Finance and Economics Discussion Series
Divisions of Research & Statistics and Monetary Affairs
Federal Reserve Board, Washington, D.C.

Bunching estimation of elasticities using Stata

Marinho Bertanha, Andrew H. McCallum, Alexis Payne, and Nathan Seegert

2021-006

Please cite this paper as:
Bertanha, Marinho, Andrew H. McCallum, Alexis Payne, and Nathan Seegert (2021). "Bunching estimation of elasticities using Stata," Finance and Economics Discussion Series 2021-006. Washington: Board of Governors of the Federal Reserve System, https://doi.org/10.17016/FEDS.2021.006.

Bunching estimation of elasticities using Stata

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

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

Alexis Payne
Board of Governors of the
Federal Reserve System
Washington, DC
alexis.m.payne@frb.gov

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

January 15, 2021

Abstract. A continuous distribution of agents that face a piecewise-linear schedule of incentives results in a distribution of responses with mass points located where the slope (kink) or intercept (notch) of the schedule changes. Bunching methods use these mass points to estimate an elasticity parameter, which summarizes agents' responses to incentives. This article introduces the command `bunching`, which implements new non-parametric and semi-parametric identification methods for estimating elasticities developed by Bertanha et al. (2021). These methods rely on weaker assumptions than currently made in the literature and result in meaningfully different estimates of the elasticity in various contexts.

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

1 Introduction

Mass points in the middle of a univariate distribution, often called bunching, have been used to estimate parameters that govern behavioral responses to changes in incentives. For example, bunching has been used to estimate the elasticity of taxable income with respect to the net of tax rate using piecewise linear tax schedules. These methods began with Saez (2010), Chetty et al. (2011), and Kleven and Waseem (2013). Following these influential papers, bunching became a popular method for estimating responses to incentives with cross sectional data.

Bunching estimators are widely applied in settings including fuel economy regulations (Sallee and Slemrod 2012), electricity demand (Ito 2014), real estate taxes (Kopczuk and Munroe 2015), labor regulations (Garicano et al. 2016), prescription drug insurance (Einav et al. 2017), marathon finishing times (Allen et al. 2017), attribute-based regulations (Ito and Sallee 2018), education (Dee et al. 2019; Caetano et al. 2020a), minimum wage (Jales 2018; Cengiz et al. 2019), and air-pollution data manipulation (Ghanem et al. 2019), among others. Variation in the size of the mass point across groups of individuals has also been used as a first stage in a two-stage approach to control for endogeneity (Chetty et al. 2013; Caetano 2015; Grossman and Khalil 2020). Bunching has also been used for causal identification in Khalil and Yildiz (2020), Caetano and Maheshri (2018), Caetano et al. (2019), and Caetano et al. (2020b). Kleven (2016) reviews the many applications and branches of the bunching literature and Jales and Yu (2017) relates bunching to regression discontinuity design (RDD).

This paper introduces a new Stata command, `bunching`, which utilizes assumptions that are weaker than current methods for partial and point identification of the bunching elasticity. The command `bunching` is a wrapper function for three other commands. The first of those commands is `bunchbounds`, which estimates upper and lower bounds on the bunching elasticity using a partial-identification approach. The second is `bunchtobit`, which uses a semi-parametric 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 are developed in Bertanha et al. (2021). That paper introduces a suite of ways to recover elasticities from bunching behavior. Each method differs in the assumptions it makes in order to achieve identification of the bunching elasticity. There is no way to determine which assumption is correct because these are assumptions about an unobserved distribution. Nevertheless, estimates that are stable across many methods indicate that different identifying assumptions do not play a major role in the construction of those estimates. On the contrary, estimates that are sensitive to different assumptions are dependent on the validity of those assumptions. Therefore, we recommend that researchers use the `bunching` package to examine the sensitivity of elasticity estimates across all available methods as a matter of routine.

2 Bunching estimators

A continuous distribution of agents that face a piecewise-linear schedule of incentives results in a distribution of responses with mass points located where the slope of the schedule changes, also called a "kink". For example, a progressive schedule of marginal income tax rates induces a mass of heterogeneous individuals to report the same income at the level where marginal rates increase (Saez 2010).

Agents maximize an iso-elastic quasi-linear utility function which results in a data generating process (DGP) for optimal reported income as follows

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

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

Equation 1 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 $[\underline{n}\,(k,\varepsilon,s_0)\,, \overline{n}\,(k,\varepsilon,s_1)]$, 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 interval that defines bunching according to

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

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

Formally, the data and model comprise five objects: 1) the CDF of the outcome F_y, 2) the kink point k, 3) the slopes of the piecewise-linear constraint s_0 and s_1; 4) the CDF of the latent variable F_{n^*}, and 5) the elasticity ε. Equation 1 is a mapping that takes objects (2)–(5) and maps them into the observed CDF, F_y. The researcher observes objects (1)–(3), but does not observe the last two objects, F_{n^*} and ε.

Intuition for how the original bunching estimators estimate ε is as follows. First, they assume a specific function F_{n^*} over the bunching interval. Second, they invert equation 2 to recover ε using their assumption about F_{n^*}. The methods developed by Bertanha et al. (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 when that kink is not preceded by a notch (a discontinuity in the level of the incentive schedule). The first strategy identifies upper and lower bounds on the elasticity —partially identifies the elasticity —by making a mild shape restriction on the non-parametric family of heterogeneity distributions F_{n^*}. The second strategy point identifies the elasticity using covariates and a semi-parametric restriction on the distribution of heterogeneity.

The first strategy, which is implemented by `bunchbounds`, partially identifies the elasticity by assuming a bound on the slope magnitude of the heterogeneity probability density function (PDF), that is, Lipschitz continuity. Intuition for identification of the elasticity in this setting is as follows. We observe the mass of agents who bunch, which equals the area under the heterogeneity PDF inside an interval. The length of this bunching interval depends on the unknown elasticity. The maximum slope magnitude of the PDF implies upper and lower bounds for all possible PDF values inside the bunching interval that are consistent with the observed bunching mass. This translates into lower and upper bounds, respectively, on the size of the bunching interval, which corresponds to lower and upper bounds on the elasticity. The partial-identification approach has valuable features, among these are that observed bunching always implies a positive elasticity and the original bunching estimator is always inside the partially identified set.

