The Stata Journal (2022)
22, Number 3, pp. 597–624 DOI: 10.1177/1536867X221124534

Bunching estimation of elasticities using Stata

Marinho Bertanha
University of Notre Dame
Notre Dame, IN
mbertanha@nd.edu

Andrew H. McCallum
Board of Governors of the
Federal Reserve System
Washington, DC
andrew.h.mccallum@frb.gov

Alexis Payne
Stanford University
Stanford, CA
ampayne@stanford.edu

Nathan Seegert
University of Utah
Salt Lake City, UT
nathan.seegert@eccles.utah.edu

Abstract. Typical censoring models have mass points at the upper or lower tails, or at both tails, of an otherwise continuous outcome distribution. In contrast, we consider a censoring model with a mass point in the interior of the outcome distribution. We refer to this mass point as "bunching" and use it to estimate model parameters. For example, economic theory suggests that, for increasing marginal income tax rates, many taxpayers will report income exactly at the threshold where the tax rate increases. This translates into a censoring model with bunching at the threshold. The size of this mass point of taxpayers can be used to estimate an elasticity parameter that summarizes taxpayers' responses to taxes. In this article, we introduce the command `bunching`, which implements new nonparametric and semiparametric identification methods for estimating elasticities developed by Bertanha, McCallum, and Seegert (2021, Technical Report 2021-002, Board of Governors of the Federal Reserve System). These methods rely on weaker assumptions than what are currently made in the literature and result in meaningfully different estimates of the elasticity.

Keywords: st0684, bunching, bunchbounds, bunchtobit, bunchfilter, midcensoring, partial identification, censored regression, income elasticity, tax

1 Introduction

Censoring models apply to distributions of an outcome variable that are continuous except for a mass point at the upper or lower tails, or at both tails, of the distribution. In this article, we consider models where the mass point occurs in the interior of the outcome distribution. We refer to this class of models as "midcensoring models". Although we use the adjective "midcensoring", the mass point may be at any point in the interior of the support of the distribution of outcomes.

Previously developed methods use such a mass point, often called "bunching", to estimate model parameters. For example, economic theory suggests that, for increasing marginal income tax rates, many taxpayers will report income exactly at the threshold where the tax rate increases. This translates to a midcensoring model with a mass point in the interior of the distribution of reported income. The size of this mass point can be used to identify an important parameter of the censoring model, which is known to economists as an elasticity parameter. In this context, the elasticity parameter describes the percent change in reported income in response to a percentage point change in marginal income tax rate. More specifically, an elasticity of 0.5 means that taxpayers reduce their reported income (labor supply) by 0.5% for each one-percentage-point increase in marginal income tax rates. Section 3.1 provides simulated data and a numerical example interpreting this elasticity in more detail. In the rest of this article, we use "bunching" to refer to a mass point in the interior of an outcome distribution, and we use "bunching methods" or "bunching estimator" to refer to the statistical methods that recover elasticity parameters from data that exhibit bunching.

Using bunching to estimate elasticities began with Saez (2010), Chetty et al. (2011), and Kleven and Waseem (2013). Following these influential articles, bunching methods became a popular way to estimate elasticities in a variety of settings, such as electricity demand (Ito 2014), real estate taxes (Kopczuk and Munroe 2015), labor regulations (Garicano, Lelarge, and Van Reenan 2016), prescription drug insurance (Einav, Finkelstein, and Schrimpf 2017), marathon finishing times (Allen et al. 2017), attribute-based regulations (Ito and Sallee 2018), education (Dee et al. 2019; Caetano, Caetano, and Nielsen 2020a), minimum wage (Jales 2018; Cengiz et al. 2019), and air-pollution data manipulation (Ghanem, Shen, and Zhang 2020), among others. Differences in mass point sizes across groups has been exploited as the first stage in a two-stage approach to control for endogeneity (Chetty, Friedman, and Saez 2013; Caetano 2015; Grossman and Khalil 2020). Bunching has also been used for causal identification in Khalil and Yildiz (2022); Caetano and Maheshri (2018); Caetano, Kinsler, and Teng (2019); and Caetano, Caetano, and Nielsen (2020b). Jales and Yu (2017) connect bunching to regression discontinuity. Lastly, Kleven (2016) conducts a detailed review of the bunching literature.

In this article, we introduce a new command, `bunching`, that uses assumptions that are weaker than current bunching methods. The command `bunching` is a wrapper function for three other commands. The first is `bunchbounds`, which estimates upper and lower bounds on the bunching elasticity using a partial-identification approach. The second is `bunchtobit`, which uses a semiparametric method with covariates for point identification. The third is `bunchfilter`, which filters friction errors from the dependent variable before applying either `bunchbounds` or `bunchtobit`.

The statistical foundations for these commands were developed by Bertanha, McCallum, and Seegert (2021). That article introduces multiple methods to recover elasticities from bunching. Each method relies on different assumptions to achieve identification of the elasticity. Because these are assumptions about an unobserved distribution, it is not possible to determine which assumption is correct. However, it is possible to check whether estimates relying on different assumptions are robust across assumptions. In practice, we recommend that researchers use the `bunching` package to use different estimation methods and check that the elasticity estimates they recover are stable across those methods.

2 Bunching estimators

The application of bunching methods used by Bertanha, McCallum, and Seegert (2021) and this article derives from bunching behavior caused by progressive marginal income taxes, as in Saez (2010). Formally, agents maximize an isoelastic quasilinear utility function of total consumption (or disposable income) and labor, which results in a data-generating process (DGP) for optimal reported taxable income as follows:

$$y_i = \begin{cases} \varepsilon s_0 + n_i^*, & \text{if } n_i^* < \underline{n}\,(k, \varepsilon, s_0) \\ k, & \text{if } \underline{n}\,(k, \varepsilon, s_0) \leq n_i^* \leq \overline{n}\,(k, \varepsilon, s_1) \\ \varepsilon s_1 + n_i^*, & \text{if } n_i^* > \overline{n}\,(k, \varepsilon, s_1) \end{cases} \tag{1}$$

y_i is the log of reported income, n_i^* is the log of unobserved heterogeneity of agent i, ε is the elasticity parameter of interest, the log of the slope of the piecewise-linear constraint changes from s_0 to s_1 at the log of the kink point k, and $s_1 < s_0$. All logs in this article are natural logs. The restriction $s_1 < s_0$ guarantees concavity of the budget set, which is fundamental for the solution in (1). In the original tax application, $s_j = \log(1 - t_j)$ and $j \in \{0, 1\}$, in which t_j is the marginal tax rate and $t_0 < t_1$. The expressions for the thresholds that determine the three cases in (1) are $\underline{n}\,(k, \varepsilon, s_0) = k - \varepsilon s_0$ and $\overline{n}\,(k, \varepsilon, s_1) = k - \varepsilon s_1$.

We use utility-maximizing agents and income taxes to motivate (1) and for exposition of the command throughout the rest of this article. Nevertheless, the methods developed by Bertanha, McCallum, and Seegert (2021), as well as the `bunching` package, apply to any dataset generated by (1). We emphasize that any data must be transformed into units that satisfy (1). In the income tax example, this is accomplished by taking logs of the outcome variable, kink, and slopes.

Our methods are applicable to nontax data. For example, Bitler, Cook, and Rothbaum (2021) study the Supplemental Nutrition Assistance Program, in which low-income individuals receive benefits for food purchases as a function of labor income, y_i. The benefit is a constant amount for labor income less than a known value, k, but decreases linearly after that. This reduction in benefits creates a piecewise linear budget set over total consumption and labor income with a kink. At $y_i = k$, the log of the slope changes from s_0 to s_1 with $s_1 < s_0$ (see Bitler, Cook, and Rothbaum [2021, fig. 1]). In this case, bunching methods identify the elasticity of labor supply, ε, with respect to the benefit reduction rate.

Another non-income-tax application is Ito (2014), who studies consumption of electricity in Southern California. Electricity price per kilowatt hour changes as a function of quantity of consumption in kilowatt hour (see figure 3 in his article). This piecewise linear pricing scheme creates a budget set over disposable income and electricity consumption with kinks. Bunching methods identify the demand elasticity with respect to electricity price.

Piecewise linear constraints frequently exhibit several kinks at different locations. `bunching` can be applied to each kink separately as long as the constraint does not have a discontinuous jump—often called a "notch"—preceding the kink under study.

Appendix B of Bertanha, McCallum, and Seegert (2021) provides a general solution to a model with multiple kinks and notches, and section 3, “Identification”, of their article discusses inference for multiple kinks.

