild curvature	Yes (smooth, correlated with X)	No	Homoskedastic
S5: Heterogeneity	Moderate	Mild curvature	No	Yes (binary subgroup)	Homoskedastic

Running Variable

The running variable is generated from a smooth distribution with support around the threshold. For all designs, we take $X \sim \text{Uniform}(-1, 1)$, with a fixed cutoff $c = 0$. The uniform distribution ensures that density is flat and continuous at the threshold, eliminating manipulation by construction. Identification therefore depends only on the curvature of the outcome surface and the strength of the discontinuity in treatment probability.

Compliance Mechanism

Treatment assignment is probabilistic and incorporates a discontinuity at the cutoff. For each unit,

$$P(D = 1 \mid X = x) = g(x) + \delta \mathbf{1}(x \geq 0),$$

where $g(x)$ is a smooth baseline compliance function and δ governs instrument strength. Varying δ generates weak, moderate, and strong first-stage settings. Because all components of $g(x)$ are smooth, the only source of discontinuity in the treatment probability is the cutoff indicator. This structure ensures that the Wald estimand and local 2SLS estimators are well-defined and isolates how compliance strength influences estimator stability.

Outcome Model

The outcome evolves according to

$$Y = \tau D + m(X) + \varepsilon,$$

where τ is the true treatment effect at the cutoff and $m(\cdot)$ captures smooth evolution in the running variable. Three variants of $m(x)$ are used. The first imposes mild curvature, approximating situations where linear local polynomials are adequate. The second uses sharper curvature to induce bias in misspecified local regressions and to test bandwidth sensitivity. The third introduces a smooth unobserved confounder,

$$U = a_0 + a_1 X + a_2 X^2 + \eta,$$

that enters the outcome additively and creates correlation between X and the error term. This variant evaluates how naïve and misspecified estimators deteriorate when curvature and confounding interact. In all designs, ε and η are independent noise terms with mean zero.

Treatment Effect Heterogeneity

To examine robustness when treatment effects vary across units, the design incorporates a binary subgroup indicator G and defines $\tau = \tau(G)$, so that compliers in different strata have different causal effects. This structure allows the simulation to evaluate whether local 2SLS and Wald estimators recover the correct complier-specific effects, particularly under moderate or weak compliance, where estimator variance increases and subgroup inference becomes difficult.

Covariates

The data generating process includes the running variable X, the treatment indicator D, the cutoff indicator $Z = \mathbf{1}\{X \geq 0\}$, and, in some scenarios, an unobserved confounder U and a binary subgroup indicator G. No additional observed covariates are included beyond functions of the running variable.

The outcome surface is specified as a smooth function of X,

$$m(X) = \beta_0 + \beta_1 X + \beta_2 X^2,$$

ensuring continuity at the cutoff in all baseline scenarios. Identification in the fuzzy regression discontinuity design therefore relies exclusively on the discontinuity in treatment probability at the cutoff, induced by the jump parameter δ in the compliance equation.

While covariate adjustment and high-dimensional controls can improve precision in fuzzy RD settings, these are intentionally omitted from the present simulation. This allows the comparison of estimators to focus on core issues of bandwidth choice, weak first stages, curvature, confounding, and treatment effect heterogeneity without conflating these effects with variable-selection or regularization behavior. As a result, all estimators target the same local average treatment effect at the cutoff, and differences in performance arise from estimator design rather than auxiliary covariate structure.

Performance Metrics

Estimator performance is assessed across several dimensions. Bias, root mean squared error, and confidence-interval coverage evaluate accuracy and inferential reliability. First-stage strength is characterized through the size of the treatment-probability discontinuity, mean first-stage coefficients, and the associated F-statistic, allowing weak-IV behavior to be examined directly. Sensitivity to bandwidth choice is assessed by comparing performance under the MSE-optimal bandwidth and bandwidths scaled to one-half and twice that value, following rdrobust conventions. Finally, placebo cutoffs are used to evaluate design validity: the mean estimated effect, false-positive rate, bias, RMSE, and coverage at placebo thresholds indicate whether estimators spuriously detect discontinuities when none exist.

Implementation Details

Local estimators are implemented using symmetric bandwidths around the cutoff. The local Wald estimator is computed using simple local mean differences in outcomes and treatment take-up on either side of the cutoff. Local two-stage least squares (local 2SLS) is implemented using a local linear specification ($p = 1$), with the running variable centered at the cutoff and side-specific slopes achieved via inclusion of the interaction term $Z \cdot (X - c)$. Uniform weighting is used within each bandwidth.

Standard errors for naïve OLS, incorrect sharp RDD, and local 2SLS are computed using conventional OLS variance formulas. For local 2SLS, estimation is implemented via a two-step fitted-regressor procedure, and reported standard errors are taken from the second-stage OLS fit on $\widehat{D}$. Bias-corrected fuzzy regression discontinuity estimates and robust standard errors are obtained using the rdrobust package following the Calonico–Cattaneo–Titiunik procedure.

First-stage strength is summarized using the estimated local discontinuity in treatment probability for the Wald estimator and the corresponding first-stage F-statistic for local 2SLS, computed as the squared t-statistic on Z in the local first stage, both evaluated within the estimation bandwidth.

Software

All simulations and estimator implementations were conducted using Python and R. The data-generating process, Monte Carlo simulation framework, and baseline estimators (naïve OLS, sharp RDD, fuzzy Wald, and local 2SLS) were implemented in Python, primarily using NumPy, pandas, and statsmodels. Bias-corrected fuzzy regression discontinuity estimation and robust inference were performed in R using the rdrobust package. All tables and figures were formatted using RMarkdown and LaTeX.

Results

Baseline Performance

Table 2 reports baseline performance under a smooth design with a moderately strong first stage at $n = 10{,}000$, using a common bandwidth for all local estimators.

Three contrasts are immediate. First, naïve OLS achieves the smallest RMSE, driven by extremely low dispersion, but its estimate reflects a global quantity rather than the local RD estimand. Second, the sharp RDD specification exhibits large negative bias, confirming the consequences of misspecification under incomplete compliance. Third, the design-aligned estimators are centered near the target effect but differ substantially in precision: the fuzzy Wald estimator shows moderate dispersion, while local 2SLS exhibits substantially higher variance despite negligible bias.

The favorable RMSE performance of naïve OLS reflects an estimand mismatch rather than superior causal recovery. By pooling observations across the full support of the running variable, the naïve regression exploits strong global signal and achieves substantial variance reduction. However, this gain comes at the cost of abandoning the local identification strategy of the regression discontinuity design. The resulting estimate targets a global average relationship rather than the local complier average treatment effect at the cutoff. Consequently, variance-based metrics such as RMSE are not comparable across estimators that target fundamentally different estimands, and low RMSE for naïve OLS should not be interpreted as evidence of causal validity within the RD framework.

Overall, the table highlights the baseline bias–variance tradeoff across estimators and provides a reference point for interpreting changes in performance under curvature, weak compliance, confounding, and heterogeneity in subsequent scenarios.

Table 2: Baseline performance (S1: smooth design, moderate first stage), $n = 10{,}000$. Local estimators use bandwidth $h = 0.10$.

Estimator	MC mean($\hat{\tau}$)	Bias	RMSE	SD($\hat{\tau}$)
Naive OLS	0.999	-0.001	0.025	0.025
Sharp RDD (misspecified)	0.323	-0.677	0.681	0.068
Fuzzy Wald (local means)	1.178	0.178	0.286	0.225
Local 2SLS	1.004	0.004	0.593	0.594

Overall Estimator Performance Across Scenarios

This section summarizes pooled estimator performance across all simulation scenarios, emphasizing broad bias–variance patterns rather than tuning-specific behavior.

Several regularities are evident in Table 6 (Appendix D). Across all scenarios, estimators aligned with the fuzzy RD design are centered near the target effect, while the sharp RDD specification exhibits substantial and persistent bias. Naïve OLS continues to achieve the lowest RMSE, reflecting low dispersion rather than alignment with the RD estimand. This pattern is stable across curvature, confounding, and heterogeneity, indicating that apparent RMSE advantages do not translate into causal validity within the RD framework.

Differences among design-aware estimators are primarily driven by precision. The fuzzy Wald estimator exhibits moderate upward bias in several scenarios and consistently higher dispersion than global methods. Local 2SLS remains closely centered on the target across baseline, curvature, confounding, and heterogeneity designs, but with substantially higher variance, particularly relative to the Wald estimator.

Weak compliance produces the most pronounced deterioration in performance. In Scenario S3, both bias and dispersion increase sharply for IV-based estimators. The Wald estimator shows sizeable upward bias, while local 2SLS remains closer to the target on average but with variance inflation. These patterns underscore that weak identification degrades performance through multiple channels and that bias and precision must be evaluated jointly.