The second strategy, which is implemented by `bunchtobit`, is a semi-parametric method that relies on the fact that bunching can be rewritten as a middle-censored regression model. The likelihood function assumes that the unobserved distribution conditional on covariates is parametric, but we demonstrate that correct specification of the conditional distribution is not necessary for consistency, as long as the unconditional distribution is correctly specified. For example, conditional normality yields a mid-censored Tobit model, which has a globally concave likelihood and is easy to implement. Nevertheless, consistency only requires that the unobserved distribution is a semi-parametric mixture of normals, and conditional normality is not necessary. Truncating the sample around the kink point improves the fit of the model and further weakens these distributional assumptions. The semi-parametric censoring model extends bunching estimators to control for observable heterogeneity for the first time. Observable individual characteristics generally account for substantial variation across agents and leave less heterogeneity unobserved. This fact suggests that identification strategies that utilize covariates should be preferred over identifying assumptions that only restrict the shape of the unobserved distribution without covariates.

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. 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 bunching individuals more than non-bunching individuals, or 3) the variance of the friction error is small. A more general filtering method requires deconvolution theory, which is an active area of research.

3 The bunchbounds command

`bunchbounds` uses bunching to partially-identify the elasticity of income with respect to tax rate. The general syntax of this command is as follows:

Syntax

`bunchbounds` *varname* [*if*] [*in*] [*weight*] , `kink(#) m(#) tax0(#) tax1(#)` [`nopic saving(`*string*`)`]

`varname` must be one dependent variable (ln of income), covariates are optional.

`if|in` like in any other Stata command, to restrict the working sample.

The main command-specific estimation and postestimation options are provided below and are expanded in the `bunchbounds` help file. Entries for the first four options, `kink(#real)`, `m(#real)`, `tax0(#real)`, and `tax1(#real)`, are required whereas options inside the square brackets are not required.

The user enters the name of the income variable (in natural logs), the location of the kink point, the maximum slope magnitude `m` of the heterogeneity PDF, and the marginal tax rates before and after the kink point. The code computes the maximum and minimum values of the elasticity that are consistent with the slope restriction on the PDF and the observed distribution of income. The code gives suggestions of `m` values based on the continuous part of the distribution, as the true value of `m` is unknown. The minimum and maximum values of `m` in the data are constructed from a histogram of the dependent variable that excludes the kink point and use the same default `binwidth` as `bunchtobit`. If that histogram happens to be too undersmoothed, the maximum value of `m` in the data might be too high (and vice-versa).

Options for `bunchbounds`

`kink(#real)` is the location of the kink point where tax rates change.

`m(#real)` is the maximum slope magnitude of the heterogeneity PDF, a strictly positive scalar.

`tax0(#real)` is the marginal income tax rate before the kink point.

`tax1(#real)` is the marginal tax rate after the kink point, which must be strictly bigger than `tax0`.

`* nopic` if you state this option, then no graphs will be displayed. Default state is to have graphs displayed.

`* saving (string [, replace])` gives you the option to save a `*.dta` file with (x,y) coordinates of the graph of the partially-identified set as a function of the slope magnitude of the heterogeneity distribution. Use `saving(filename.dta)` or `saving(filename.dta, replace)` if `filename.dta` already exists in the working directory.

Only `fweight` or `fw` (frequency weights) are allowed; see help file for option `weight` in Stata. Options marked by "*" are not required.

4 The bunchtobit command

`bunchtobit` uses bunching, Tobit regressions and covariates to point identify the elasticity of income with respect to tax rates. The general syntax of the command is as follows:

Syntax

`bunchtobit` *varname* [*if*] [*in*] [*weight*] , `kink(#)` `tax0(#)` `tax1(#)` [`grid(numlist)` `verbose` `numiter(#)` `binwidth(#)` `nopic` `saving(`*string*`)`]

`varname` must be one dependent variable (ln of income), covariates are optional.

`if|in` like in any other Stata command, to restrict the working sample.

The main command-specific estimation and postestimation options are provided below and are expanded in the `bunchtobit` help file. Entries for the first three options, `kink(#real)`, `tax0(#real)`, and `tax1(#real)`, are required whereas options inside the square brackets are not required.

The user enters the name of the income variable (in natural logs), the names of explanatory variables, the location of the kink point, the marginal tax rates before and after the kink point. The code runs a sequence of mid-censored Tobit regressions using different sub-samples of the data. It starts with the entire sample, then it truncates the value of the income variable in shrinking symmetric windows centered at the kink point. The elasticity estimate is plotted as a function of the percentage of data used by the truncation windows. The code also plots the histogram of the income variable along with the best-fit Tobit distribution for each truncation window.

Options for bunchtobit

kink(#real) is the location of the kink point where tax rates change.

tax0(#real) is the marginal income tax rate before the kink point.

tax1(#real) is the marginal tax rate after the kink point, which must be strictly bigger than tax0.

* grid(numlist) grid with integer numbers between 1 and 99. The number of grid points determines the number of symmetric truncation windows around the kink point on which the Tobit regressions are run. The value of the grid points correspond to the percentage of the sample that is selected by each truncation window. The code will always add 100 (full sample) to the grid, so the number of grid points is always one more than the number of grid points provided by the user. The default value for the grid is 10(10)90.

* verbose if provided, this option makes the code display detailed output of Tobit regressions and likelihood iterations. Non-verbose mode is the default.

* numiter(#int) maximum number of iterations for likelihood maximizations of Tobit regressions. Default is 500.

* binwidth(#real) the width of the bins for histograms. Default value is half of what is automatically produced by the command histogram. A strictly positive value.

* nopic if you state this option, then no graphs will be displayed. Default state is to have graphs displayed.

* saving(string [, replace]) gives you the option to save a *.dta file with Tobit estimates for each truncation window. The *.dta file contains eight variables corresponding to the matrices that the code stores in r(). See below for more details. Use saving(filename.dta) or saving(filename.dta, replace) if filename.dta already exists in the working directory.

Only fweight or fw (frequency weights) are allowed; see help file for option weight in Stata. Options marked by "*" are not required.