Our estimation methods rely on (1), which maps the continuously distributed unobserved n_i^* into a mixed continuous-discrete observed distribution for y_i for given values of $(s_0, s_1, k, \varepsilon)$. For higher values of n_i^*, higher values of y_i will be observed except when n_i^* falls inside the bunching interval, that is, $[\underline{n}(k,\varepsilon,s_0), \overline{n}(k,\varepsilon,s_1)]$, in which case y_i remains constant and equal to k. Therefore, (1) leads to bunching in the distribution of y_i at the kink point k. In other words, the distribution of y_i has a mass point at k, $\mathbb{P}(y_i = k) > 0$, but is continuous otherwise. The mass of the point at k depends on the size of the bunching interval according to

$$\begin{aligned} B \equiv \mathbb{P}(y_i = k) &= \mathbb{P}\{\underline{n}(k,\varepsilon,s_0) \leq n_i^* \leq \overline{n}(k,\varepsilon,s_1)\} \\ &= F_{n^*}\{\overline{n}(k,\varepsilon,s_1)\} - F_{n^*}\{\underline{n}(k,\varepsilon,s_0)\} \end{aligned} \tag{2}$$

in which F_{n^*} is the cumulative distribution function (CDF) of the unobserved n^*.

The data and model formally consist of five elements: 1) the CDF of the outcome F_y; 2) the kink point k; 3) the slopes of the budget constraint on the left, s_0, and right, s_1, of the kink point; 4) the CDF of unobserved heterogeneity, F_{n^*}; and 5) the elasticity ε. Equation (1) maps elements 2–5 into the observed CDF, F_y. The researcher observes elements 1–3 but not the last two elements, F_{n^*} and ε.

Original bunching estimators recover ε in two steps (Saez 2010; Chetty et al. 2011). First, they assume a specific function F_{n^*} over the bunching interval. Second, they invert (2) to recover ε using their assumption about F_{n^*}. The methods developed by Bertanha, McCallum, and Seegert (2021) that are implemented by the `bunching` command are quite different than these original approaches.

`bunching` implements two novel identification strategies for the elasticity using a mass point at a kink.

The first strategy partially identifies the elasticity by assuming Lipschitz continuity and is implemented by `bunchbounds`. In other words, it assumes that the probability density function (PDF) of the unobserved heterogeneity has bounded slope magnitude. How this assumption recovers the elasticity is as follows. The observed bunching mass equals the area under the heterogeneity PDF inside an interval. The size of this bunching interval is a function of the unknown elasticity parameter. The highest and lowest values for possible PDFs inside the bunching interval are set by the Lipschitz bound on the slope magnitude of the PDFs. With a fixed bunching mass, these PDF bounds determine the maximum and minimum widths of the bunching interval and imply lower and upper bounds for the elasticity. `bunchbounds` has two particularly valuable features. First, when bunching is observed, the elasticity lower bound must be positive. Second, the bunching estimator based on the trapezoidal approximation (Saez 2010) is always within the bounds (partially identified set of elasticities).

The second strategy rewrites (1) as a midcensored regression model and is implemented by `bunchtobit`. The method assumes that the unobserved heterogeneity conditional on covariates follows a normal distribution, but we prove that conditional normality is not required for consistency of the elasticity when the unconditional distribution of income is correctly specified. This approach effectively assumes that the unconditional distribution of heterogeneity belongs to a semiparametric family of normal mixtures. Conditional normality implies a tobit model that has a globally concave log likelihood that is easy to maximize. `bunchtobit` also truncates the sample using a sequence of smaller windows around the kink point. Consistency of the elasticity using these smaller windows requires weaker assumptions on the distribution of heterogeneity because the model tends to better fit the unconditional distribution of income as the window size decreases. To the best of our knowledge, this is the first bunching estimation strategy that uses covariates and semiparametric assumptions to recover the elasticity. Covariates can control for a substantial amount of individual heterogeneity, and `bunchtobit` only places assumptions on the remaining portion of heterogeneity that is unobserved. In general, researchers should prefer methods that control for observable heterogeneity using covariates over methods that omit covariates and instead restrict both observed and unobserved heterogeneity.

Many datasets have friction errors that imply that the bunching mass is dispersed in a small interval near, instead of exactly at, the kink. When friction errors are present, they must first be filtered out before a bunching estimation method can be applied. The procedure implemented by `bunchfilter` is a practical way of removing friction errors and works well when 1) the researcher has an accurate prior on the support of the friction error distribution, 2) the friction error affects nonbunching individuals more than it affects bunching individuals, or 3) the friction error has a small variance. A more general filtering method requires deconvolution theory, which is an active area of research.

2.1 The bunchbounds command

`bunchbounds` uses bunching to partially identify the elasticity of a response variable with respect to changes in the slope of the budget set. The syntax, options, and description of this command are as follows:

2.1.1 Syntax for bunchbounds

`bunchbounds` *depvar* [*if*] [*in*] [*weight*], `kink(`#`) s0(`#`) s1(`#`) m(`#`)` [`nopic`
`savingbounds(`*filename*[`, replace`]`)`]

depvar must be one dependent variable (the response in logs in many applications).

`fweight`s are allowed; see [U] **11.1.6 weight**.

2.1.2 Options for bunchbounds

`kink(`#`)` is the location of the kink point and must be a real number in the same units as the response variable. `kink()` is required.

`s0(`#`)` is a real number. In many applications, it is the log of the slope before the kink point. `s0()` is required.

`s1(`#`)` must be a real number that is strictly less than `s0()`. In many applications, it is the log of the slope after the kink point. `s1()` is required.

`m(`#`)` is the maximum magnitude of the heterogeneity PDF slope and must be a strictly positive real number. `m()` is required.

`nopic` suppresses displaying graphs. The default is to display graphs.

`savingbounds(`*filename*[`, replace`]`)` saves *filename*`.dta` with coordinates of the partially identified set as a function of the slope magnitude of the heterogeneity distribution. Use `replace` if *filename*`.dta` already exists in the working directory.

2.1.3 Description for bunchbounds

The user enters the name of the response variable, the location of the kink point, the slopes before and after the kink point, and the maximum slope magnitude of the heterogeneity PDF. Before applying the command, all of these entries must be transformed into units that satisfy the DGP from (1). For example, in the tax setting of Saez (2010), dollars of taxable income and the dollar value of the kink point are transformed by taking logs, and the slopes are the log of 1 minus the respective marginal tax.

`bunchbounds` computes the maximum and minimum values of the elasticity that are consistent with the slope restriction on the PDF specified in `m()`, the observed distribution of the response variable, and values of the PDF of the response variable evaluated at the left and right limits approaching the kink. These limits are computed nonparametrically using the method of Cattaneo, Jansson, and Ma (2020) as implemented by their package `lpdensity`, discussed by Cattaneo, Jansson, and Ma (2022). Thus, the user needs to install `lpdensity` before using `bunchbounds`.

It is important to emphasize that the true value of the slope magnitude is unknowable, but `bunchbounds` provides four sample values as suggestions for the user. The first two sample values are estimated using the continuous part of the distribution. Specifically, minimum and maximum slope magnitude sample values are constructed from a histogram of the dependent variable that excludes the kink point and uses a bin width that is half the default bin width for the command `histogram`. The third sample value is the maximum slope magnitude that results in a finite upper bound on the elasticity. The fourth sample value is the minimum slope magnitude for which the elasticity bounds exist and are equal. This is the same elasticity estimate that one obtains with the trapezoidal approximation made by Saez (2010). `bunchbounds` outputs elasticity bounds for three values of the slope: trapezoidal approximation, user-provided slope magnitude `m()`, and the maximum slope magnitude that results in a finite upper bound.

2.2 The bunchtobit command

`bunchtobit` uses bunching, tobit regressions, and covariates to identify the elasticity of a response variable with respect to changes in the slope of the budget set. The syntax, options, and description of this command are as follows.

2.2.1 Syntax for bunchtobit

`bunchtobit` *depvar* [*indepvars*] [*if*] [*in*] [*weight*], `kink(`#`)` `s0(`#`)` `s1(`#`)`
  [`binwidth(`#`)` `grid(`*numlist*`)` `nopic` `numiter(`#`)`
  `savingtobit(`*filename*[`, replace`]`)` `verbose`]

depvar must be one dependent variable (the response in logs in many applications).

indepvars is a variable list of covariates. Heterogeneity is a linear function of these covariates and an unobserved error that is normally distributed conditional on these covariates.

`fweight`s are allowed; see [U] **11.1.6 weight**.

2.2.2 Options for bunchtobit

`kink(`#`)` is the location of the kink point and must be a real number in the same units as the response variable. `kink()` is required.

`s0(`#`)` is a real number. In many applications, it is the log of the slope before the kink point. `s0()` is required.

`s1(`#`)` must be a real number that is strictly less than `s0()`. In many applications, it is the log of the slope after the kink point. `s1()` is required.

`binwidth(#)` is the width of the bins for the histograms. It must be a strictly positive real number. The default value is half of what is automatically produced by the command `histogram`.