Table 7 (Appendix D) further clarifies the role of sample size. Increasing n from $2,000$ to $10,000$ substantially reduces both bias and dispersion for local 2SLS in all scenarios except the weak first-stage case, where improvements are present but limited. Coverage remains near nominal levels across scenarios and sample sizes, indicating that inference adapts appropriately to increased uncertainty even when point estimates are highly variable.

Overall, the results show that respecting the RD design trades precision for robustness. Estimators that ignore the discontinuity can appear favorable under variance-based metrics, but this performance is not informative about causal validity. These patterns motivate the focused examination of bandwidth sensitivity and inference behavior in the sections that follow.

Bandwidth Sensitivity

This section examines how bandwidth choice affects the performance of the local 2SLS estimator. Results are reported in Table 8 (Appendix D) for $h \in \{0.05, 0.10, 0.20\}$, holding the data-generating process and sample size fixed within each scenario.

Across scenarios, bandwidth choice operates primarily through first-stage strength. As the bandwidth widens, the effective local sample increases, leading to a monotonic rise in the first-stage F-statistic. This mechanically improves precision: both $\mathrm{SD}(\hat{\tau})$ and RMSE decline with h, while estimated standard errors contract accordingly.

In the baseline, curvature, confounding, and heterogeneity scenarios, local 2SLS remains approximately unbiased across all bandwidths. Bandwidth changes affect dispersion rather than the center of the sampling distribution, with wider bandwidths yielding tighter sampling distributions and lower RMSE as first-stage strength increases.

The weak first-stage scenario exhibits the same mechanism in an extreme regime. At $h = 0.05$, first-stage strength is very low and local 2SLS displays severe finite-sample distortions, with both bias and dispersion becoming large. As bandwidth increases, first-stage strength improves and estimator behavior stabilizes: bias contracts to modest levels and RMSE falls sharply. Outside this extreme weak-ID region, remaining bandwidth sensitivity is driven primarily by variance rather than systematic bias.

Empirical coverage remains close to nominal across bandwidths in all scenarios, indicating that inference adapts appropriately to changes in sampling variability and first-stage strength. Overall, the results show that bandwidth choice in fuzzy RDD affects local 2SLS mainly through its impact on instrument strength. When the first stage is sufficiently strong, performance is stable across bandwidths; when compliance is weak, excessively narrow bandwidths can induce substantial instability, motivating data-driven bandwidth selection and bias-corrected inference in applied work.

Weak First-Stage Behavior

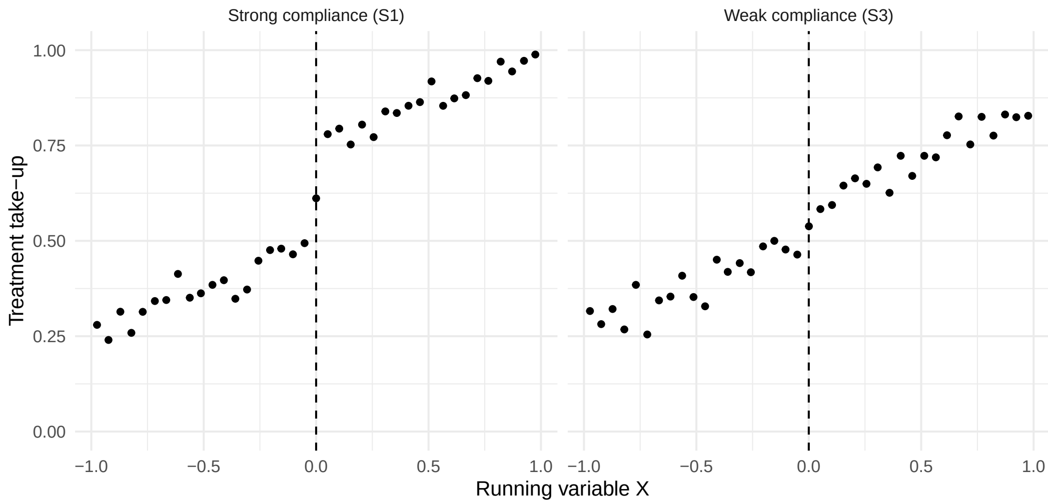
Table 9 (Appendix D) illustrates that weak compliance is the primary source of finite-sample instability in fuzzy regression discontinuity designs, with markedly different behavior across local 2SLS and fuzzy Wald estimators. Results are shown for the weak-compliance scenario at $n = 10{,}000$, where the contrast between local 2SLS and the fuzzy Wald estimator is pronounced. For local 2SLS, very weak first stages at narrow bandwidths produce extreme dispersion, with RMSE exceeding 100 at $h = 0.05$ and remaining orders of magnitude larger than in other scenarios even as the bandwidth widens. This behavior reflects the near-zero denominator problem inherent in fuzzy RD under weak compliance: when the discontinuity in treatment probability is small, sampling noise in the estimated first stage can push the denominator of the Wald or IV ratio arbitrarily close to zero, mechanically producing extreme estimates even when the reduced-form discontinuity remains well behaved. Although increasing the bandwidth strengthens the local first stage and substantially reduces variability, precision remains poor relative to stronger-compliance designs.

In contrast, the fuzzy Wald estimator is markedly more numerically stable. While its RMSE increases under weak compliance, the magnitude of the increase is modest and does not approach the instability observed for local 2SLS. This difference arises because the Wald estimator relies on local mean differences and avoids propagation of first-stage estimation error through fitted regressions, whereas local 2SLS compounds weak-instrument noise through the interacted first-stage and second-stage fits.

Despite severe point-estimate instability for local 2SLS at narrow bandwidths, empirical coverage remains close to nominal, indicating that inference appropriately adapts through widened standard errors. Taken together, these results demonstrate that weak compliance primarily manifests through variance inflation rather than systematic bias, that the severity of this inflation depends strongly on estimator structure, and that robust inference procedures can remain reliable even when point estimation becomes highly unstable.

To distinguish typical estimator behavior from rare but extreme failures, we report robust summaries for the weak-compliance scenario in Table 10 (Appendix D). Median and interquartile-range estimates show that the local 2SLS estimator exhibits a modest downward bias but remains stably centered near the target effect across all bandwidths, including the narrowest window. Trimmed-mean estimates are also downward biased but to a lesser extent, indicating that part of the bias in conventional averages is driven by extreme realizations associated with near-zero first-stage discontinuities. The extreme RMSE values observed under weak compliance are therefore driven by a small number of replications in which the estimated first-stage discontinuity is close to zero, producing mechanically inflated ratio estimates. These results indicate that weak compliance induces instability primarily through tail behavior rather than large systematic bias in the center of the sampling distribution.

The following figures provide a direct visual illustration of the first-stage mechanism underlying the weak-compliance results. In the strong-compliance design, treatment take-up exhibits a clear and economically meaningful discontinuity at the cutoff, ensuring a well-behaved denominator for fuzzy RD estimators. In contrast, the weak-compliance design shows only a very small change in treatment probability at the cutoff relative to sampling variation. As a result, the estimated first-stage discontinuity is frequently close to zero, making ratio-based estimators highly sensitive to noise. This visual contrast clarifies why weak compliance leads to extreme dispersion and occasional blow-ups in local 2SLS estimates, even when the reduced-form relationship remains stable.



Treatment-Effect Heterogeneity

This section examines estimator behavior under treatment-effect heterogeneity using Scenario S5 with $n = 10{,}000$. Results for pooled and subgroup-specific estimates are reported in Table 11 (Appendix D).

At the pooled level, local 2SLS recovers a stable estimate of the complier-weighted average treatment effect across bandwidths, with small bias and declining RMSE as bandwidth increases. This confirms that the fuzzy RD design continues to identify a well-defined local average treatment effect in the presence of heterogeneity.

Subgroup-specific results show clear separation between groups $G0$ and $G1$. Estimated effects are centered around distinct magnitudes consistent with the data-generating process, indicating that heterogeneity is transmitted cleanly through the discontinuity and that local 2SLS targets the appropriate subgroup-specific LATE when estimation is restricted to each group.

Relative to pooled estimates, subgroup analyses exhibit higher dispersion and RMSE across bandwidths. This pattern reflects reduced effective support within each subgroup and is most pronounced at narrower bandwidths. Despite this loss of precision, coverage remains close to nominal across groups and bandwidths.

Overall, the heterogeneity results illustrate the tradeoff between precision and resolution in fuzzy regression discontinuity designs. Pooling yields more precise estimates of the average local effect, while subgroup analyses recover distinct complier-specific effects at the cost of increased sampling variability.

rdrobust as an Inference Benchmark

This section evaluates the bias-corrected fuzzy regression discontinuity estimator implemented in rdrobust as a benchmark for inference. Results are summarized in Table 3 across all five simulation scenarios using MSE-optimal bandwidth selection, robust standard errors, and bias-corrected point estimates.