5 The bunchfilter command

bunchfilter filters out friction errors of data drawn from a mixed continuous-discrete distribution with one mass point plus a continuously distributed friction error. The distribution of the data with error is continuous and its PDF typically exhibits a hump around the location of the mass point. This type of data arises in bunching applications in economics, for example, the distribution of reported income usually has a hump around the kink points where marginal tax rate changes. The general syntax of this command is as follows:

Syntax

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

varname must be one dependent variable (ln of income), covariates are optional.

if|in like in any other Stata command, to restrict the working sample

The main command-specific estimation and postestimation options are provided below and are expanded in the bunchfilter help file. Entries for the first four options, generate(newvar), deltam(#real), deltap(#real), and kink(#real), are required whereas options inside the square brackets are not required.

The user enters the variable to be filtered (e.g., ln of income), the location of the mass point, and length of a window around the mass point that contains the hump (i.e., `kink - deltam, kink + deltap`). The procedure 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 mass point. The fitted regression predicts values of the empirical CDF in the hump window with a jump discontinuity at the mass point. The filtered data equals the inverse of the predicted CDF evaluated at the empirical CDF value of 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 error. A proper deconvolution theory still needs to be developed for a filtering procedure with general validity.

Options for `bunchfilter`

`generate(newvar)` generates the filtered variable with a user-specified name of varname. If this option is used, then options `deltam` and `deltap` must also be specified.

`deltam(#real)` is the lower half-length of the hump window, that is, the distance between the mass point to the lower-bound of the hump window. If this option is used, then options `generate` and `deltap` must also be specified.

`deltap(#real)` is the upper half-length of the hump window, that is, the distance between the mass point to the upper-bound of the hump window. If this option is used, then options `generate` and `deltam` must also be specified.

`kink(#real)` is the location of the mass point.

* `nopic` if you state this option, then no graphs will be displayed. Default state is to have graphs displayed.

* `binwidth(#real)` the width of the bins for histograms. Default value is half of what is automatically produced by the command `histogram`. A strictly positive value.

* `perc_obs(#real)` for better fit, the polynomial regression uses observations in a symmetric window around the kink point that contains `perc_obs` percent of the sample. Default value is 40, (integer, min = 1, max = 99).

* `polorder(#integer)` maximum order of polynomial regression. Default value is 7, min = 2; max = 7.

Only `fweight` or `fw` (frequency weights) are allowed; see help file for option `weight` in Stata. Options marked by "*" are not required.

6 The bunching command

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

Syntax

`bunching` *varname* [*indepvars*] [*if*] [*in*] [*weight*] , `kink(#) tax0(#) tax1(#) m(#)` [`generate(`*varname*`) deltam(#) deltap(#) perc_obs(#) polorder(#) grid(numlist) numiter(#) verbose savingbounds(`*string*`) savingtobit(`*string*`) binwidth(#) nopic`]

`varname` must be one dependent variable (ln of income), covariates are optional.

`if|in` like in any other Stata command, to restrict the working sample.

The main command-specific estimation and postestimation options are provided below and are expanded in the `bunching` help file. Entries for the first four options, `kink(#real)`, `tax0(#real)`, `tax1(#real)`, and `m(#real)` are required whereas options inside the square brackets are not required.

Options for `bunching`

`kink(#real)` is the location of the mass point.

`tax0(#real)` is the marginal income tax rate before the kink point.

`tax1(#real)` is the marginal tax rate after the kink point, which must be strictly bigger than `tax0`.

`m(#real)` is the maximum slope magnitude of the heterogeneity PDF, a strictly positive scalar (option of `bunchbounds`).

* `generate(newvar)` generates the filtered variable with a user-specified name of varname (option of `bunchfilter`). If this option is used, then options `deltam` and `deltap` must also be specified.

* `deltam(#real)` is the lower half-length of the hump window, that is, the distance between the mass point to the lower-bound of the hump window (option of `bunchfilter`). If this option is used, then options `generate` and `deltap` must also be specified.

* `deltap(#real)` is the upper half-length of the hump window, that is, the distance between the mass point to the upper-bound of the hump window (option of `bunchfilter`). If this option is used, then options `generate` and `deltam` must also be specified.

* `perc_obs(#real)` for better fit, the polynomial regression of `bunchfilter` uses observations in a symmetric window around the kink point that contains `perc_obs` percent of the sample. Default value is 40, (integer, min = 1, max = 99).

* `polorder(#integer)` maximum order of polynomial regression of bunchfilter. Default value is 7, min = 2; max = 7.

* `grid(numlist)` grid with integer numbers between 1 and 99 (option of bunchtobit). The number of grid points determines the number of symmetric truncation windows around the kink point, on which the Tobit regressions are run. The value of the grid points correspond to the percentage of the sample that is selected by each truncation window. The code will always add 100 (full sample) to the grid, so the number of grid points is always one more than the number of grid points provided by the user. The default value for the grid is 10(10)90.

* `numiter(#int)` maximum number of iterations for likelihood maximizations of Tobit regressions. Default is 500.

* `verbose` if provided, this option makes the code display detailed output of Tobit regressions and likelihood iterations. Non-verbose mode is the default.

* `savingbounds(string [, replace])` gives you the option to save a `*.dta` file with (x,y) coordinates of the graph of the partially-identified set as a function of the slope magnitude of the heterogeneity distribution (option of `bunchbounds`). Use `saving(filename.dta)` or `saving(filename.dta, replace)` if `filename.dta` already exists in the working directory.

* `savingtobit(string [, replace])` gives you the option to save a `*.dta` file with Tobit estimates for each truncation window. The `*.dta` file contains eight variables corresponding to the matrices that the code stores in `r()`. See below for more details. Use `saving(filename.dta)` or `saving(filename.dta, replace)` if `filename.dta` already exists in the working directory.

* `binwidth(#real)` the width of the bins for histograms of `bunchfilter` and `bunchtobit`. Default value is half of what is automatically produced by the command `histogram`. A strictly positive value.

* `nopic` if you state this option, then no graphs will be displayed. Default state is to have graphs displayed.

Only `fweight` or `fw` (frequency weights) are allowed; see help file for option `weight` in Stata. Options marked by "*" are not required.

7 Examples with simulated data

In this section, we use simulated data to illustrate `bunchbounds`, `bunchtobit`, `bunchfilter`, and `bunching`. First, we demonstrate the commands without friction errors. Second, we show how to remove the friction errors as a precursor to estimating the relevant elasticity. These examples are motivated by the Earned Income Tax Credit that is investigated by Saez (2010) and Bertanha et al. (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 corresponds to log thousands of dollars.