`grid(`*numlist*`)` is a list of integers from 1 to 99. The values in *numlist* correspond to percentages of the sample that define symmetric truncation windows around the kink point. The truncated tobit model is fit on each of these samples and also the full sample so that the number of estimates is always one more than the number of entries in *numlist*. For example, if `grid(15 82)` is specified, then `bunchtobit` fits the tobit model three times using 100%, 82%, and 15% of the data around the kink point. The default is `grid(10(10)90)`, which provides 10 estimates.

`nopic` suppresses displaying graphs. The default is to display graphs.

`numiter(#)` is the maximum number of iterations allowed when maximizing the tobit log likelihood. It must be a positive integer. The default is `numiter(500)`.

`savingtobit(`*filename*[`, replace`]`)` saves *filename*`.dta` with tobit estimates for each truncation window. The *filename*`.dta` file contains eight variables corresponding to the matrices that the code stores in `r()`. See section 3.3.1 for more details. Use `replace` if *filename*`.dta` already exists in the working directory.

`verbose` displays detailed output from the tobit estimation including iterations of maximizing the log likelihood. Nonverbose mode is the default.

2.2.3 Description for bunchtobit

The user enters the name of the response variable, the location of the kink point, and the slopes before and after the kink point. Before applying the command, all of these entries must be transformed into units that satisfy the DGP from (1). For example, in the tax setting of Saez (2010), dollars of taxable income and the dollar value of the kink point are transformed by taking logs, and the slopes must be input as the log of 1 minus the marginal tax rates.

`bunchtobit` estimates multiple midcensored tobit regressions using specified subsamples of the data. It starts with the entire sample, and then it truncates the sample to symmetric windows centered at the kink as specified by the user. The elasticity estimate is plotted as a function of the percentage of data used in each truncation window. The code also plots the histogram of the response variable along with the best-fit tobit distribution for each truncation window.

The user has the option of entering covariates that help explain the unobserved heterogeneity. Lemma 2 by Bertanha, McCallum, and Seegert (2021) demonstrates that the distribution of the unobserved heterogeneity conditional on covariates does not need to be normal for the tobit estimates to be consistent. Consistency requires that 1) the unconditional distribution of heterogeneity is a semiparametric mixture of normal distributions averaged over the included covariates and 2) the unconditional distribution of the response variable predicted by the tobit model fits the observed distribution of the response variable well. If the user does not enter covariates, then the unconditional distribution of heterogeneity needs to be normal.

3 Examples for bunchbounds and bunchtobit

In this section, we use simulated data to illustrate `bunchbounds` and `bunchtobit`. These examples are motivated by the Earned Income Tax Credit that is investigated by Saez (2010) and Bertanha, McCallum, and Seegert (2021). As such, sometimes we refer to the simulated outcome data as "earnings" and the slope of the incentive schedule as "marginal tax rates". The units of the outcome also correspond to log thousands of dollars.

3.1 Simulated data

We consider a DGP from (1) with one kink at $k = \log(8) = 2.079$ given by

$$y_i = \begin{cases} 0.5\log(1.3) + n_i^*, & \text{if } n_i^* < \log(8) - 0.5\log(1.3) \\ \log(8), & \text{if } \log(8) - 0.5\log(1.3) \leq n_i^* \leq \log(8) - 0.5\log(0.9) \\ 0.5\log(0.9) + n_i^*, & \text{if } n_i^* > \log(8) - 0.5\log(0.9) \end{cases} \tag{3}$$

in which the elasticity is $\varepsilon = 0.5$ and the slopes of the budget constraint to the left and right of the kink are $s_0 = \log(1.3) = 0.2624$ and $s_1 = \log(0.9) = -0.1054$ (representing tax rates of $t_0 = -0.3$ and $t_1 = 0.1$). To be concrete, the income tax rate changes from -30% to 10%, a 40-percentage-point increase, and translates into a slope change in the budget set of $-0.368 = \log(0.9) - \log(1.3)$. The elasticity of 0.5 means that taxpayers respond to this marginal tax rate increase by decreasing their labor supply (and income) by about 18.4% ($-0.184 = -0.368 \times 0.5$).

We assume that ability is a function of covariates and unobserved error given by $n_i^* = 2 - 0.2x_{1i} + 2.5x_{2i} + 0.4x_{3i} + \nu_i$, $\nu_i \sim N(0, 0.5)$. The covariates x_1, x_2, and x_3 are correlated binary variables with properties given in table 1.

Table 1. Covariates' properties

	Mean	Std. dev.
x_1	0.2	0.4
x_2	0.5	0.5
x_3	0.3	0.46

Correlations	x_1	x_2	x_3
x_1	1		
x_2	0.2	1	
x_3	0.1	0.4	1

We simulate about 1,000,000 weighted (100,000 unweighted) observations according to (3). Frequency weights are drawn from a standard uniform distribution, and we demonstrate how to use weights throughout the `bunching` package.

In figure 1, we graph the histogram of the 1,000,000 observations in 100 bins. The simulated outcome variable is bimodal because of the covariates, which highlights that the unconditional distribution is not normally distributed. We graph the budget constraint (black thick solid line) in (log income, log consumption) space. That budget set has a kink, that is, a change in slope from 1.3 to 0.9 at the value of 2.079 (black thin solid line) for log income. The histogram in the same figure shows that individuals bunch exactly at the kink point (gray bar).

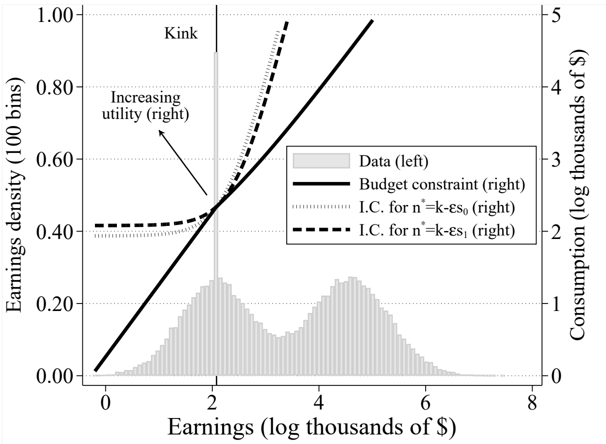


Figure 1. Histogram of simulated data

Bertanha, McCallum, and Seegert (2021) provide a complete description for how utility maximization with heterogeneous preferences and income tax brackets results in figure 1, and we provide an overview here. The heterogeneity of agents' preferences is captured by n^*, and each value of n^* corresponds to a different indifference curve (IC). We graph two specific ICs that correspond to the lower (black dotted line) and upper (black dashed line) numerical thresholds in (3), whose theoretical counterparts are $\underline{n}\,(k,\varepsilon,s_0) = k - \varepsilon s_0$ and $\overline{n}\,(k,\varepsilon,s_1) = k - \varepsilon s_1$ in (1). Many ICs that are not graphed touch the budget set at the kink. In fact, the mass point at the kink corresponds to all agents whose preference heterogeneity, n^*, lies in the bunching interval, that is, $n^* \in [\log(8) - 0.5\log(1.3)\,, \log(8) - 0.5\log(0.9)]$.

The simulated data also exhibit bunching exactly at the kink point. In many empirical applications, however, the bunching mass is dispersed in a small interval near, instead of exactly at, the kink. We provide a solution to this issue in section 4.