Across designs, rdrobust delivers stable bias-corrected estimates that remain well centered on the target local average treatment effect. While point-estimate dispersion increases in more challenging settings, particularly under weak first-stage conditions, this behavior reflects genuine identification difficulty rather than estimator failure. Importantly, the estimator adapts its smoothing through data-driven bandwidth choice, yielding comparable effective windows on either side of the cutoff across scenarios.

Inference performance is the primary strength of rdrobust. Empirical coverage of nominal 95 percent confidence intervals is consistently close to target across baseline, curvature, confounding, and heterogeneity designs, and remains conservative under weak first stages. This stands in contrast to classical local estimators, whose standard errors can be poorly calibrated when curvature, weak compliance, or finite-sample effects are present.

The robustness of rdrobust arises from two features working jointly: explicit bias correction for local polynomial estimation at the boundary and variance estimation that accounts for the resulting smoothing uncertainty. As a result, the method prioritizes correct uncertainty quantification rather than minimizing point-estimate variance.

Overall, these results support the use of rdrobust as a best-practice tool for inference in fuzzy regression discontinuity designs. While it is not designed to dominate simpler estimators in terms of raw point-estimate precision, it consistently delivers well-calibrated confidence intervals and stable behavior across a wide range of data-generating conditions. This makes rdrobust the appropriate reference standard for inference throughout the remainder of the analysis.

Table 3: rdrobust benchmark ($n = 10{,}000$): bias-corrected fuzzy RD estimates with robust standard errors, MSE-optimal bandwidths, and empirical 95% coverage.

Scenario	MC mean($\hat{\tau}$)	Bias	RMSE	SD($\hat{\tau}$)	Mean $\widehat{\mathrm{SE}}_{\mathrm{rb}}$	Coverage (95%)	Mean h_ℓ	Mean h_r
S1_baseline	0.986	-0.014	0.399	0.399	0.390	0.945	0.313	0.313
S2_curvature	0.991	-0.009	0.406	0.407	0.387	0.940	0.307	0.307
S3_weakFS	0.931	-0.069	2.251	2.256	1.493	0.990	0.311	0.311
S4_confounding	1.002	0.002	0.437	0.438	0.493	0.975	0.315	0.315
S5_heterogeneity	0.991	-0.009	0.427	0.428	0.415	0.950	0.316	0.316

Density Continuity

Density continuity at the cutoff is assessed using the McCrary (2008) test. Across all scenarios and sample sizes, estimated test statistics are small and provide no evidence of a discontinuity in the running-variable density at the threshold. No systematic excess mass is detected on either side of the cutoff, consistent with density continuity in finite samples. Full results are reported in Table 5 (Appendix C).

Placebo Tests

Placebo tests are conducted to verify that the estimation procedure does not detect discontinuities where none should exist. Because the fuzzy Wald estimand is undefined away from the true cutoff, placebo analyses use sharp-RDD specifications applied to the outcome at false cutoffs.

Across all scenarios, placebo estimates are tightly centered near zero, with dispersion and empirical coverage close to nominal levels. No systematic placebo effects are observed, indicating that the estimation procedure does not spuriously generate discontinuities due to curvature, bandwidth choice, or finite-sample artifacts.

Discussion

This simulation study was designed to clarify how fuzzy regression discontinuity designs behave in practice, with emphasis on applied implementation rather than estimator optimization. The results highlight where the design is reliable, where it becomes fragile, and how those limits should inform empirical practice.

The findings reaffirm fuzzy RDD as a locally credible IV design when its identifying assumptions hold in a neighborhood of the cutoff. Under continuity, no manipulation, and a nontrivial treatment discontinuity, design-aligned estimators recover the intended complier LATE even in the presence of curvature, smooth confounding, or treatment-effect heterogeneity. The strength of the design lies in isolating quasi-experimental variation locally, rather than in modeling global outcome relationships.

At the same time, the results make clear that fuzzy RDD inherits the fundamental constraints of local IV methods. When compliance is weak, identifying variation is limited, and estimator behavior can deteriorate sharply. In such settings, point estimates become unstable and sensitive to implementation choices, even though the underlying estimand remains well defined. This limits the feasibility of precise estimation and, in particular, of subgroup or heterogeneity analyses unless sample size and compliance strength are sufficient.

The simulations also underscore the distinction between estimator stability and inferential validity. Under weak compliance, robust inference procedures can continue to deliver well-calibrated uncertainty statements even when point estimates are highly variable. This reinforces the practical importance of treating inference, rather than point estimation alone, as the primary output of fuzzy RDD analyses in challenging designs.

Finally, the diagnostic and falsification checks confirm that the observed performance patterns reflect properties of the estimators and identification strength rather than artifacts of the data-generating process. Density continuity and placebo tests behave as expected across scenarios, supporting the interpretation that estimated discontinuities at the true cutoff arise from the treatment mechanism itself.

Overall, the results point to a clear applied takeaway: fuzzy RDD is a powerful design when local compliance is sufficiently strong, but it offers limited protection against weak identification. Careful attention to first-stage diagnostics, realistic expectations about precision, and the use of robust inference methods are essential for credible empirical applications.

Conclusion

This paper provides a simulation-based assessment of fuzzy regression discontinuity estimators under conditions that commonly shape empirical performance. When the identifying assumptions hold locally and the first stage is sufficiently strong, design-aware estimators recover the complier LATE with small bias. Practical fragility arises primarily through weak compliance and bandwidth choice, which can induce substantial variance inflation and, in extreme cases, nontrivial bias.

Naïve estimators can appear favorable under variance-based metrics in smooth settings, but that performance is not evidence of causal validity because these estimators target a different quantity. Sharp RDD estimators applied to fuzzy designs remain biased regardless of tuning, underscoring that estimator choice is as important as bandwidth selection.

Across all scenarios, rdrobust delivers the most reliable inference behavior, with empirical coverage close to the nominal 95 percent level even when point estimates become highly variable under weak compliance. These results support a clear applied recommendation within the scope of this study: treat first-stage diagnostics as central, interpret bandwidth sensitivity through the lens of local instrument strength, and prefer bias-corrected procedures such as rdrobust when valid inference is the primary goal.

Bibliography

Cunningham, S. (2021). Regression Discontinuity. In Causal Inference: The Mixtape. Yale University Press. Retrieved from https://mixtape.scunning.com/06-regression_discontinuity.

Chernozhukov, V., Hansen, C., Kallus, N., Spindler, M., & Syrgkanis, V. (2025). Regression Discontinuity Designs (Chapter 17). In Applied Causal Inference Powered by Machine Learning and AI (Version 0.1.1). MIT, Chicago Booth, Cornell University, Hamburg University, and Stanford University. Retrieved from https://chapters.causalml-book.org/CausalML_chap_17.pdf.

Sharpe, M. (2023). Fuzzy Regression Discontinuity Design. In Data Analysis: A Primer for Social Scientists. Retrieved from https://bookdown.org/mike/data_analysis/sec-fuzzy-regression-discontinuity-design.html.

Harris, C. R., Millman, K. J., van der Walt, S. J., et al. (2020). Array programming with NumPy. Nature, 585, 357–362.

McKinney, W. (2010). Data structures for statistical computing in Python. Proceedings of the 9th Python in Science Conference.

Seabold, S., & Perktold, J. (2010). Statsmodels: Econometric and statistical modeling with Python. Proceedings of the 9th Python in Science Conference.

Python Software Foundation. (2024). Python Language Reference. https://www.python.org.

R Core Team. (2025). R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing, Vienna, Austria.

Calonico, S., Cattaneo, M. D., Farrell, M. H., & Titiunik, R. (2022). rdrobust: Robust Data-Driven Statistical Inference in Regression-Discontinuity Designs. R package.

Appendix

A: Data Generating Process and Validation

Data Generating Process

This appendix documents diagnostic checks confirming that the data generating process (DGP) used in the simulation study satisfies the structural requirements of a fuzzy regression discontinuity design. The full DGP specification is provided in the main text. No estimators are introduced here. All validation checks are conducted using very large samples to isolate deterministic features of the design from Monte Carlo noise.

DGP Validation

Local First-Stage Check

We verify that the DGP generates the intended discontinuity in treatment probability at the cutoff by computing local differences in mean treatment rates within symmetric windows. Using a very large sample (n = 10,000,000), the empirical first-stage jump converges to the discontinuity parameter δ plus a small deterministic contribution from the smooth baseline compliance function.

In the strong first-stage scenarios ($\delta = 0.25$), the observed jumps stabilize between approximately 0.26 and 0.30 across bandwidths. In the weak first-stage scenario ($\delta = 0.10$), the jump stabilizes between approximately 0.11 and 0.15. These results confirm that the DGP induces the intended treatment discontinuity while preserving smooth behavior away from the cutoff.