7.1 Simulated data

We consider a data generating process from equation 1 with one kink at $k = \ln(8)$ given by

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

in which the elasticity is $\varepsilon = 0.5$ and the slopes of the budget constraint to the left and right of the kink are $s_0 = \ln(1.3)$ and $s_1 = \ln(0.9)$ (representing tax rates of $t_0 = -0.3$ and $t_1 = 0.1$). 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.

We simulate about one million weighted (100,000 unweighted) observations according to equation 3. Frequency weights are drawn from a standard uniform distribution and demonstrate how to employ weights throughout the `bunching` package. In Figure 1, we graph the histogram of the one million observations in 100 bins. The simulated outcome variable is bimodal due to the covariates and highlight that the unconditional distribution is not normally distributed. The simulated data also exhibits bunching exactly at the kink point. In many empirical applications 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 7.4.

	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

Table 1: Covariates' proprieties

7.2 Estimating elasticity bounds

We begin by estimating the elasticity bounds using the location of the kink, ($\ln(8) = 2.0794$, `k(2.0794)`), tax rates on either side of the kink (`tax0(-0.3)` and `tax1(0.1)`), and a choice of the maximum slope (`m(2)`).

```
. ssc install bunching
checking bunching consistency and verifying not already installed...
installing into c:\ado\plus\...
installation complete.
```

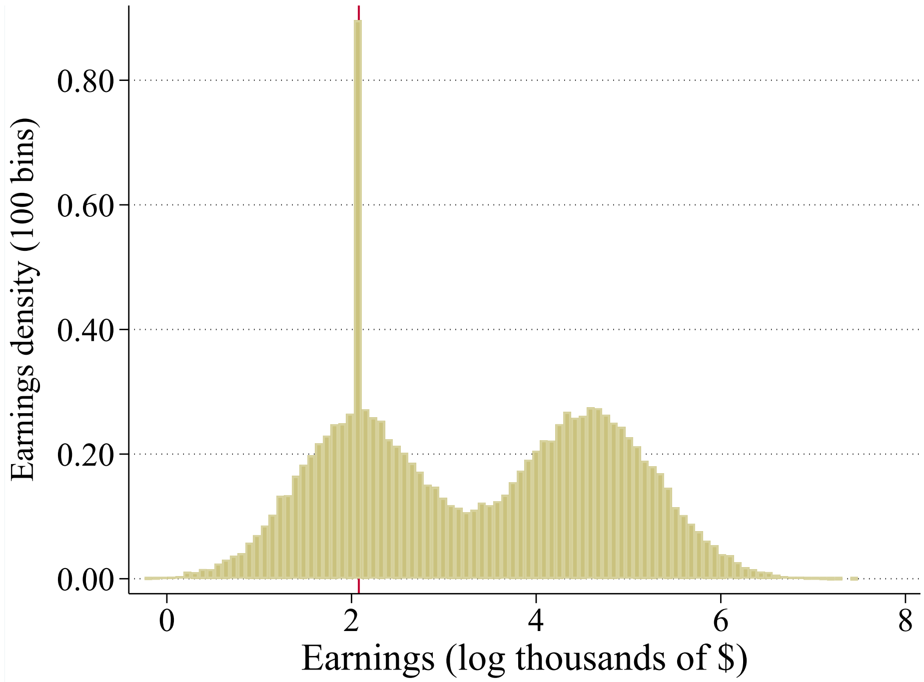


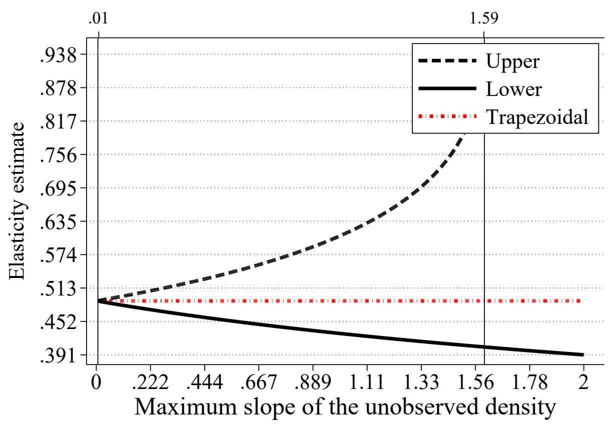
Figure 1: Histogram of simulated data

```
. webuse set "http://fmwww.bc.edu/repec/bocode/b/"
(prefix now "http://fmwww.bc.edu/repec/bocode/b")
. webuse bunching.dta

. bunchbounds y [fweight=w], k(2.0794) tax0(-0.3) tax1(0.1) 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:
  1.5923
 minimum choice of M for existence of bounds:
  0.0065

Elasticity Estimates
 Point id., trapezoidal approx.:
  0.4895
 Partial id., M = 2.0000 :
  [0.3914 , +Inf]
 Partial id., M = 1.59 :
  [0.4056 , 0.9385]
```

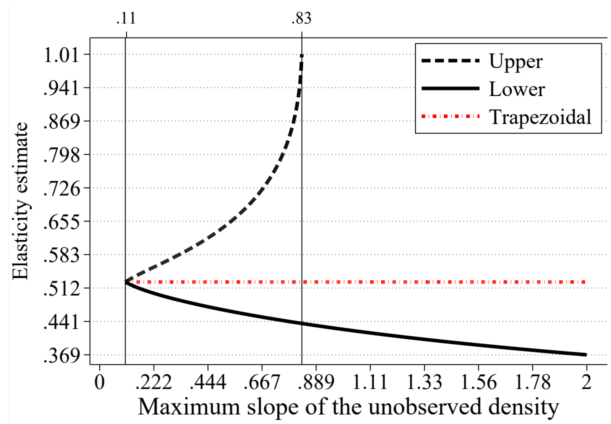
The `bunchbounds` command estimates the bounds for the elasticity using different slope values. First, the output shows that we entered a maximum slope of 2 and the bounds for this slope are $[0.3914, \infty]$. Second, the command also estimates the bounds using the maximum slope for a finite upper bound, when the maximum slope given is greater than that value. In this case, the maximum slope for a finite upper bound is 1.5923, resulting in the bounds $[0.4056, 0.9385]$. 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 probability density function 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 0.4895 is also provided.