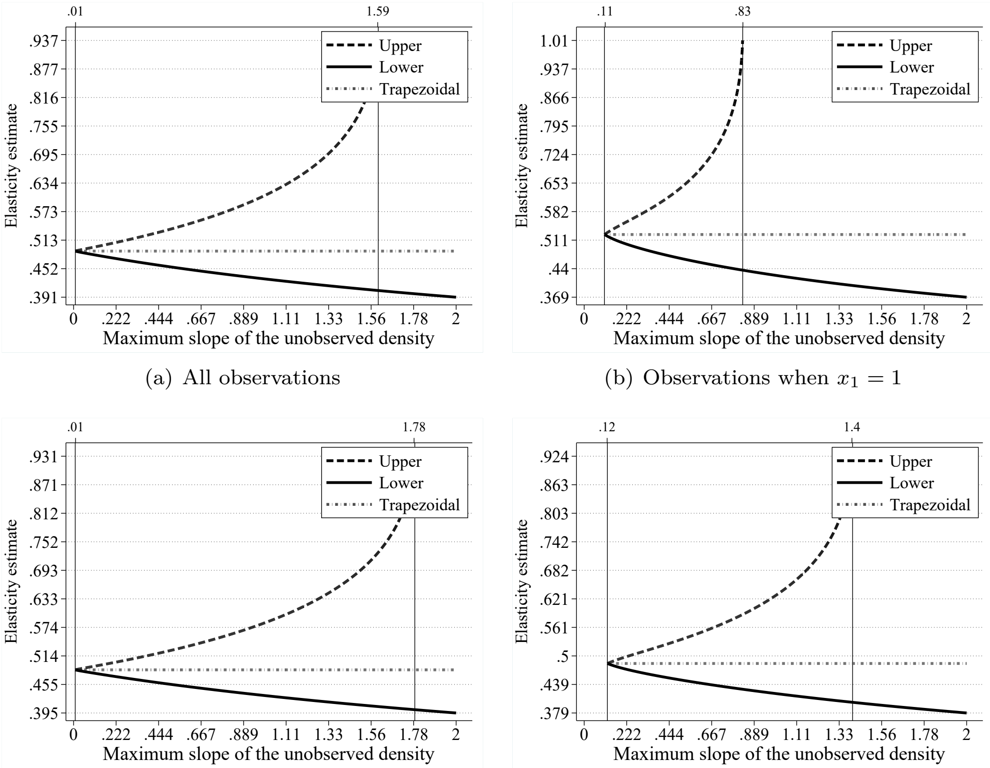
3.2 Estimating elasticity bounds

We begin by estimating the elasticity bounds using the location of the kink, $\log(8) = 2.0794$, `k(2.0794)`; tax rates on either side of the kink, `s0()` $= \log(1.3) = 0.2624$ and `s1()` $= \log(0.9) = -0.1054$; and a choice of the maximum slope, `m(2)`.

```
. use bunching
. bunchbounds y [fw=w], k(2.0794) s0(0.2624) s1(-0.1054) m(2)
Your choice of M:
2.0000
Sample values of slope magnitude M
 minimum value M in the data (continuous part of the PDF):
  0.0000
 maximum value M in the data (continuous part of the PDF):
  0.3879
 maximum choice of M for finite upper bound:
  1.5930
 minimum choice of M for existence of bounds:
  0.0090
Elasticity Estimates
 Point id., trapezoidal approx.:
  0.4894
 Partial id., M = 2.0000 :
  [0.3913 , +Inf]
 Partial id., M = 1.59 :
  [0.4055 , 0.9374]
```

The `bunchbounds` command estimates the bounds for the elasticity using different slope values. First, the output shows that we entered a maximum slope of 2 and the bounds for this slope are $[0.3913, \infty]$. Second, the command also estimates the bounds using the maximum slope for a finite upper bound, when the maximum slope given is greater than that value. In this case, the maximum slope for a finite upper bound is 1.5930, resulting in the bounds $[0.4055, 0.9374]$. In both cases, the true elasticity estimate of 0.5 is within these bounds. The output also gives the estimated minimum and maximum slopes of the continuous portion of the PDF of the data. These slopes are 0 and 0.3879. The point-identified elasticity using the trapezoidal approximation (which is the Saez [2010] estimator) of 0.4894 is also provided.

The nonparametric bounds are also graphed by `bunchbounds` for different maximum slope magnitudes of the unobserved heterogeneity PDF. These different slope magnitudes are plotted on the horizontal axis, and the corresponding bounds are plotted on the vertical axis. For this example, these are given in figure 2(a). This figure shows how the upper bound, depicted as a dashed line, increases and the lower bound, depicted as a solid line, decreases as the maximum slope increases. The vertical lines in figure 2(a) at 0.01 and 1.59 denote the minimum slope for the existence of the bounds and the maximum slope for a finite upper bound, respectively. The point-identified elasticity using the trapezoidal approximation occurs where the bounds come together—the dash-dot horizontal gray line in figure 2(a).



(a) All observations

(b) Observations when $x_1 = 1$

(c) Observations when $x_1 = 0$

(d) Observations when $x_1 = 1$ and $x_3 = 0$

Figure 2. Estimating elasticity bounds

The `bunchbounds` command can also be combined with conditional statements that restrict to subsamples of the data based on values of different covariates but cannot otherwise be conditional on covariates. For example,

```
bunchbounds y if x1 == 1 & x3 == 0 [fw=w], k(2.0794) s0(0.2624) s1(-0.1054) m(2)
```

estimates the bounds when $x_1 = 1$ and $x_3 = 0$. Restricting to subsamples when $x_1 = 1$ or $x_1 = 0$ have similar syntaxes. The output from these commands (not shown) is similar to the output without conditioning, and the bound estimates for each subsample are graphed in figures 2(b), 2(c), and 2(d). The bounds shift only slightly for each subsample because the true elasticity is 0.5 for all subsamples and because the number of weighted observations is large.

3.3 Semiparametric point estimates of the elasticity

We estimate the elasticity using a truncated tobit model that allows for covariates. Truncation and covariates provide robust estimation that relies on semiparametric assumptions and does not require the unobserved heterogeneity PDF to be normally distributed (Bertanha, McCallum, and Seegert 2021). We demonstrate the robustness of this method by comparing estimates of the correctly specified model with estimates from a misspecified model that still recover the true elasticity.

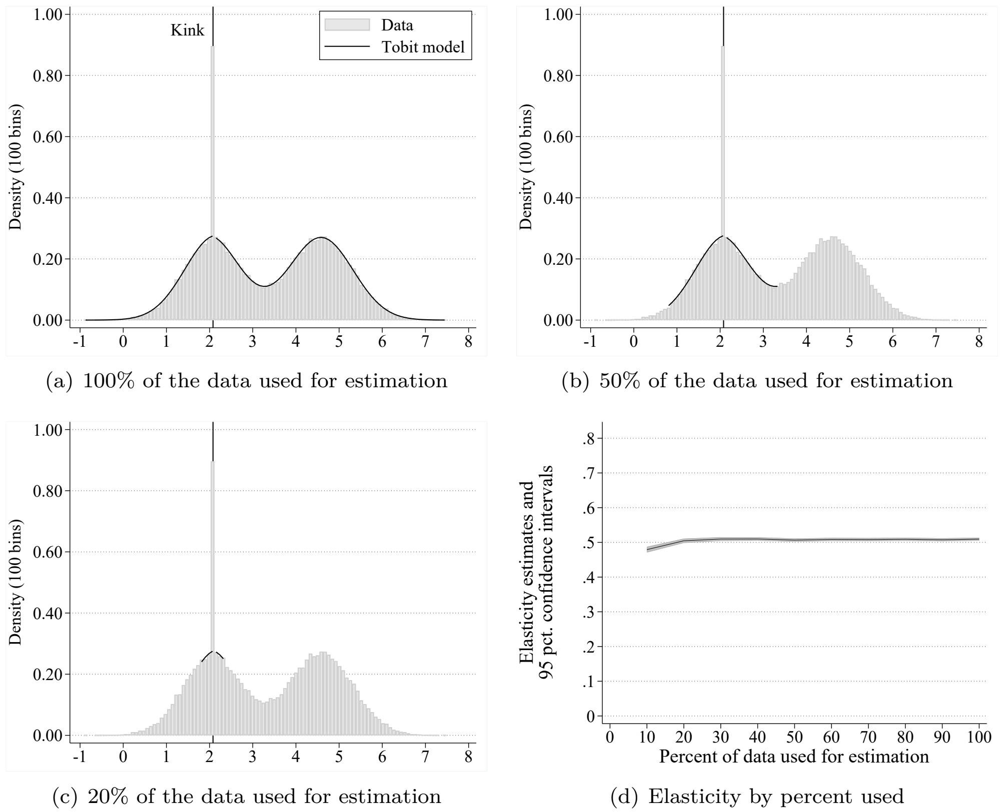
3.3.1 Correctly specified tobit model

We begin by fitting the correctly specified model using `bunchtobit`.

```
. bunchtobit y x1 x2 x3 [fw=w], k(2.0794) s0(0.2624) s1(-0.1054) binwidth(0.084)

Obtaining initial values for ML optimization.
Truncation window number 1 out of 10, 100% of data.
Truncation window number 2 out of 10, 90% of data.
Truncation window number 3 out of 10, 80% of data.
Truncation window number 4 out of 10, 70% of data.
Truncation window number 5 out of 10, 60% of data.
Truncation window number 6 out of 10, 50% of data.
Truncation window number 7 out of 10, 40% of data.
Truncation window number 8 out of 10, 30% of data.
Truncation window number 9 out of 10, 20% of data.
Truncation window number 10 out of 10, 10% of data.
bunchtobit_out[10,5]
        data %  elasticity     std err  # coll cov        flag
 1         100   .50938668   .00218386           0           0
 2          90   .50756197   .00224619           0           0
 3          80   .50898083   .00227815           0           0
 4          70   .50808053   .00229178           0           0
 5          60   .50848689   .00231719           0           0
 6          50   .50660888   .00236933           0           0
 7          40   .50975777   .00251876           0           0
 8          30   .50959025   .00273068           0           0
 9          20   .50463572   .00317585           0           0
10          10   .47913201   .00419053           0           0
```

The command estimates the elasticity for 10 different subsamples by default. The first uses all the data, the second uses 90% of the data around the kink, the third uses 80% around the kink, and so on. Estimation proceeds in 10-percentage-point intervals, declining down to the last subsample that uses only 10% of the data. Each subsample is truncated symmetrically, is centered around the kink, and includes the observations at the kink. For the data simulated by (3) and using the 90% truncated subsample as an example, about 42.5% of the data are from below the kink, about 42.5% of the data are from above the kink, and about 5% of the data are from the kink. The fraction of data at the kink does not change with this type of truncation. For example, the 10% subsample uses about 2.5% of the data above and below the kink and about 5% from the kink.