Local Reduced-Form Check

We compute local mean differences in outcomes within symmetric windows around the cutoff to verify the reduced-form discontinuity. As sample size grows, the reduced-form jump converges to $\tau \cdot \delta$ plus a small deterministic contribution from the smooth outcome surface and compliance function.

In the strong first-stage scenarios, the reduced-form jump converges to approximately 0.28–0.40. In the weak first-stage scenario, it converges to approximately 0.13–0.25. These magnitudes align with the theoretical targets implied by $\tau \cdot \delta$, confirming that the outcome discontinuity is generated solely by treatment.

Local Wald Ratio Check

Using the same windows, we compute the naïve local Wald ratio as a diagnostic of the limiting fuzzy RD estimand prior to smoothing or bias correction. In the baseline and curvature scenarios, the Wald ratio ranges from approximately 1.07 to 1.33 as bandwidth increases, reflecting curvature in the smooth outcome surface and compliance function.

In the weak first-stage scenario, the Wald ratio inflates substantially, rising from roughly 1.16 at $h = 0.05$ to 1.65 at $h = 0.20$, consistent with amplification under weak instruments. The confounding scenario produces the largest distortions (up to approximately 1.76), while the heterogeneity scenario behaves similarly to the baseline. These patterns confirm that the naïve Wald ratio is biased under curvature, confounding, or weak compliance, motivating proper FRD estimation.

Baseline Smoothness Check for $m(X)$

To confirm that the baseline outcome surface is smooth at the cutoff, we compute local mean differences in $m(X)$ around zero. Across scenarios, these differences are small and increase deterministically with bandwidth (approximately 0.025 at $h = 0.05$, 0.05 at $h = 0.10$, and 0.10 at $h = 0.20$), reflecting only quadratic curvature. No discontinuity in $m(X)$ is present at the cutoff.

Confounding Check

In the confounding scenario, the latent confounder U is strongly correlated with the outcome ($\text{corr}(Y, U) \approx 0.69$) and moderately correlated with treatment ($\text{corr}(D, U) \approx 0.21$) in the full sample. However, local means of U remain nearly balanced around the cutoff, with only a small left–right difference (approximately 0.10). This confirms that confounding operates globally while remaining locally smooth, preserving identification.

Heterogeneous Treatment Effect Check

Scenario S5 introduces treatment-effect heterogeneity with $\tau_0 = 0.5$ and $\tau_1 = 1.5$ for two equally sized subgroups. Using a large sample and $h = 0.10$, subgroup-specific naïve Wald estimates are approximately 0.67 for $G = 0$ and 1.69 for $G = 1$, closely matching the true effects. The pooled Wald ratio is approximately 1.18, consistent with the complier-weighted average. This confirms correct encoding of heterogeneity in the DGP.

Bandwidth Sanity Checks

As a final diagnostic, we examine local sample sizes and discontinuities for representative simulated datasets at bandwidths $h = 0.05$, 0.10, and 0.20. All bandwidths retain substantial observations on both sides of the cutoff, exhibit the intended treatment discontinuity, and produce visible jumps in both treatment take-up and outcomes. These checks confirm that the bandwidths used in the analysis provide adequate local support and are appropriate for the estimator comparisons and sensitivity analyses.

B: Formal Definitions of Estimators

This appendix provides formal definitions of all estimators evaluated in the simulation study. Unless otherwise noted, all expectations and limits are taken with respect to the running variable X relative to the cutoff c. Define the cutoff indicator

$$Z_i = \mathbf{1}(X_i \geq c),$$

and let h denote a chosen bandwidth; local estimators use the sample

$$\mathcal{S}_h = \{i : |X_i - c| \leq h\}.$$

Naïve OLS (ignoring the cutoff)

The naïve estimator fits a global linear regression of Y on D and X, ignoring the discontinuity:

$$\hat{\beta}_{\text{naive}} = \arg\min_{\beta} \sum_i (Y_i - \beta_0 - \beta_1 D_i - \beta_2 X_i)^2.$$

The naïve treatment-effect estimate is $\hat{\tau}_{\text{naive}} = \hat{\beta}_1$.

Incorrect Sharp-RDD Estimator

The incorrect sharp estimator assumes deterministic treatment at the cutoff and regresses the outcome on the cutoff indicator:

$$\hat{\tau}_{\text{sharp}} = \arg\min_{\tau} \sum_i (Y_i - \gamma_0 - \gamma_1 Z_i)^2,$$

estimated on $\mathcal{S}_h$. Formally, $\hat{\tau}_{\text{sharp}} = \hat{\gamma}_1$.

Fuzzy Wald Estimator

The nonparametric fuzzy RDD estimand is the local Wald ratio:

$$\tau_{\text{FRD}} = \frac{\lim_{x \downarrow c} E[Y \mid X = x] - \lim_{x \uparrow c} E[Y \mid X = x]}{\lim_{x \downarrow c} E[D \mid X = x] - \lim_{x \uparrow c} E[D \mid X = x]}.$$

The sample analogue uses separate local polynomial regressions for Y and D:

$$\widehat{\Delta Y} = \widehat{E}[Y \mid X = c^{+}] - \widehat{E}[Y \mid X = c^{-}], \quad \widehat{\Delta D} = \widehat{E}[D \mid X = c^{+}] - \widehat{E}[D \mid X = c^{-}],$$

with

$$\widehat{\tau}_{\text{Wald}} = \frac{\widehat{\Delta Y}}{\widehat{\Delta D}}.$$

Here $\widehat{E}[Y \mid X = c^{+}]$ denotes the right–hand limit estimated by evaluating the fitted local polynomial regression on $X > c$ at the cutoff; $\widehat{E}[Y \mid X = c^{-}]$ is defined analogously for $X < c$. The same definitions apply to $\widehat{E}[D \mid X = c^{+}]$ and $\widehat{E}[D \mid X = c^{-}]$.

Local 2SLS (Local IV)

Local 2SLS fits the fully interacted polynomial specification within $\mathcal{S}_h$, estimating the first stage and reduced form separately while allowing the functional form to differ on either side of the cutoff.

First stage:

$$D_i = \alpha_0 + \alpha_1 Z_i + f_{-}(X_i - c)(1 - Z_i) + f_{+}(X_i - c)Z_i + u_i,$$

where f_{-} and f_{+} are polynomials of degree p estimated separately on each side.

Reduced form:

$$Y_i = \beta_0 + \beta_1 Z_i + g_{-}(X_i - c)(1 - Z_i) + g_{+}(X_i - c)Z_i + \varepsilon_i.$$

Fitted first-stage treatment:

$$\widehat{D}_i = \hat{\alpha}_0 + \hat{\alpha}_1 Z_i + \hat{f}_{-}(X_i - c)(1 - Z_i) + \hat{f}_{+}(X_i - c)Z_i.$$

Second stage:

$$Y_i = \hat{\gamma}_0 + \hat{\gamma}_1 \widehat{D}_i + \hat{h}_{-}(X_i - c)(1 - Z_i) + \hat{h}_{+}(X_i - c)Z_i + \hat{\eta}_i.$$

The local 2SLS estimator is $\widehat{\tau}_{\text{2SLS}} = \hat{\gamma}_1$. Because the model is just-identified, the 2SLS coefficient satisfies the algebraic identity

$$\widehat{\tau}_{\text{2SLS}} = \frac{\widehat{\beta}_1}{\widehat{\alpha}_1},$$

matching the Wald ratio.

Bias-Corrected Local Polynomial FRD (rdrobust)

Let

$$\widehat{E}^{(1)}[Y \mid X = c^+], \quad \widehat{E}^{(1)}[Y \mid X = c^-]$$

denote the conventional right- and left-hand local linear estimates of the conditional expectation of Y at the cutoff. Let $\widehat{b}_{Y,+}(c)$ and $\widehat{b}_{Y,-}(c)$ denote the corresponding bias estimates obtained from higher-order local polynomial fits.

The bias-corrected discontinuity in the outcome is

$$\widehat{\Delta Y}^{\text{BC}} = \big(\widehat{E}^{(1)}[Y \mid X = c^+] - \widehat{b}_{Y,+}(c)\big) - \big(\widehat{E}^{(1)}[Y \mid X = c^-] - \widehat{b}_{Y,-}(c)\big).$$

Define the bias-corrected first-stage jump analogously:

$$\widehat{\Delta D}^{\text{BC}} = \big(\widehat{E}^{(1)}[D \mid X = c^+] - \widehat{b}_{D,+}(c)\big) - \big(\widehat{E}^{(1)}[D \mid X = c^-] - \widehat{b}_{D,-}(c)\big).$$

The resulting CCT fuzzy RDD estimator is

$$\widehat{\tau}_{\text{CCT}} = \frac{\widehat{\Delta Y}^{\text{BC}}}{\widehat{\Delta D}^{\text{BC}}}.$$

Standard errors follow the nearest-neighbor variance estimator implemented in rdrobust, which corrects for boundary irregularities inherent in local polynomial estimation.