The non-parametric 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 2a. 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 2a 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 red line in Figure 2a.

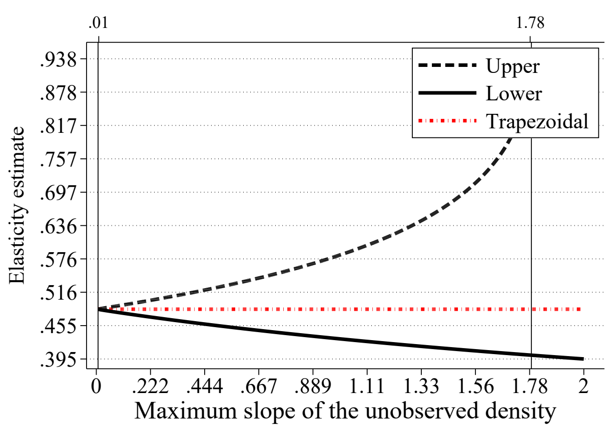
The `bunchbounds` command can also be combined with conditional statements that restricts to subsamples of the data based on values of different covariates. For example, `bunchbounds y if x1==1 & x3==0 [fw=w], k(2.0794) tax0(-0.3) tax1(0.1) m(2)` estimates the bounds when $x_1 = 1$ and $x_3 = 0$. Restricting 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 2b, 2c, and 2d. 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.



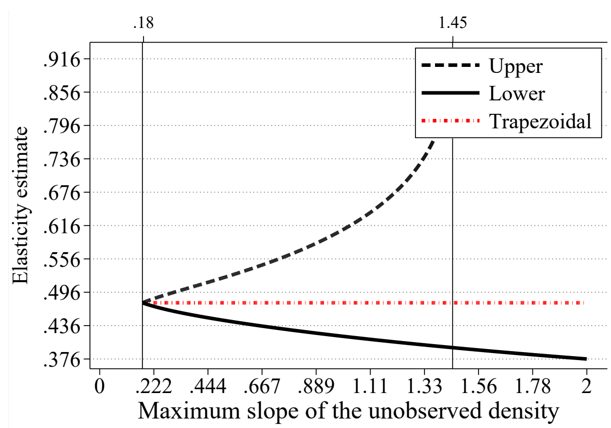
(a) All observations



(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

7.3 Semi-parametric 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 semi-parametric assumptions and does not require the unobserved heterogeneity PDF to be normality distributed (Bertanha et al. 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.

Correctly specified Tobit model

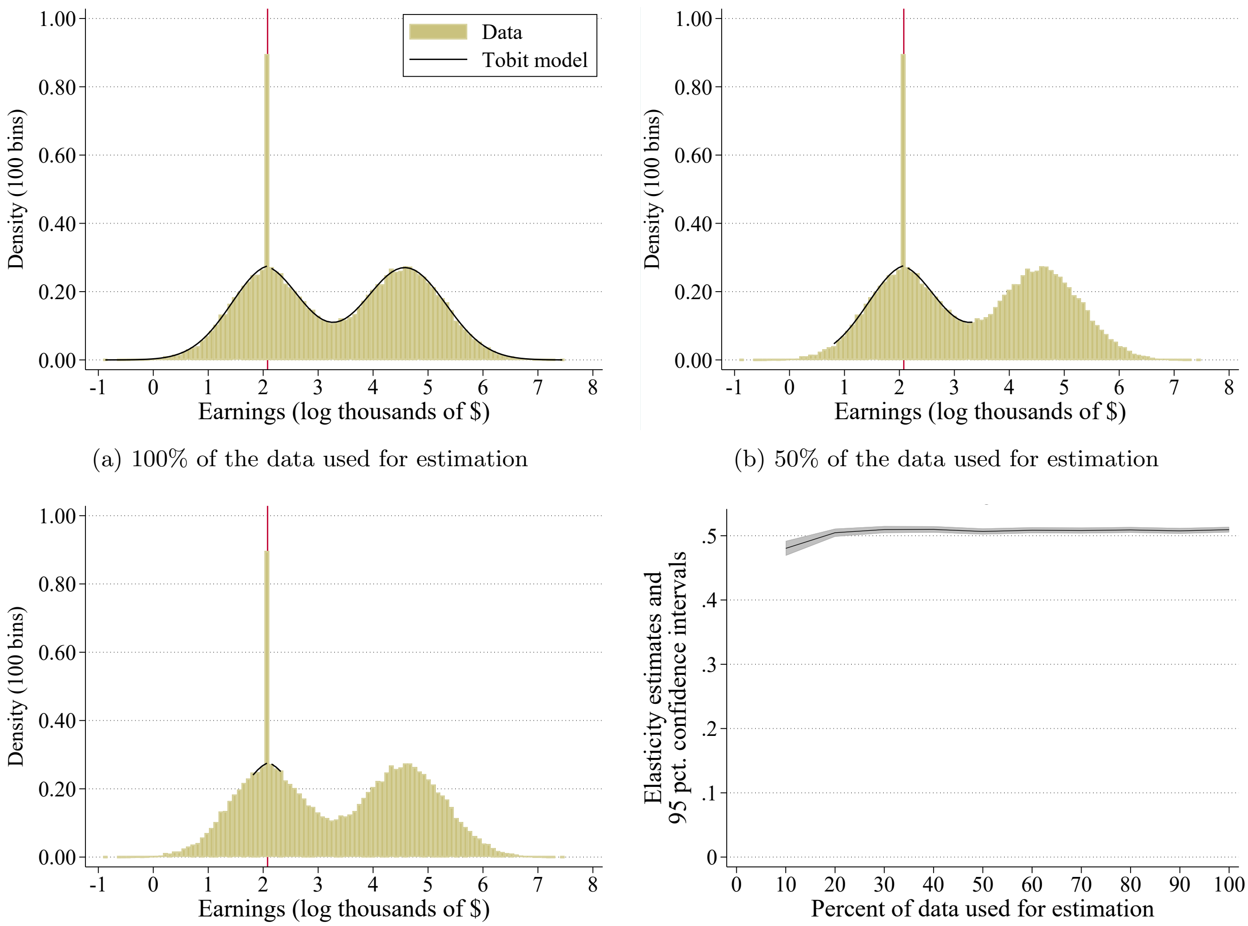
We begin by estimating the correctly specified model using `bunchtobit`.

```
. bunchtobit y x1 x2 x3 [fw=w], k(2.0794) tax0(-0.3) tax1(0.1) 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   .50942038   .00218416           0           0
 2          90   .50757751   .00224641           0           0
 3          80   .50901731   .00227846           0           0
 4          70    .5081248   .00229212           0           0
 5          60    .5085289   .00231752           0           0
 6          50   .50665244   .00236967           0           0
 7          40   .50980096   .00251911           0           0
 8          30   .50959091   .00273072           0           0
 9          20   .50469997   .00317656           0           0
10          10   .48034144   .00585388           0           0
```