Because the model is correctly specified, the estimates reported in the `elasticity` column are always very close to the true value of 0.5 for any truncated subsample. Standard errors in column `std err` are small because the simulated data include 1,000,000 weighted observations. The standard errors increase as the percent of data used decreases because we use fewer observations. The table also reports the number of covariates that were omitted because they were collinear in column `# coll cov` and when optimizing the log likelihood did not converge to a maximum in column `flag`.



(a) 100% of the data used for estimation

(b) 50% of the data used for estimation

(c) 20% of the data used for estimation

(d) Elasticity by percent used

Figure 3. Correctly specified truncated tobit estimates

Along with this numeric output, `bunchtobit` also produces a best-fit graph for each subsample and a graph of the elasticity estimate for all subsamples. Figures 3(a), 3(b), and 3(c) display these best-fit graphs for the 100%, 50%, and 20% truncation subsamples, respectively. Each of these panels presents a histogram of y_i (gray bars) and the estimate of the correctly specified and truncated tobit model implied outcome variable (black solid line). The model is correctly specified, so it fits the data well for all truncated subsamples. Figure 3(d) plots the estimate (black solid line) and 95% confidence interval (gray shading) for each truncated subsample corresponding to the `elasticity` column. The elasticity is the main parameter of interest, but the covariate coefficients for the smallest value in *numlist* provided in `grid(`*numlist*`)` can be obtained by using the `estimates replay` command. For example, truncating to 77% of the data for the correctly specified model, and then using `estimates replay` provides the following output:

```
. bunchtobit y x1 x2 x3 [fw=w], k(2.0794) s0(0.2624) s1(-0.1054) binwidth(0.084)
> grid(77)
Obtaining initial values for ML optimization.
Truncation window number 1 out of 2, 100% of data.
Truncation window number 2 out of 2, 77% of data.

bunchtobit_out[2,5]
       data %  elasticity     std err  # coll cov        flag
1         100   .50938668   .00218386           0           0
2          77   .50849786   .00228162           0           0

. estimates replay
--------------------------------------------------------------------------------

active results
--------------------------------------------------------------------------------

Log pseudolikelihood = -.96353496                      Number of obs = 770,197

 ( 1)  [eq_l]x1 - [eq_r]x1 = 0
 ( 2)  [eq_l]x2 - [eq_r]x2 = 0
 ( 3)  [eq_l]x3 - [eq_r]x3 = 0
------------------------------------------------------------------------------
             |               Robust
             | Coefficient  std. err.      z    P>|z|     [95% conf. interval]
-------------+----------------------------------------------------------------
eq_l         |
          x1 |  -.2876614   .0035942   -80.03   0.000    -.2947059   -.2806168
          x2 |   3.541998   .0038313   924.49   0.000     3.534488    3.549507
          x3 |   .5509258   .0036639   150.37   0.000     .5437448    .5581069
       _cons |   3.022123   .0033913   891.13   0.000     3.015476     3.02877
-------------+----------------------------------------------------------------
eq_r         |
          x1 |  -.2876614   .0035942   -80.03   0.000    -.2947059   -.2806168
          x2 |   3.541998   .0038313   924.49   0.000     3.534488    3.549507
          x3 |   .5509258   .0036639   150.37   0.000     .5437448    .5581069
       _cons |   2.757436   .0035784   770.58   0.000     2.750422    2.764449
-------------+----------------------------------------------------------------
lngamma      |
       _cons |    .347303    .001056   328.87   0.000     .3452331    .3493728
-------------+----------------------------------------------------------------
       sigma |   .7065912   .0014946                      .7051302    .7080553
      cons_l |   2.135406   .0030205                      2.129486    2.141326
      cons_r |    1.94838   .0033687                      1.941778    1.954983
         eps |   .5084979   .0022816                       .504026    .5129697
------------------------------------------------------------------------------
```

Olsen (1978) introduces a reparameterization that is discussed in Hayashi (2000, chap. 8.3) that ensures the log likelihood of a classical tobit model that is globally concave. That reparameterization divides each coefficient of the covariates by the standard deviation of the errors, and we use the same reparameterization in our log likelihood. The results output by `estimates replay` report these reparameterized coefficients instead of the original coefficients. The reparameterization can be reversed by multiplying the reparameterized coefficients by the standard deviation. For example, the estimate of the coefficient on x_2 from (3) can be recovered as $3.54 \times 0.71 = 2.51$.

The elasticity reported in column `elasticity` for the 77% subsample is from the estimate `eps` in the `active results` table shown by `estimates replay`. The first equation (`eq_l`) and the coefficient estimates on x_1, x_2, and x_3 are from the left-hand side of the kink and are the same as the estimates from the second equation, `eq_r`, on the right of the kink. These coefficients are constrained to be the same on the left and right sides of the kink as reflected by the three constraints `( 1)`, `( 2)`, and `( 3)` at the top of the table and consistent with (3). Because the model is correctly specified, the covariate coefficient estimates are consistent and the estimates shown by `estimates replay` are close to the (reparameterized) truth for each coefficient.

3.3.2 Incorrectly specified tobit model

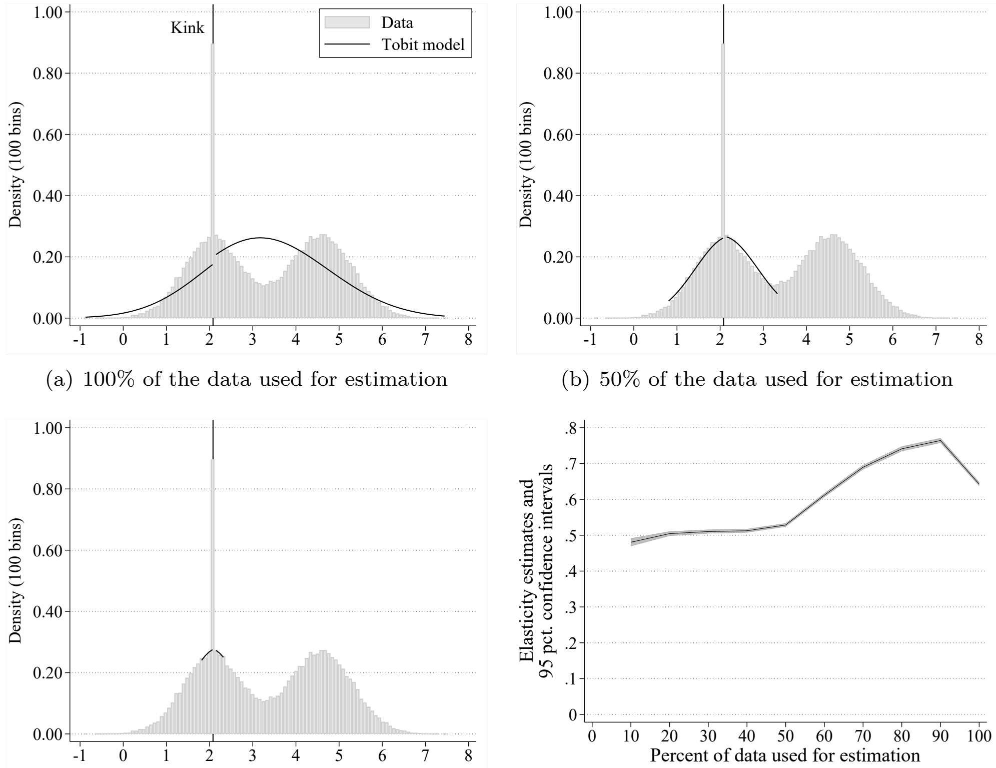
The correctly specified tobit model from the previous section satisfies the assumption that ν_i is normal and therefore always fits the observed distribution of y_i. A misspecified model that does not have normally distributed errors will not always fit the distribution of y_i well. However, Bertanha, McCallum, and Seegert (2021) prove that if the tobit model's best-fit distribution matches the observed distribution of y_i, then the tobit model estimates the elasticity consistently whether or not the distribution of ν_i is normal. This section demonstrates this robustness property using a misspecified model that does not have normal errors. Specifically, we omit the covariate x_2 and fit the following model.

```
. bunchtobit y x1 x3 [fw=w], k(2.0794) s0(0.2624) s1(-0.1054) binwidth(0.084)
Obtaining initial values for ML optimization.
Truncation window number 1 out of 10, 100% of data.
Truncation window number 2 out of 10, 90% of data.
Truncation window number 3 out of 10, 80% of data.
Truncation window number 4 out of 10, 70% of data.
Truncation window number 5 out of 10, 60% of data.
Truncation window number 6 out of 10, 50% of data.
Truncation window number 7 out of 10, 40% of data.
Truncation window number 8 out of 10, 30% of data.
Truncation window number 9 out of 10, 20% of data.
Truncation window number 10 out of 10, 10% of data.

bunchtobit_out[10,5]
        data %  elasticity     std err  # coll cov        flag
 1         100   .64269795   .00284279           0           0
 2          90    .7643775   .00347177           0           0
 3          80   .74113376   .00338469           0           0
 4          70   .68969711   .00316174           0           0
 5          60   .61191992   .00282291           0           0
 6          50   .52858461   .00248579           0           0
 7          40    .5125595   .00253649           0           0
 8          30    .5103475   .00273716           0           0
 9          20   .50446138   .00317555           0           0
10          10   .48052761    .0056067           0           0
```