C: Bandwidth and Local Sample Diagnostics

This appendix documents a set of numerical diagnostics intended to verify that the implementation of the fuzzy regression discontinuity design is sensible in finite samples. The goal is not estimator evaluation or inference, but to confirm that the chosen bandwidths generate meaningful local samples, preserve the intended treatment discontinuity, and support the analyses reported in the main results.

Bandwidth Sanity Checks

We examine three symmetric bandwidths around the cutoff, $h \in \{0.05, 0.10, 0.20\}$, which correspond to the values used in the main simulation study and to the range produced by MSE-optimal selectors in rdrobust. For a representative simulated dataset, we count the number of observations falling within each window on either side of the cutoff and verify that all bandwidths retain substantial local support.

Across scenarios, even the narrowest window ($h = 0.05$) contains sufficient observations on both sides of the cutoff to support local estimation, while wider windows mechanically increase effective sample size as expected. No bandwidth produces degenerate samples or extreme imbalance between left and right neighborhoods.

Local First-Stage and Reduced-Form Diagnostics

Within each bandwidth, we compute mean treatment take-up immediately above and below the cutoff and report the implied discontinuity. As the bandwidth widens, the local sample size increases and the estimated treatment jump stabilizes, converging toward the intended first-stage magnitude specified by the DGP. This confirms that the candidate bandwidths preserve a meaningful and well-defined first stage in finite samples.

We perform the analogous check for outcomes by computing local mean differences in Y across the cutoff. The resulting reduced-form jumps are of the expected order of magnitude and scale proportionally with the treatment discontinuity, producing plausible implied Wald ratios. These calculations are used solely as numerical sanity checks and are not interpreted as estimators.

Table 4: Bandwidth sanity checks: local sample size and treatment take-up around the cutoff. Reported quantities are descriptive diagnostics only.

Bandwidth h	$n_{\|X-c\|\leq h}$	$P(D=1 \mid X \geq c)$	$P(D=1 \mid X < c)$	ΔD	$P(Z=1 \mid X \geq c)$	$P(Z=1 \mid X < c)$
0.05	110	0.660	0.579	0.081	1	0
0.10	215	0.748	0.473	0.274	1	0
0.20	401	0.774	0.497	0.276	1	0

Density Continuity (McCrary Test)

As a formal check of the no-manipulation assumption, we apply the McCrary (2008) density test to the running variable at the cutoff for each simulation scenario and sample size. The test evaluates whether the density of the running variable exhibits a discontinuity at the threshold that would indicate sorting or precise manipulation.

Across scenarios and sample sizes, the estimated test statistics are small and do not provide evidence of systematic density discontinuities at the cutoff. Reported p-values are generally well above conventional significance levels, with no consistent pattern of excess mass on either side of the threshold. These results are consistent with the construction of the data-generating process, which enforces smooth density by design.

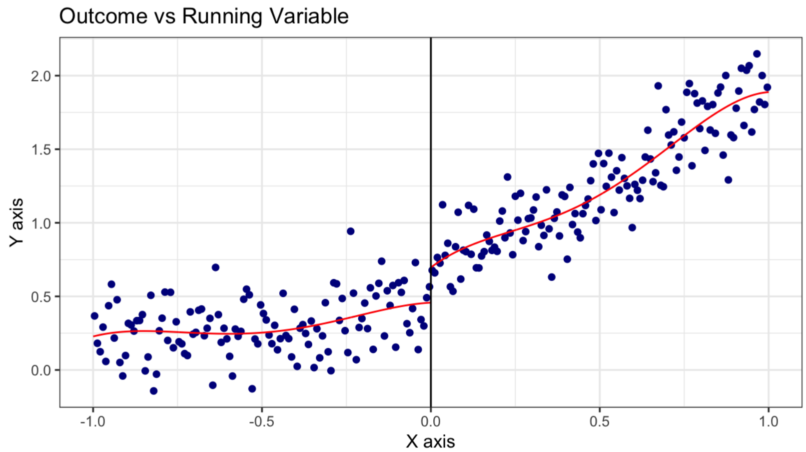
Table 5: McCrary density tests at the cutoff across simulation scenarios and sample sizes. Reported quantities assess continuity of the running-variable density and are used solely as design-validity diagnostics.

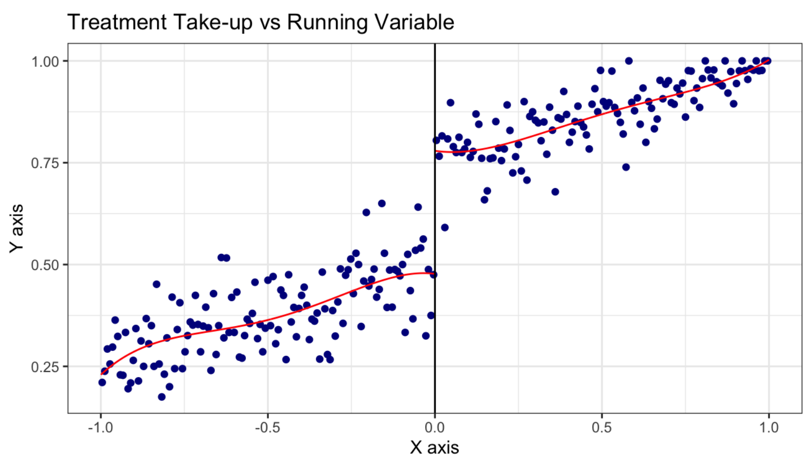
Sample size n	Scenario	McCrary test statistic	p-value
2000	S1_baseline	0.411	0.681
2000	S2_curvature	-0.576	0.565
2000	S3_weakFS	0.583	0.560
2000	S4_confounding	-0.170	0.865
2000	S5_heterogeneity	1.874	0.061
10000	S1_baseline	1.886	0.059
10000	S2_curvature	1.192	0.233
10000	S3_weakFS	0.395	0.693
10000	S4_confounding	-0.962	0.336
10000	S5_heterogeneity	1.594	0.111

Graphical RD Diagnostics

Finally, we produce standard regression discontinuity plots for both treatment take-up and the outcome as functions of the running variable. These plots display a visible and well-localized jump in treatment probability at the cutoff, alongside a corresponding jump in the outcome, with smooth behavior away from the threshold.

Together, these diagnostics confirm that the numerical implementation behaves as intended: the running variable is well populated near the cutoff, the treatment discontinuity is clearly present within all candidate bandwidths, and the local samples used by the estimators in the main analysis are well supported.





D: Supplementary Monte Carlo Tables

Local 2SLS Performance at Baseline Bandwidth

Table 6: Baseline estimator performance across all simulation scenarios, n = 10,000. Local estimators evaluated at bandwidth $h = 0.10$.

Scenario	Estimator	Bandwidth	MC mean($\hat{\tau}$)	Bias	RMSE	SD($\hat{\tau}$)
S1_baseline	Naive OLS	Global	0.999	-0.001	0.025	0.025
S1_baseline	Sharp RDD (misspecified)	h=0.10	0.323	-0.677	0.681	0.068
S1_baseline	Fuzzy Wald (local means)	h=0.10	1.178	0.178	0.286	0.225
S1_baseline	Local 2SLS	h=0.10	1.004	0.004	0.593	0.594
S2_curvature	Naive OLS	Global	1.002	0.002	0.027	0.027
S2_curvature	Sharp RDD (misspecified)	h=0.10	0.326	-0.674	0.678	0.073
S2_curvature	Fuzzy Wald (local means)	h=0.10	1.178	0.178	0.296	0.238
S2_curvature	Local 2SLS	h=0.10	1.011	0.011	0.566	0.568
S3_weakFS	Naive OLS	Global	1.001	0.001	0.023	0.023
S3_weakFS	Sharp RDD (misspecified)	h=0.10	0.173	-0.827	0.830	0.076
S3_weakFS	Fuzzy Wald (local means)	h=0.10	1.368	0.368	0.705	0.603
S3_weakFS	Local 2SLS	h=0.10	1.059	0.059	3.381	3.389
S4_confounding	Naive OLS	Global	1.004	0.004	0.035	0.034
S4_confounding	Sharp RDD (misspecified)	h=0.10	0.385	-0.615	0.620	0.079
S4_confounding	Fuzzy Wald (local means)	h=0.10	1.404	0.404	0.493	0.283
S4_confounding	Local 2SLS	h=0.10	1.019	0.019	0.610	0.611
S5_heterogeneity	Naive OLS	Global	1.000	0.000	0.025	0.025
S5_heterogeneity	Sharp RDD (misspecified)	h=0.10	0.322	-0.678	0.682	0.075
S5_heterogeneity	Fuzzy Wald (local means)	h=0.10	1.180	0.180	0.320	0.266
S5_heterogeneity	Local 2SLS	h=0.10	0.997	-0.003	0.558	0.560

Table 7: Local 2SLS performance across scenarios and sample sizes, evaluated at the baseline bandwidth $h = 0.10$.