The command estimates the elasticity for ten 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, centered around the kink, and includes the observations at the kink. For the data simulated by equation 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 `st err` are small because the simulated data includes one million 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 likelihood did not converge to a maximum in column `flag`.

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 3a, 3b, and 3c display these best-fit graphs for the 100%, 50%, and 20% truncation subsamples, respectively. Each of these panels presents a histogram of y_i (sand colored bars) and the estimate of the correctly specified and truncated Tobit model implied outcome variable (black line). The model is correctly specified and so it fits the data well for all truncated subsamples. Figure 3d plots the estimate (black line) and 95% confidence interval (gray shading) for each truncated subsample corresponding to the `elasticity` column.



(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

The elasticity is the main parameter of interest but the covariate coefficients for the last subsample can be obtained by using the `estimates replay` command. `bunchtobit` always uses the full sample and then the percent of the sample specified in `grid(numlist)`. 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) tax0(-0.3) tax1(0.1) 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   .50942038   .00218416           0           0
2          77   .50853448   .00228193           0           0

. estimates replay

-----------------------------------------------------------------------------
```

```
------------------------------------------------------------------------------
active results
------------------------------------------------------------------------------
------------------------------------------------------------------------------

                                                Number of obs     =    770,197
                                                Wald chi2(0)      =          .
Log pseudolikelihood = -.96353313               Prob > chi2       =          .

 ( 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
             |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
eq_l         |
          x1 |  -.2876602   .0035942   -80.03   0.000    -.2947048   -.2806157
          x2 |   3.541997   .0038313   924.49   0.000     3.534488    3.549507
          x3 |   .5509277   .0036639   150.37   0.000     .5437466    .5581087
       _cons |   3.022084   .0033913   891.14   0.000     3.015438    3.028731
-------------+----------------------------------------------------------------
eq_r         |
          x1 |  -.2876602   .0035942   -80.03   0.000    -.2947048   -.2806157
          x2 |   3.541997   .0038313   924.49   0.000     3.534488    3.549507
          x3 |   .5509277   .0036639   150.37   0.000     .5437466    .5581087
       _cons |   2.757434   .0035783   770.60   0.000     2.750421    2.764448
-------------+----------------------------------------------------------------
lngamma      |
       _cons |   .3472965    .001056   328.87   0.000     .3452267    .3493662
-------------+----------------------------------------------------------------
       sigma |   .7065958   .0014945                      .7051348    .7080598
      cons_l |   2.135392   .0030204                      2.129472    2.141312
      cons_r |   1.948391   .0033686                      1.941789    1.954994
         eps |   .5085345   .0022819                       .504062     .513007
------------------------------------------------------------------------------
```

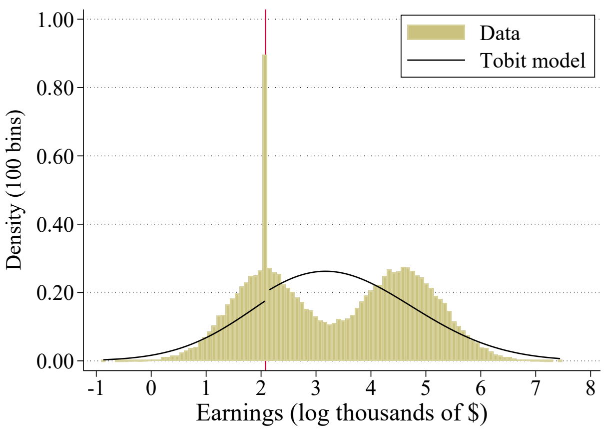
The elasticity reported in column `elasticity` for the 77% subsample is from the estimate `eps` in the `active results` table shown by `estimates replay`. The first equation, `eq_l`, coefficient estimates on x_1, x_2, and x_3 are from the left-hand side of the kink and are the same as the estimates from 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 equation 3. Because the model is correctly specified, the covariate coefficient estimates are consistent and the estimates shown by `estimates replay` are close to the truth for each coefficient.

Incorrectly specified Tobit model

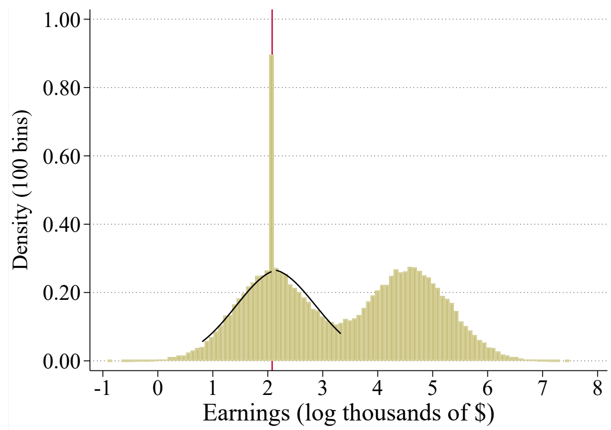
The correctly specified Tobit model from the previous section satisfies the assumption that ν_i is normal and therefore always fits the observed distribution of y_i. A misspecified model that does not have normally distributed residuals will not always fit the distribution of y_i well. However, Bertanha et al. (2021) prove that when the Tobit model best-fit distribution matches the observed distribution of y_i, the elasticity estimated by the Tobit is consistent for the true elasticity, regardless of whether ν_i is normal. This section demonstrates this robustness property using a misspecified model that does not have normal residuals. Specifically, we omit the covariate x_2 and estimate the following model.