The misspecified model returns an elasticity estimate of 0.643 using 100% of the data. This is a substantially biased estimate of the true elasticity of 0.5, and figure 4(a) shows that the misspecified model does not fit well.

We can truncate the sample to use data only local to the kink, however, to attenuate the effect of omitting x_2. In Bertanha, McCallum, and Seegert (2021, lemma 2), we show that if the tobit distribution of the fitted outcome [the black solid lines in figures 4(a) to 4(c)] matches the true distribution of the outcome variable (the gray bars in those figures), and the unconditional distribution of n^* is a mixture of normals, then the elasticity estimated by the tobit is consistent for the true elasticity, regardless of whether the conditional unobserved distribution, $F_{n^*|X}$, is normal.

Moreover, the smaller the truncation window, the easier it is to fit the unconditional distribution of the outcome variable with a tobit, and the stronger is our conviction that the estimate of the elasticity is consistent.



(a) 100% of the data used for estimation

(b) 50% of the data used for estimation

(c) 20% of the data used for estimation

(d) Elasticity by percent of data used

Figure 4. Incorrectly specified truncated tobit estimates

Figure 4 demonstrates that using smaller truncation windows around the kink improves the estimated distribution fit. Figure 4(b) uses 50% of the data and fits much better than the estimate that uses all the data, in figure 4(a). Figure 4(c) uses 20% of the data local to the kink and fits even better than the 50% subsample. Figure 4(d) shows that for all subsamples that use 50% of the data or less, we recover an estimate that is close to the true elasticity of 0.5. The largest truncation region for which the estimated distribution fits the observed distribution is context specific. For the example given in figure 4, using 50% of the data around the kink is the largest subsample of data that provides a good fit to the outcome distribution. But for other datasets, the largest truncation region that fits the outcome distribution well could use any fraction of the data, so it could be very small indeed.

4 Friction errors

Many datasets have friction errors, which are defined as when the bunching mass is dispersed in a small interval near, instead of exactly at, the kink. Friction errors can be caused by measurement error, optimizing frictions (Chetty et al. 2011), or other distortions. When friction errors are present, they must first be filtered out before a bunching estimation method can be applied.

The procedure implemented by `bunchfilter` is a practical way of filtering out friction errors. It works by fitting a polynomial to the empirical CDF of the response variable with friction errors, $\texttt{yfric}_i$. It excludes observations in a specified interval around the kink during estimation and allows the intercepts to differ to the left and right of that interval. The estimated CDF is then extrapolated into the excluded interval, which constitutes an estimate of the CDF of the response variable without friction errors, y_i. The inverse of the extrapolated CDF evaluated at each observation produces the filtered variable, and the difference between the intercepts at the kink provides the estimate of the bunching mass.

This filtering method produces consistent estimates of the distribution of the response variable without frictions under three conditions. First, the friction error, e_i, must be independent and identically distributed with known and bounded support. We emphasize that it is not necessary for the friction error to be mean zero, or for the distribution of friction error, $f(e_i)$, to be symmetric or parametric. Second, friction errors must only affect bunching individuals. Third, the CDF of y_i without friction errors must equal a polynomial in a known neighborhood of the kink that is bigger than the support of the friction error.

4.1 The bunchfilter command

`bunchfilter` removes friction errors from data generated by a mixed continuous-discrete distribution with one mass point plus a continuously distributed friction error. The distribution of the data with friction errors is continuous and does not have a mass point. This type of data is common in economic bunching applications. For example, the distribution of taxable income usually has a hump around the kink where the marginal tax rate changes instead of a mass point at the kink. The syntax, options, and description of this command are as follows:

4.1.1 Syntax for bunchfilter

`bunchfilter` *depvar* [*if*] [*in*] [*weight*], `kink(`#`)` `deltam(`#`)` `deltap(`#`)`
`generate(`*varname*`)` [`binwidth(`#`)` `nopic` `pctobs(`#`)` `polorder(`#`)`]

depvar must be one dependent variable (the response in logs in many applications).

`fweight`s are allowed; see [U] **11.1.6 weight**.

4.1.2 Options for bunchfilter

`kink(#)` is the location of the kink point and must be a real number in the same units as the response variable. `kink()` is required.

`deltam(#)` is the distance between the kink point and the lower bound of the support of the friction error to be filtered. It must be a real number in the same units as the response variable. `deltam()` is required.

`deltap(#)` is the distance between the kink point and the upper bound of the support of the friction error to be filtered. It must be a real number in the same units as the response variable. `deltap()` is required.

`generate(`*varname*`)` generates the filtered variable with a user-specified name of *varname*. `generate()` is required.

`binwidth(#)` is the width of the bins for the histograms. It must be a strictly positive real number. The default value is half of what is automatically produced by the command `histogram`.

`nopic` suppresses displaying graphs. The default is to display graphs.

`pctobs(#)` specifies that, for a better fit, the polynomial regression uses observations in a symmetric window around the kink point that contains # percent of the sample. It must be a positive integer between 1 and 99. The default is `pctobs(40)`.

`polorder(#)` specifies the order of polynomial for the filtering regression. It must be a positive integer between 2 and 7. The default is `polorder(7)`.

4.1.3 Description for bunchfilter

The user enters the variable to be filtered (for example, the log of income), the location of the kink, and the size of the region around the mass point that contains the hump (in other words, `kink()` − `deltam()`, `kink()` + `deltap()`). `bunchfilter` fits a polynomial regression to the empirical CDF of the variable observed with error. This regression excludes points in the hump window and has a dummy for observations on the left or right of the kink. The fitted regression is used to predict values of the empirical CDF in the hump window with a jump discontinuity at the mass point. The filtered variable is then recovered from the inverse of the predicted CDF evaluated at the empirical CDF value for each observation in the sample.

This procedure works well for cases where the friction error has bounded support and only affects observations that would be at the kink in the absence of errors. A proper deconvolution theory still needs to be developed for a filtering procedure with general validity.

4.2 Example for bunchfilter

We show how to remove the friction errors as a precursor to estimating the relevant elasticity in this example. We simulate the outcome variable with friction errors as

$$\texttt{yfric}_i = y_i + e_i \mathbb{I}\left\{y_i = \log\left(8\right)\right\} \tag{4}$$

in which y_i is from (3), e_i are independent and identically distributed truncated normal from $f\left(e_i\right) = \phi\left(e_i\right) / \left[\Phi\left\{\log\left(1.1\right)\right\} - \Phi\left\{\log\left(0.9\right)\right\}\right]$, the standard normal PDF is $\phi\left(\cdot\right)$, and $\Phi\left(\cdot\right)$ is the standard normal CDF. The errors have known and bounded support $\left[\log\left(0.9\right), \log\left(1.1\right)\right]$, which ensures that frictions never add to or subtract from y_i by more than log 10%. The three conditions needed for `bunchfilter` to consistently estimate y_i, discussed in section 4, are satisfied by (4).