Scenario	Sample size n	MC mean($\hat{\tau}$)	Bias	RMSE	SD($\hat{\tau}$)	Mean SE	Coverage (95%)	Mean first-stage F
S1_baseline	2000	1.310	0.310	8.433	8.449	2.928	0.945	4.452
S1_baseline	10000	1.004	0.004	0.593	0.594	0.600	0.975	19.248
S2_curvature	2000	1.123	0.123	4.811	4.821	5.874	0.960	4.955
S2_curvature	10000	1.011	0.011	0.566	0.568	0.578	0.985	20.351
S3_weakFS	2000	1600.460	1599.460	22606.463	22606.396	2865.623	0.985	1.575
S3_weakFS	10000	1.059	0.059	3.381	3.389	2.139	0.970	4.016
S4_confounding	2000	1.931	0.931	9.695	9.674	3.119	0.975	4.348
S4_confounding	10000	1.019	0.019	0.610	0.611	0.724	0.980	19.406
S5_heterogeneity	2000	1.414	0.414	3.235	3.216	2.567	0.975	4.668
S5_heterogeneity	10000	0.997	-0.003	0.558	0.560	0.613	0.970	19.745

Bandwidth Sensitivity Results

Table 8: Bandwidth sensitivity of the local 2SLS estimator across scenarios at $n = 10{,}000$, for $h \in \{0.05, 0.10, 0.20\}$. Reported quantities are Monte Carlo summaries.

Scenario	Estimator	Bandwidth h	Bias	RMSE	SD($\hat{\tau}$)	Mean SE	Coverage (95%)	Mean first-stage F
S1_baseline	local_2sls	**0.05**	-0.012	1.203	1.206	0.951	0.985	9.833
S1_baseline	local_2sls	**0.10**	0.004	0.593	0.594	0.600	0.975	19.248
S1_baseline	local_2sls	**0.20**	-0.007	0.368	0.369	0.405	0.965	38.327
S2_curvature	local_2sls	**0.05**	-0.042	0.948	0.949	0.913	0.955	10.581
S2_curvature	local_2sls	**0.10**	0.011	0.566	0.568	0.578	0.985	20.351
S2_curvature	local_2sls	**0.20**	-0.009	0.382	0.383	0.403	0.970	38.631
S3_weakFS	local_2sls	**0.05**	-12.222	124.017	123.723	25.040	0.965	2.587
S3_weakFS	local_2sls	**0.10**	0.059	3.381	3.389	2.139	0.970	4.016
S3_weakFS	local_2sls	**0.20**	0.361	5.339	5.340	1.934	0.950	6.836
S4_confounding	local_2sls	**0.05**	0.090	1.279	1.279	1.143	0.965	11.024
S4_confounding	local_2sls	**0.10**	0.019	0.610	0.611	0.724	0.980	19.406
S4_confounding	local_2sls	**0.20**	-0.014	0.425	0.426	0.498	0.985	37.947
S5_heterogeneity	local_2sls	**0.05**	-0.060	0.830	0.830	0.924	0.970	10.176
S5_heterogeneity	local_2sls	**0.10**	-0.003	0.558	0.560	0.613	0.970	19.745
S5_heterogeneity	local_2sls	**0.20**	-0.002	0.402	0.403	0.432	0.955	38.111

Weak First-Stage Monte Carlo Results

Table 9: Weak first-stage scenario (S3, $n = 10{,}000$): Monte Carlo precision, first-stage strength, and empirical coverage across bandwidths.

Scenario	Estimator	Bandwidth h	MC RMSE	SD($\hat{\tau}$)	Mean first-stage F	Mean first-stage jump	Mean SE	Coverage (95%)
S3_weakFS	local_2sls	h=0.05	124.017	123.723	2.587	NA	25.040	0.965
S3_weakFS	local_2sls	h=0.10	3.381	3.389	4.016	NA	2.139	0.970
S3_weakFS	local_2sls	h=0.20	5.339	5.340	6.836	NA	1.934	0.950
S3_weakFS	local_wald	h=0.05	1.104	1.081	NA	0.118	NA	NA
S3_weakFS	local_wald	h=0.10	0.705	0.603	NA	0.128	NA	NA
S3_weakFS	local_wald	h=0.20	0.740	0.327	NA	0.150	NA	NA

Table 10: Weak compliance (S3, $n = 10{,}000$): robust summaries for local 2SLS. Median, IQR, and a 10–90% trimmed mean summarize typical behavior, while tail probability and $\Pr(F < 1)$ quantify rare blow-ups under a weak first stage.

	Robust center and spread			Tail / weak first stage frequency	
Bandwidth h	Median($\hat{\tau}$)	IQR($\hat{\tau}$)	Trimmed mean (10–90%)	$\Pr(\lvert\hat{\tau}\rvert > 5)$	$\Pr(F < 1)$
0.05	0.940	2.567	0.980	0.125	0.390
0.10	0.896	1.631	0.963	0.050	0.240
0.20	0.918	1.257	0.947	0.030	0.075

Treatment-Effect Heterogeneity Results

Table 11: Treatment-effect heterogeneity (S5, n = 10,000): local 2SLS performance by subgroup and bandwidth.

			Monte Carlo performance			Inference / first stage		
Group	Estimator	Bandwidth h	MC mean($\hat{\tau}$)	MC bias	MC RMSE	Mean SE	Coverage (95%)	Mean first-stage F
pooled	local_2sls	h=0.05	0.940	-0.060	0.830	0.924	0.970	10.176
pooled	local_2sls	h=0.10	0.997	-0.003	0.558	0.613	0.970	19.745
pooled	local_2sls	h=0.20	0.998	-0.002	0.402	0.432	0.955	38.111
G0	local_2sls	h=0.05	0.534	0.034	1.737	1.494	0.975	5.337
G0	local_2sls	h=0.10	0.520	0.020	0.814	0.813	0.950	10.083
G0	local_2sls	h=0.20	0.492	-0.008	0.540	0.549	0.960	19.596
G1	local_2sls	h=0.05	1.451	-0.049	2.335	1.766	1.000	5.877
G1	local_2sls	h=0.10	1.464	-0.036	0.991	0.997	0.975	10.587
G1	local_2sls	h=0.20	1.503	0.003	0.622	0.671	0.970	19.547