```
. bunchtobit y x1 x3 [fw=w], k(2.0794) tax0(-0.3) tax1(0.1) 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.
```

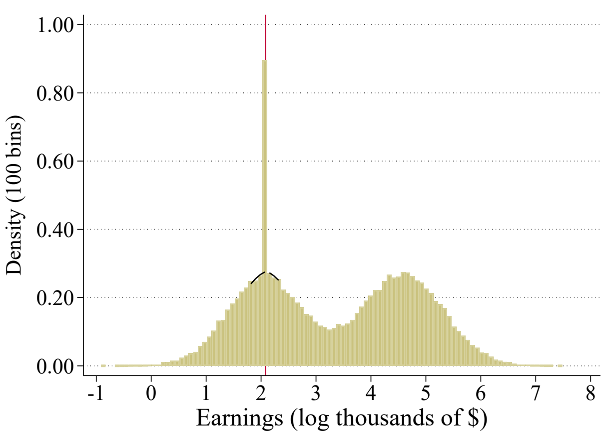
```
Truncation window number 9 out of 10, 20% of data.
Truncation window number 10 out of 10, 10% of data.

bunchtobit_out[10,5]
        data %  elasticity     std err  # coll cov        flag
 1         100   .64273758   .00284316           0           0
 2          90   .76439701   .00347207           0           0
 3          80   .74118047   .00338512           0           0
 4          70   .68976169   .00316223           0           0
 5          60   .61198546   .00282339           0           0
 6          50   .52865914    .0024863           0           0
 7          40    .5126028   .00253684           0           0
 8          30    .5103484   .00273721           0           0
 9          20   .50452522   .00317563           0           0
10          10   .48193233   .00616719           0           0
```



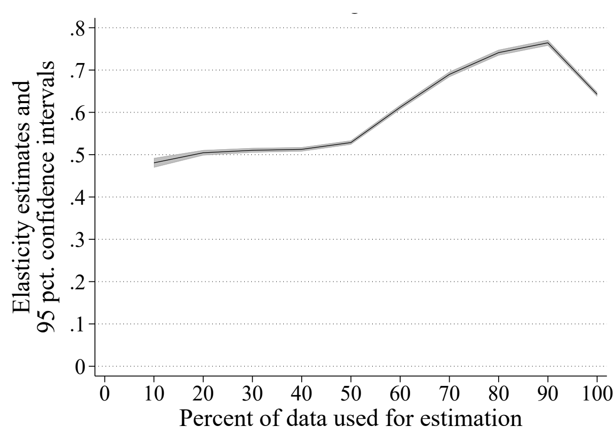
(a) 100% of the data used for estimation



(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

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 4a shows that the misspecified model does not fit well. Using data local to the kink, however, can overcome the effect of omitting x_2. Figure 4b uses 50% of the data and fits much better than the estimate that uses all of the data. Figure 4c uses 20% of the data local to the kink and fits even better than the 50% subsample. The smaller the truncation window around the kink, the easier it is to fit the unconditional distribution of the outcome variable and the stronger is our conviction that the estimate of the elasticity is consistent. Figure 4d shows that for all subsamples that use 50% of the data or less, we recover the true elasticity of 0.5.

7.4 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, $yfric_i$. 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 linearly 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 *iid* with known and bounded support. There is no need for frictions to be mean zero nor for the distribution of the 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 error must equal a polynomial in a known neighborhood of the kink that is bigger than the support of the friction error.

In terms of the simulated data, we generate the outcome variable with friction errors as

$$yfric_i = y_i + e_i \mathbb{I}(y_i = \ln(8)), \tag{4}$$

in which y_i is from equation 3, e_i are *iid* truncated normal from
$f(e_i) = \phi(e_i) / [\Phi(\ln(1.1)) - \Phi(\ln(0.9))]$, $\phi(\cdot)$ is the standard normal PDF, and $\Phi(\cdot)$ is the standard normal CDF. The errors have known and bounded support $[\ln(0.9), \ln(1.1)]$, which ensures frictions never add to or subtracts from y_i by more than log 10 percent. The three conditions needed for `bunchfilter` to consistently estimate y_i are satisfied by equation 4.

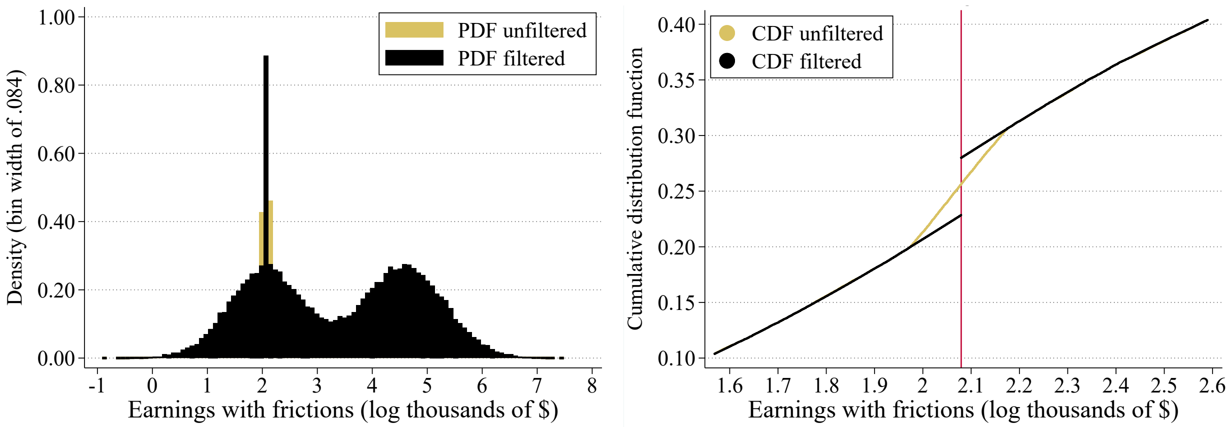
This example generates the filtered variable, `generate(yfiltered)` excluding the interval `deltam(0.12)`, `deltap(0.12)` around the kink,