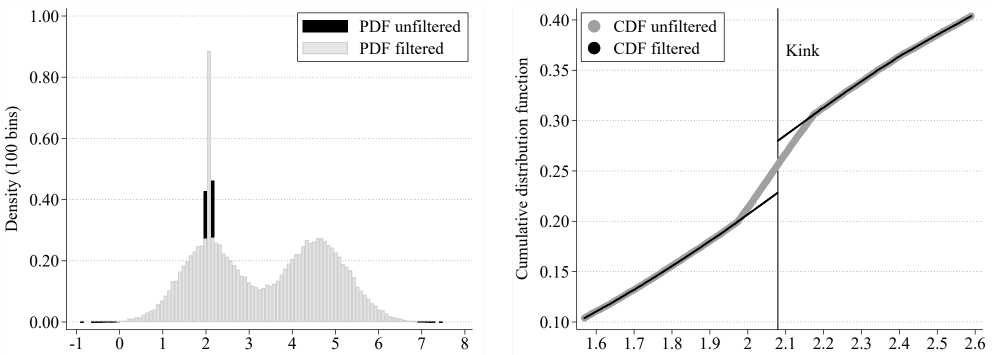
We generate the filtered variable, `yfiltered`, and figure 5 by applying `bunchfilter` to the outcome variable with friction errors using the following command:

```
. bunchfilter yfric [fw=w], kink(2.0794) deltam(0.12) deltap(0.12)
> generate(yfiltered) binwidth(0.084) pctobs(30)
```

(*output omitted*)

We exclude log 12% below the kink, `deltam(0.12)`, and log 12% above the kink, `deltap(0.12)`, because we know these excluded regions will capture the support of the friction errors because the example frictions in (4) never add to or subtract from y_i by more than log 10%.

Without the friction errors, 5.17% of the responses bunch at the kink in the simulated data from (3). Including friction errors lowers this fraction to zero because no observations are exactly at the kink in (4). After removing the frictions with `bunchfilter`, the filtered data have 5.15% of the responses at the kink.



(a) PDFs with frictions and after filtering

(b) CDFs with frictions and after filtering

Figure 5. Effect of `bunchfilter` on data with friction errors

The histogram of $\texttt{yfric}_i$ is shown in figure 5(a). The unfiltered data (black bars) exhibit diffuse bunching around the kink point. The filtered data are saved in the variable `yfiltered` by invoking the option `generate(yfiltered)`. The histogram for the filtered data is depicted in the gray bars with evident reassignment of original dispersed observations around the kink to the kink point exactly. This reassignment can also be seen in the contrast between the filtered and unfiltered CDFs in figure 5(b). Both of these figures are produced by the `bunchfilter` command.

5 Automatic estimation

Despite friction errors and model misspecification, `bunching` provides multiple estimates of the true elasticity by implementing `bunchbounds`, `bunchtobit`, and `bunchfilter` automatically. The user can provide outcome data with friction errors and a misspecified model, and `bunching` can still recover estimates that are close to the true elasticity.

5.1 The bunching command

The command `bunching` is a wrapper function for three other commands: `bunchbounds`, `bunchtobit`, and `bunchfilter`.

5.1.1 Syntax

`bunching` *depvar* [*indepvars*] [*if*] [*in*] [*weight*], `kink(#) s0(#) s1(#)` `m(#)` [`nopic savingbounds(`*filename*[, `replace`]) `binwidth(#)` `grid(`*numlist*`) numiter(#) savingtobit(`*filename*[, `replace`]) `verbose` `deltam(#) deltap(#) generate(`*varname*`) pctobs(#) polorder(#)`]

The syntax and options for `bunching` are inherited from the three commands for which it is a wrapper function, so we do not repeat them here. Entries for the first four options—`kink()`, `s0()`, `s1()`, and `m()`—are required, whereas options inside the square brackets are not required. `bunching` always implements `bunchbounds` and `bunchtobit`. In contrast, `bunchfilter` is only called by `bunching` if all the required entries for `bunchfilter`—namely, `deltam()`, `deltap()`, and `generate()`—are specified.

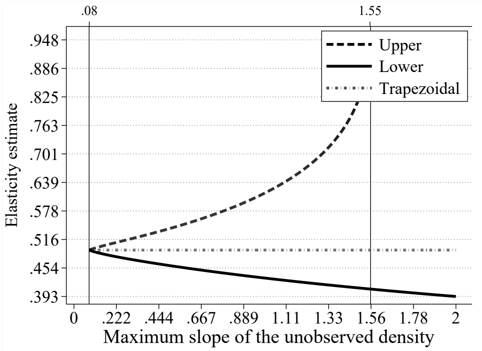
5.2 Example using bunching

The following example uses `bunching` with the outcome data from (4) but omits weights and the covariate x_2 to demonstrate the robustness of this package.

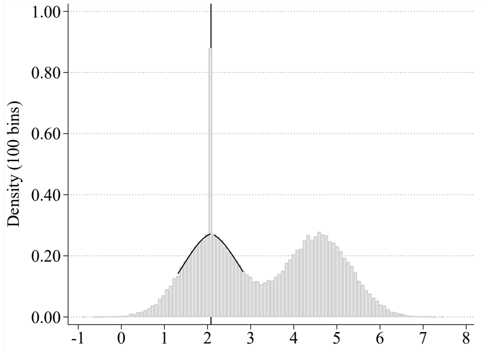
```
. bunching yfric x1 x3, kink(2.0794) s0(0.2624) s1(-0.1054) m(2) binwidth(0.084)
> deltam(0.12) deltap(0.12) generate(ybunching) pctobs(30)
***********************************************
Bunching - Filter
***********************************************
[ 10% 20% 30% 40% 50% 60% 70% 80% 90% 100% ]
***********************************************
Bunching - Bounds
***********************************************

Your choice of M:
2.0000

Sample values of slope magnitude M
 minimum value M in the data (continuous part of the PDF):
  0.0000
 maximum value M in the data (continuous part of the PDF):
  0.3334
 maximum choice of M for finite upper bound:
  1.5530
 minimum choice of M for existence of bounds:
  0.0792

Elasticity Estimates
 Point id., trapezoidal approx.:
  0.4930
 Partial id., M = 2.0000 :
  [0.3926 , +Inf]
 Partial id., M = 1.55 :
  [0.4087 , 0.9480]

***********************************************
Bunching - Tobit
***********************************************

Obtaining initial values for ML optimization.
Truncation window number 1 out of 10, 100% of data.
Truncation window number 2 out of 10, 90% of data.
Truncation window number 3 out of 10, 80% of data.
Truncation window number 4 out of 10, 70% of data.
Truncation window number 5 out of 10, 60% of data.
Truncation window number 6 out of 10, 50% of data.
Truncation window number 7 out of 10, 40% of data.
Truncation window number 8 out of 10, 30% of data.
Truncation window number 9 out of 10, 20% of data.
Truncation window number 10 out of 10, 10% of data.

bunchtobit_out[10,5]
        data %  elasticity     std err  # coll cov        flag
 1         100   .63579175   .00894356           0           0
 2          90   .75808395   .01094832           0           0
 3          80   .73437664   .01066292           0           0
 4          70    .6836851   .00996446           0           0
 5          60   .60786249   .00891428           0           0
 6          50   .52680042   .00787451           0           0
 7          40   .50716643   .00796644           0           0
 8          30   .50457921   .00858105           0           0
 9          20     .501674   .01001586           0           0
10          10    .5076258   .02029615           0           0
```

`bunching` first filters the data using `bunchfilter`. It then implements `bunchbounds` on the filtered outcome using the full sample and maximum slope magnitude as specified.

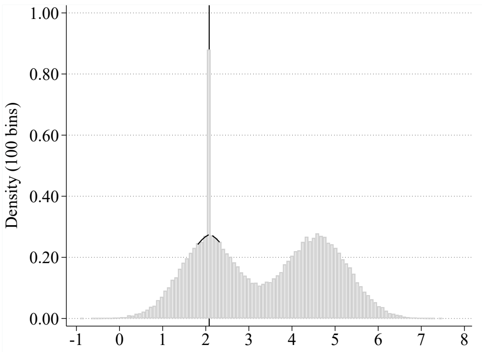
Finally, it uses `bunchtobit` on the filtered outcome with the covariates x_1 and x_3 specified for each of the 10 default truncated subsamples.



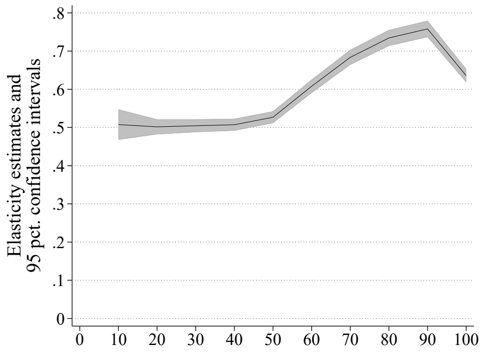
(a) 100% of the data used for estimation



(b) 40% of the data used for estimation



(c) 20% of the data used for estimation



(d) Elasticity by percent of data used

Figure 6. Elasticity estimates with friction errors and model misspecification

Along with numeric output, `bunching` produces the same graphs produced by the `bunchfilter`, `bunchbounds`, and `bunchtobit` commands. Selections from these graphs are shown in figure 6. The output from `bunching` shows that, after we filter the data, the bounds contain the true value of 0.5 [figure 6(a)]. Likewise, estimates from the tobit model in the numeric output show that using a 40% subsample or less recovers the true elasticity of 0.5 despite friction errors and model misspecification. Truncating to 40% of the data provides a good fit, as shown in figure 6(b), and figure 6(c) shows that truncating to 20% also provides a good fit. Figure 6(d) shows that estimates with confidence intervals include the true elasticity of 0.5 for subsamples with 40% of the data and less.