E: Code

Python (DGP, Estimators, Monte Carlo)

```
import numpy as np
import pandas as pd
import statsmodels.api as sm
import time

def simulate_dataset(n, scenario_params, rng):
    """
    Simulate one dataset.

    Parameters
    ----------
    n : int
        Sample size.
    scenario_params : dict
        Dictionary with keys such as:
        - gamma0, gamma1, delta
        - beta0, beta1, beta2, tau
        - sigma_y
        - confounding (bool), heterogeneity (bool)
        - a0, a1, a2, sigma_u, gamma_u  (if confounding)
        - tau0, tau1                     (if heterogeneity)
    rng : np.random.Generator
        Numpy random number generator.

    Returns
    -------
    df : pandas.DataFrame
        Columns: Y, D, X, Z, G
    """

    # Running variable and cutoff
    X = rng.uniform(-1, 1, size=n)
    c = 0.0
    Z = (X >= c).astype(int)  # cutoff indicator / instrument

    # Compliance mechanism
    gamma0 = scenario_params["gamma0"]
    gamma1 = scenario_params["gamma1"]
    delta  = scenario_params["delta"]

    g = 1 / (1 + np.exp(-(gamma0 + gamma1 * X)))   # smooth baseline
    p_d = g + delta * Z                             # add discontinuity at cutoff
    p_d = np.clip(p_d, 1e-4, 1 - 1e-4)              # avoid 0/1 probabilities

    D = rng.binomial(1, p_d)

    # Baseline outcome surface m(X)
    beta0 = scenario_params["beta0"]
    beta1 = scenario_params["beta1"]
    beta2 = scenario_params["beta2"]
```

```
    tau   = scenario_params["tau"]  # reference effect (e.g., for homogeneous cases)

    m = beta0 + beta1 * X + beta2 * X**2
    eps = rng.normal(0, scenario_params["sigma_y"], size=n)

    # Build treatment effect vector (allows heterogeneity)
    if scenario_params.get("heterogeneity", False):
        # Binary subgroup, 0/1
        G = rng.binomial(1, 0.5, size=n)
        tau0 = scenario_params["tau0"]
        tau1 = scenario_params["tau1"]
        tau_vec = np.where(G == 1, tau1, tau0)
    else:
        # Homogeneous effect
        G = np.zeros(n, dtype=int)
        tau_vec = tau * np.ones(n)

    # Confounding branch
    if scenario_params.get("confounding", False):
        eta = rng.normal(0, scenario_params["sigma_u"], size=n)
        a0 = scenario_params["a0"]
        a1 = scenario_params["a1"]
        a2 = scenario_params["a2"]
        gamma_u = scenario_params["gamma_u"]

        U = a0 + a1 * X + a2 * X**2 + eta
        Y = tau_vec * D + m + gamma_u * U + eps
    else:
        # No additional confounder
        Y = tau_vec * D + m + eps

    # Return a consistent schema across all scenarios
    df = pd.DataFrame({
        "Y": Y,
        "D": D,
        "X": X,
        "Z": Z,
        "G": G  # will be 0 for all observations when heterogeneity=False
    })

    return df
```

```
scenarios = {
    "S1_baseline": {
        # First-stage / compliance
        "gamma0": 0.0,
        "gamma1": 1.0,
        "delta": 0.25,      # medium first stage

        # Outcome surface m(X) = beta0 + beta1 X + beta2 X^2
        "beta0": 0.0,
        "beta1": 0.5,
        "beta2": 0.5,       # mild curvature
```

```
        # Treatment effect and noise
        "tau": 1.0,         # homogeneous effect
        "sigma_y": 1.0,

        # Scenario flags
        "confounding": False,
        "heterogeneity": False
    },

    "S2_curvature": {
        "gamma0": 0.0,
        "gamma1": 1.0,
        "delta": 0.25,      # same first stage as S1

        "beta0": 0.0,
        "beta1": 0.5,
        "beta2": 2.0,       # stronger curvature to stress bandwidth / misspecification

        "tau": 1.0,
        "sigma_y": 1.0,

        "confounding": False,
        "heterogeneity": False
    },

    "S3_weakFS": {
        "gamma0": 0.0,
        "gamma1": 1.0,
        "delta": 0.10,      # weak first stage

        "beta0": 0.0,
        "beta1": 0.5,
        "beta2": 0.5,

        "tau": 1.0,
        "sigma_y": 1.0,

        "confounding": False,
        "heterogeneity": False
    },

    "S4_confounding": {
        "gamma0": 0.0,
        "gamma1": 1.0,
        "delta": 0.25,      # medium first stage

        "beta0": 0.0,
        "beta1": 0.5,
        "beta2": 0.5,

        "tau": 1.0,
        "sigma_y": 1.0,
```

```
        "confounding": True,
        "heterogeneity": False,

        # Smooth confounder U = a0 + a1 X + a2 X^2 + eta
        "a0": 0.0,
        "a1": 0.8,
        "a2": 0.8,
        "sigma_u": 1.0,
        "gamma_u": 0.8      # strength of confounder in outcome
    },

    "S5_heterogeneity": {
        "gamma0": 0.0,
        "gamma1": 1.0,
        "delta": 0.25,      # medium first stage

        "beta0": 0.0,
        "beta1": 0.5,
        "beta2": 0.5,

        # Group-specific effects: tau0 for G=0, tau1 for G=1
        "tau0": 0.5,
        "tau1": 1.5,
        "tau": 1.0,         # reference value for summaries (overall target)

        "sigma_y": 1.0,

        "confounding": False,
        "heterogeneity": True
    }
}
```

```
# Naive global OLS
def estimate_naive_ols(df, tau_true=None):
    """
    Naïve global OLS: regress Y on D and X, ignoring the cutoff and fuzziness.

    Model: Y_i = beta0 + beta1 * D_i + beta2 * X_i + e_i
    """

    y = df["Y"].values
    X_design = np.column_stack([df["D"].values, df["X"].values])
    X_design = sm.add_constant(X_design)

    model = sm.OLS(y, X_design).fit()

    beta1 = model.params[1]
    se1   = model.bse[1]
    n     = df.shape[0]

    return {
        "estimator": "naive_ols",
        "tau_true": tau_true,
        "tau_hat": float(beta1),
```

```python
        "se_tau": float(se1),
        "n": int(n)
    }

# Incorrect sharp RDD
def estimate_sharp_local(df, h, tau_true=None, cutoff=0.0):
    """
    Incorrect sharp-RDD estimator applied to a fuzzy design.

    Within |X - cutoff| <= h, fit:
        Y_i = gamma0 + gamma1 * Z_i + e_i,
    where Z_i = 1{X_i >= cutoff}.
    """

    tmp = df[np.abs(df["X"] - cutoff) <= h].copy()
    n = tmp.shape[0]

    if n == 0:
        return {
            "estimator": "sharp_local",
            "bandwidth": h,
            "tau_true": tau_true,
            "tau_hat": np.nan,
            "se_tau": np.nan,
            "n_left": 0,
            "n_right": 0,
            "n": 0
        }

    # Ensure Z exists
    if "Z" not in tmp.columns:
        tmp["Z"] = (tmp["X"] >= cutoff).astype(int)

    n_left  = int((tmp["X"] <  cutoff).sum())
    n_right = int((tmp["X"] >= cutoff).sum())

    y = tmp["Y"].values
    Z_design = sm.add_constant(tmp["Z"].values)

    model = sm.OLS(y, Z_design).fit()
    gamma1 = model.params[1]
    se1    = model.bse[1]

    return {
        "estimator": "sharp_local",
        "bandwidth": h,
        "tau_true": tau_true,
        "tau_hat": float(gamma1),
        "se_tau": float(se1),
        "n_left": n_left,
        "n_right": n_right,
        "n": int(n)
    }
```

```python
# Local mean fuzzy Wald
def estimate_local_wald(df, h, tau_true=None, cutoff=0.0):
    """
    Local-mean fuzzy FRD Wald estimator using a symmetric window |X - cutoff| <= h.

        tau_hat = (jump in Y) / (jump in D)
                = (E[Y | X in [0,h]] - E[Y | X in [-h,0))) /
                  (E[D | X in [0,h]] - E[D | X in [-h,0))).
    """

    tmp = df[np.abs(df["X"] - cutoff) <= h]

    left  = tmp[tmp["X"] <  cutoff]
    right = tmp[tmp["X"] >= cutoff]

    n_left  = left.shape[0]
    n_right = right.shape[0]

    if n_left == 0 or n_right == 0:
        return {
            "estimator": "local_wald",
            "bandwidth": h,
            "tau_true": tau_true,
            "tau_hat": np.nan,
            "first_stage_jump": np.nan,
            "jump_Y": np.nan,
            "n_left": n_left,
            "n_right": n_right
        }

    y_left,  y_right  = left["Y"].mean(), right["Y"].mean()
    d_left,  d_right  = left["D"].mean(), right["D"].mean()

    jump_Y = y_right - y_left
    jump_D = d_right - d_left

    tau_hat = np.nan if jump_D == 0 else jump_Y / jump_D

    return {
        "estimator": "local_wald",
        "bandwidth": h,
        "tau_true": tau_true,
        "tau_hat": float(tau_hat),
        "first_stage_jump": float(jump_D),
        "jump_Y": float(jump_Y),
        "n_left": int(n_left),
        "n_right": int(n_right)
    }

# Local 2SLS (proper fuzzy RDD/local IV)
def estimate_local_2sls(df, h, tau_true=None, cutoff=0.0):
    """
    Local 2SLS / local IV fuzzy RDD estimator with fully interacted local-linear spec.
```

```
    Window: |X - cutoff| <= h.
    First stage:
        D_i = a0 + a1 * Z_i + a2 * Xc_i + a3 * (Z_i * Xc_i) + u_i
    Second stage:
        Y_i = g0 + g1 * D_hat_i + g2 * Xc_i + g3 * (Z_i * Xc_i) + eta_i

    Returns tau_hat = g1 and a simple first-stage F-stat on Z.
    """

    tmp = df[np.abs(df["X"] - cutoff) <= h].copy()
    n = tmp.shape[0]

    if n == 0:
        return {
            "estimator": "local_2sls",
            "bandwidth": h,
            "tau_true": tau_true,
            "tau_hat": np.nan,
            "se_tau": np.nan,
            "first_stage_F": np.nan,
            "n_left": 0,
            "n_right": 0,
            "n": 0
        }

    # Ensure Z exists
    if "Z" not in tmp.columns:
        tmp["Z"] = (tmp["X"] >= cutoff).astype(int)

    Xc = tmp["X"].values - cutoff
    Z  = tmp["Z"].values.astype(float)
    D  = tmp["D"].values
    Y  = tmp["Y"].values

    # Fully interacted local-linear controls: const, Xc, Z*Xc
    W = np.column_stack([np.ones(n), Xc, Z * Xc])   # exogenous controls
    Z_col = Z.reshape(-1, 1)

    # First stage: D ~ Z + W (so regressors: const, Xc, Z*Xc, Z)
    X_fs = np.column_stack([W, Z_col])
    fs_model = sm.OLS(D, X_fs).fit()
    D_hat = fs_model.fittedvalues

    # Extract t-stat for Z coefficient (last column) and compute F
    t_Z = fs_model.tvalues[-1]
    fs_F = float(t_Z ** 2)

    # Second stage: Y ~ D_hat + W (regressors: const, Xc, Z*Xc, D_hat)
    X_ss = np.column_stack([W, D_hat])
    ss_model = sm.OLS(Y, X_ss).fit()

    tau_hat = ss_model.params[-1]   # coefficient on D_hat
    se_tau  = ss_model.bse[-1]
```

```python
    n_left  = int((tmp["X"] <  cutoff).sum())
    n_right = int((tmp["X"] >= cutoff).sum())

    return {
        "estimator": "local_2sls",
        "bandwidth": h,
        "tau_true": tau_true,
        "tau_hat": float(tau_hat),
        "se_tau": float(se_tau),
        "first_stage_F": fs_F,
        "n_left": n_left,
        "n_right": n_right,
        "n": int(n)
    }
```

```python
# Reproducibility
rng = np.random.default_rng(123456)


start_time = time.time()

# Monte Carlo settings
B = 200
n_grid = [2000, 10000]
bandwidths = [0.05, 0.10, 0.20]

all_results = []

for n in n_grid:

    # ---- DGP output file for this n ----
    dgp_path = f"frd_dgp_n{n}.csv"
    wrote_header = False  # to control header writing on first append

    for scen_name, scen_params in scenarios.items():
        tau_ref = scen_params["tau"]

        for b in range(1, B + 1):

            # 1) Simulate one dataset
            df = simulate_dataset(n, scen_params, rng)

            # Tag with sample size, scenario, and replication id
            df["n"] = n
            df["scenario"] = scen_name
            df["rep"] = b

            # 1a) Append DGP directly to disk (avoid huge in-memory concat)
            df.to_csv(
                dgp_path,
                mode="a",
                header=(not wrote_header),
                index=False
```

```
        )
        wrote_header = True

        # 2) Run estimators (pooled)
        res_naive = estimate_naive_ols(df, tau_true=tau_ref)
        res_naive.update({
            "scenario": scen_name,
            "rep": b,
            "n": n,
            "h": np.nan,
            "group": "pooled"
        })
        all_results.append(res_naive)

        for h in bandwidths:
            res_sharp = estimate_sharp_local(df, h, tau_true=tau_ref)
            res_sharp.update({
                "scenario": scen_name,
                "rep": b,
                "n": n,
                "h": h,
                "group": "pooled"
            })
            all_results.append(res_sharp)

            res_wald = estimate_local_wald(df, h, tau_true=tau_ref)
            res_wald.update({
                "scenario": scen_name,
                "rep": b,
                "n": n,
                "h": h,
                "group": "pooled"
            })
            all_results.append(res_wald)

            res_2sls = estimate_local_2sls(df, h, tau_true=tau_ref)
            res_2sls.update({
                "scenario": scen_name,
                "rep": b,
                "n": n,
                "h": h,
                "group": "pooled"
            })
            all_results.append(res_2sls)

        # 3) Subgroup analysis in S5_heterogeneity
        if scen_name == "S5_heterogeneity":
            for group_label, df_sub in {
                "G0": df[df["G"] == 0],
                "G1": df[df["G"] == 1],
            }.items():
                if df_sub.empty:
                    continue
```

```
                    tau_true_group = scen_params["tau0"] if group_label == "G0"
                      else scen_params["tau1"]

                    res_naive_g = estimate_naive_ols(df_sub, tau_true=tau_true_group)
                    res_naive_g.update({
                        "scenario": scen_name,
                        "rep": b,
                        "n": n,
                        "h": np.nan,
                        "group": group_label
                    })
                    all_results.append(res_naive_g)

                    for h in bandwidths:
                        res_sharp_g = estimate_sharp_local(df_sub, h, tau_true=tau_true_group)
                        res_sharp_g.update({
                            "scenario": scen_name,
                            "rep": b,
                            "n": n,
                            "h": h,
                            "group": group_label
                        })
                        all_results.append(res_sharp_g)

                        res_wald_g = estimate_local_wald(df_sub, h, tau_true=tau_true_group)
                        res_wald_g.update({
                            "scenario": scen_name,
                            "rep": b,
                            "n": n,
                            "h": h,
                            "group": group_label
                        })
                        all_results.append(res_wald_g)

                        res_2sls_g = estimate_local_2sls(df_sub, h, tau_true=tau_true_group)
                        res_2sls_g.update({
                            "scenario": scen_name,
                            "rep": b,
                            "n": n,
                            "h": h,
                            "group": group_label
                        })
                        all_results.append(res_2sls_g)

# 4) Write results (all n pooled)
results_df = pd.DataFrame(all_results)
results_df.to_csv("frd_sim_results_python.csv", index=False)

end_time = time.time()
elapsed = end_time - start_time
print(f"Total time: {elapsed/60:.2f} minutes")

print("Wrote DGP files:")
```

```
for n in n_grid:
    print(f"  frd_dgp_n{n}.csv")
```