6 Concluding remarks

Our new `bunching` package provides a series of estimation methods that enables researchers to examine the sensitivity of their elasticity estimates to different identification assumptions. The new techniques include bounds based on nonparametric assumptions and a midcensored regression based on semiparametric assumptions and covariates. The nonparametric assumptions are the most flexible of the two approaches and nest the trapezoidal approximation assumption, which was the method used by the original bunching estimator. In cases with multiple kinks, these methods can be applied at each kink separately if the constraint is continuous preceding the kink under study. `bunching` has broad applicability because budget constraints with kinks occur in a variety of fields within economics and other social sciences.

7 Acknowledgments

The views expressed in this article represent the views of the authors and do not indicate concurrence either by the Board of Governors of the Federal Reserve System or by other members of the Federal Reserve System. We gratefully acknowledge the contributions of Andrey Ampilogov. Michael A. Navarrete provided excellent research assistance.

8 Programs and supplemental materials

To install a snapshot of the corresponding software files as they existed at the time of publication of this article, type

```
. net sj 22-3
. net install st0684      (to install program files, if available)
. net get st0684          (to install ancillary files, if available)
```

The `bunching` package and the simulated data can also be downloaded from the Statistical Software Components Archive by typing

```
. ssc install bunching
```

9 References

Allen, E. J., P. M. Dechow, D. G. Pope, and G. Wu. 2017. Reference-dependent preferences: Evidence from marathon runners. *Management Science* 63: 1657–1672. https://doi.org/10.1287/mnsc.2015.2417.

Bertanha, M., A. H. McCallum, and N. Seegert. 2021. Better bunching, nicer notching. Technical Report 2021-002, Board of Governors of the Federal Reserve System. https://doi.org/10.17016/FEDS.2021.002.

Bitler, M., J. Cook, and J. Rothbaum. 2021. Working for your bread: The labor supply effects of SNAP. *American Economic Association Papers and Proceedings* 111: 496–500. https://doi.org/10.1257/pandp.20211094.

Caetano, C. 2015. A test of exogeneity without instrumental variables in models with bunching. *Econometrica* 83: 1581–1600. https://doi.org/10.3982/ECTA11231.

Caetano, C., G. Caetano, and E. R. Nielsen. 2020a. Should children do more enrichment activities? Leveraging bunching to correct for endogeneity. Technical Report 2020-036, Board of Governors of the Federal Reserve System. https://doi.org/10.17016/FEDS.2020.036.

———. 2020b. Correcting for endogeneity in models with bunching. Technical Report 2020-080, Board of Governors of the Federal Reserve System. https://doi.org/10.17016/FEDS.2020.080.

Caetano, G., J. Kinsler, and H. Teng. 2019. Towards causal estimates of children's time allocation on skill development. *Journal of Applied Econometrics* 34: 588–605. https://doi.org/10.1002/jae.2700.

Caetano, G., and V. Maheshri. 2018. Identifying dynamic spillovers of crime with a causal approach to model selection. *Quantitative Economics* 9: 343–394. https://doi.org/10.3982/QE756.

Cattaneo, M. D., M. Jansson, and X. Ma. 2020. Simple local polynomial density estimators. *Journal of the American Statistical Association* 115: 1449–1455. https://doi.org/10.1080/01621459.2019.1635480.

———. 2022. lpdensity: Local polynomial density estimation and inference. *Journal of Statistical Software* 101(2): 1–25. https://doi.org/10.18637/jss.v101.i02.

Cengiz, D., A. Dube, A. Lindner, and B. Zipperer. 2019. The effect of minimum wages on low-wage jobs. *Quarterly Journal of Economics* 134: 1405–1454. https://doi.org/10.1093/qje/qjz014.

Chetty, R., J. N. Friedman, T. Olsen, and L. Pistaferri. 2011. Adjustment costs, firm responses, and micro vs. macro labor supply elasticities: Evidence from Danish tax records. *Quarterly Journal of Economics* 126: 749–804. https://doi.org/10.1093/qje/qjr013.

Chetty, R., J. N. Friedman, and E. Saez. 2013. Using differences in knowledge across neighborhoods to uncover the impacts of the EITC on earnings. *American Economic Review* 103: 2683–2721. http://doi.org/10.1257/aer.103.7.2683.

Dee, T. S., W. Dobbie, B. A. Jacob, and J. Rockoff. 2019. The causes and consequences of test score manipulation: Evidence from the New York regents examinations. *American Economic Journal: Applied Economics* 11: 382–423. https://doi.org/10.1257/app.20170520.

Einav, L., A. Finkelstein, and P. Schrimpf. 2017. Bunching at the kink: Implications for spending responses to health insurance contracts. *Journal of Public Economics* 146: 27–40. https://doi.org/10.1016/j.jpubeco.2016.11.011.

Garicano, L., C. Lelarge, and J. Van Reenan. 2016. Firm size distortions and the productivity distribution: Evidence from France. *American Economic Review* 106: 3439–3479. https://doi.org/10.1257/aer.20130232.

Ghanem, D., S. Shen, and J. Zhang. 2020. A censored maximum likelihood approach to quantifying manipulation in China's air pollution data. *Journal of the Association of Environmental and Resource Economists* 7: 965–1003. https://doi.org/10.1086/709649.

Grossman, D., and U. Khalil. 2020. Neighborhood networks and program participation. *Journal of Health Economics* 70: 102257. https://doi.org/10.1016/j.jhealeco.2019.102257.

Hayashi, F. 2000. *Econometrics*. Princeton, NJ: Princeton University Press.

Ito, K. 2014. Do consumers respond to marginal or average price? Evidence from nonlinear electricity pricing. *American Economic Review* 104: 537–563. https://doi.org/10.1257/aer.104.2.537.

Ito, K., and J. M. Sallee. 2018. The economics of attribute-based regulation: Theory and evidence from fuel economy standards. *Review of Economics and Statistics* 100: 319–336. https://doi.org/10.1162/REST_a_00704.

Jales, H. 2018. Estimating the effects of the minimum wage in a developing country: A density discontinuity design approach. *Journal of Applied Econometrics* 33: 29–51. https://doi.org/10.1002/jae.2586.

Jales, H., and Z. Yu. 2017. Identification and estimation using a density discontinuity approach. In *Advances in Econometrics: Vol. 38—Regression Discontinuity Designs: Theory and Applications*, ed. M. D. Cattaneo and J. C. Escanciano, 29–72. Bingley, U.K.: Emerald. https://doi.org/10.1108/S0731-905320170000038003.

Khalil, U., and N. Yildiz. 2022. A test of the selection on observables assumption using a discontinuously distributed covariate. *Journal of Econometrics* 226: 423–450. https://doi.org/10.1016/j.jeconom.2021.09.018.

Kleven, H. J. 2016. Bunching. *Annual Review of Economics* 8: 435–464. https://doi.org/10.1146/annurev-economics-080315-015234.

Kleven, H. J., and M. Waseem. 2013. Using notches to uncover optimization frictions and structural elasticities: Theory and evidence from Pakistan. *Quarterly Journal of Economics* 128: 669–723. https://doi.org/10.1093/qje/qjt004.

Kopczuk, W., and D. Munroe. 2015. Mansion tax: The effect of transfer taxes on the residential real estate market. *American Economic Journal: Economic Policy* 7: 214–257. https://doi.org/10.1257/pol.20130361.

Olsen, R. J. 1978. Note on the uniqueness of the maximum likelihood estimator for the tobit model. *Econometrica* 46: 1211–1215. https://doi.org/10.2307/1911445.

Saez, E. 2010. Do taxpayers bunch at kink points? *American Economic Journal: Economic Policy* 2: 180–212. https://doi.org/10.1257/pol.2.3.180.

About the authors

Marinho Bertanha is the Gilbert F. Schaefer Assistant Professor of Economics at the University of Notre Dame.

Andrew H. McCallum is a principal economist in the International Finance Division of the Board of Governors of the Federal Reserve System. He also teaches econometrics as an adjunct professor at the McCourt School of Public Policy at Georgetown University.

Alexis M. Payne is an economics PhD student at Stanford University and was a senior research assistant in the International Finance Division of the Board of Governors of the Federal Reserve System. She received her BA in economics from William and Mary in 2019.

Nathan Seegert is an assistant professor of finance in the Eccles School of Business at the University of Utah.