R (rdrobust sim using full Monte Carlo DGP created in Python)

```r
# ==========================
# rdrobust inference benchmark (n = 10000)
# Run rdrobust once per (scenario, rep), then extract in a separate step
# ==========================

library(tidyverse)
library(readr)
library(rdrobust)
library(knitr)
library(kableExtra)

setwd("~/Desktop/JHU Courses/Causal Inference/Final Project/")

# ---- Load DGP for n = 10000 ----
dgp <- read_csv("frd_dgp_n10000.csv", show_col_types = FALSE)
if ("...1" %in% names(dgp)) dgp <- dgp %>% select(-...1)

dgp <- dgp %>%
  mutate(
    n = as.integer(n),
    rep = as.integer(rep),
    scenario = as.character(scenario),
    X = as.numeric(X),
    Y = as.numeric(Y),
    D = as.integer(D)
  ) %>%
  filter(n == 10000)

# True tau by scenario (pooled truth)
tau_map <- tibble(
  scenario = c("S1_baseline", "S2_curvature", "S3_weakFS", "S4_confounding", "S5_heterogeneity"),
  tau_true = c(1.0, 1.0, 1.0, 1.0, 1.0)
)

# ---- 1) Run rdrobust and store the fit objects ----
keys <- dgp %>% distinct(scenario, rep)

fits_tbl <- keys %>%
  mutate(
    fit = purrr::map2(scenario, rep, ~{
      tmp <- dgp %>% filter(scenario == .x, rep == .y)

      rdrobust(
        y = tmp$Y,
        x = tmp$X,
        c = 0,
        fuzzy = tmp$D,
        masspoints = "off"
```

```r
    )
  })
)

# ---- 2) Extract (tau.bc, se.rb, bandwidths) ----
extract_from_fit <- function(fit) {
  est <- as.matrix(fit$Estimate)

  tibble(
    tau_bc = as.numeric(est[1, "tau.bc"]),
    se_rb  = as.numeric(est[1, "se.rb"]),
    h_left  = if (!is.null(fit$bws)) as.numeric(fit$bws[1, 1]) else NA_real_,
    h_right = if (!is.null(fit$bws)) as.numeric(fit$bws[1, 2]) else NA_real_
  )
}

rd_out <- fits_tbl %>%
  mutate(extract = purrr::map(fit, extract_from_fit)) %>%
  tidyr::unnest(extract) %>%
  left_join(tau_map, by = "scenario") %>%
  mutate(
    ci_lo = tau_bc - 1.96 * se_rb,
    ci_hi = tau_bc + 1.96 * se_rb,
    covered_95 = (tau_true >= ci_lo & tau_true <= ci_hi)
  )


# Summary table: bias, RMSE, SD, mean bandwidth, mean SE, and coverage
rd_summary <- rd_out %>%
  group_by(scenario) %>%
  summarise(
    sims_used = n(),
    mean_tau_hat = mean(tau_bc),
    bias = mean(tau_bc - tau_true),
    rmse = sqrt(mean((tau_bc - tau_true)^2)),
    sd_tau_hat = sd(tau_bc),
    mean_se_rb = mean(se_rb),
    coverage_95 = mean(covered_95),
    mean_h_left  = mean(h_left),
    mean_h_right = mean(h_right),
    .groups = "drop"
  ) %>%
  mutate(across(where(is.numeric), ~ round(.x, 3))) %>%
  arrange(scenario)
```