```
. bunchfilter yfric [fw=w], kink(2.0794) generate(yfiltered) deltam(0.12) deltap(0.12)
> perc_obs(30) binwidth(0.084)
[ 10% 20% 30% 40% 50% 60% 70% 80% 90% 100% ]
```

Without the friction errors, 5.16% of the responses bunch at the kink in the simulated data from equation 3. Including friction errors lowers this fraction to zero because no observation are exactly at the kink in equation 4. After removing the frictions with `bunchfilter`, the filtered data has 5.15% of the responses at the kink. The histogram of $yfric_i$ is shown in Figure 5a. The unfiltered data (sand colored bars) exhibits diffuse bunching around the kink point. The histogram for the filtered data, `generate(yfiltered)`, is depicted in the (black 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 5b. Both of these figures are produced by the `bunchfilter` command.

Automatic filtering, bounds, and semi-parametric estimates

Despite friction errors and model misspecification, `bunching` provides robust 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 the true elasticity. For example, using `bunching` with the outcome data from equation 4 and omitting the covariate x_2 gives the following output



(a) PDFs with frictions and after filtering

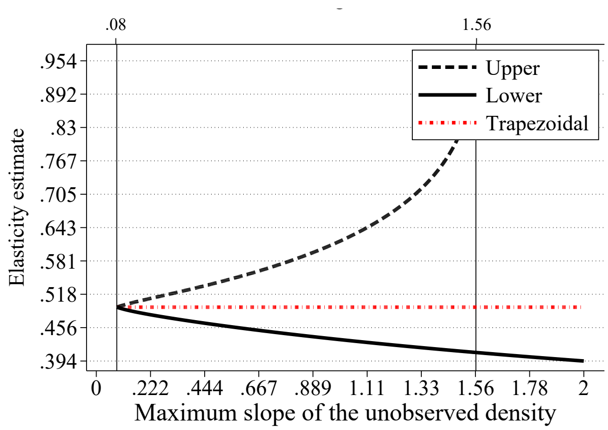
(b) CDFs with frictions and after filtering

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

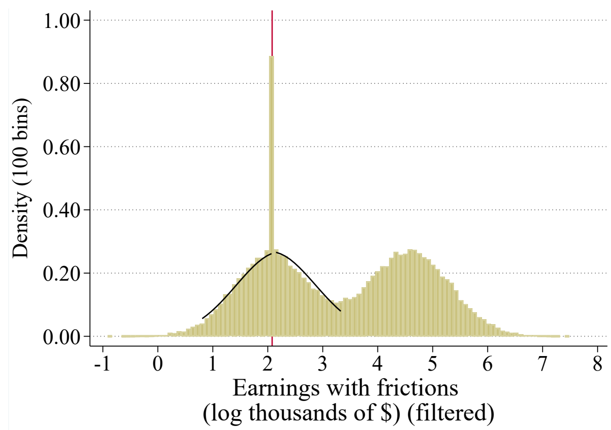
```
. bunching yfric x1 x3 [fw=w], k(2.0794) tax0(-0.3) tax1(0.1) m(2) gen(ybunching)
> deltam(0.12) deltap(0.12) perc_obs(30) binwidth(0.084)

***********************************************
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):
  0.3879
 maximum choice of M for finite upper bound:
  1.5574
 minimum choice of M for existence of bounds:
  0.0840

Elasticity Estimates
 Point id., trapezoidal approx.:
  0.4944
 Partial id., M = 2.0000 :
  [0.3939 , +Inf]
 Partial id., M = 1.56 :
  [0.4099 , 0.9542]
***********************************************
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   .64061846   .00283852           0          0
 2          90   .76174075   .00346511           0          0
 3          80   .73858248    .0033782           0          0
 4          70   .68731806   .00315564           0          0
 5          60   .60975678   .00281718           0          0
 6          50   .52658294   .00248003           0          0
 7          40   .51043494   .00252942           0          0
 8          30   .50822486   .00272937           0          0
 9          20   .50383069   .00317681           0          0
10          10   .50830561   .01367125           0          0
```

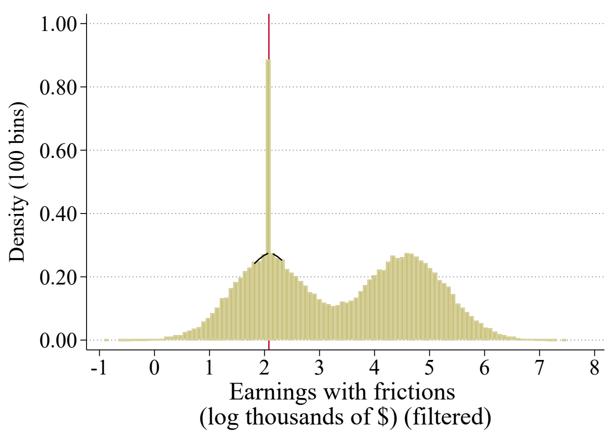
`bunching` first filters the data using `bunchfilter`. It then implements `bunchbounds` on the filtered outcome using the full sample and maximum slope magnitude as specified. Finally, it uses `bunchtobit` on the filtered outcome with the covariates specified, x_1 and x_3, for each of the 10 default truncated subsamples. Along



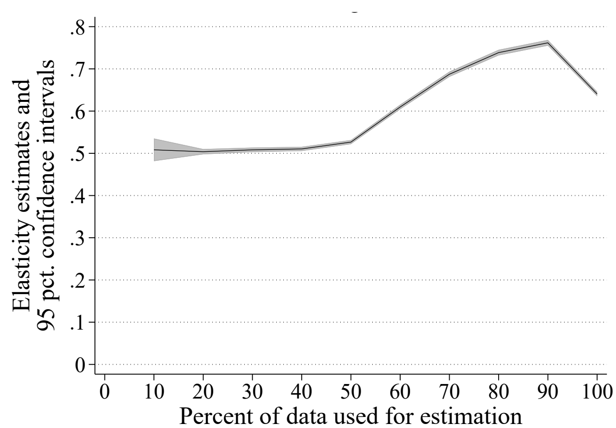
(a) 100% of the data used for estimation



(b) 50% of the data used for estimation



(c) 20% of the data used for estimation



(d) Elasticity by percent of data used

Figure 6: Elasticity estimates with friction errors and model misspecification

with numeric output, `bunching` produces the graphs produced by each of `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 6a). Likewise, estimates from the Tobit model in the numeric output shows that using a 50%

subsample or less of the recovers the true elasticity of 0.5 despite friction errors and model misspecification. Truncating to 50% of the data provides a good fit as shown in Figure 6b and Figure 6c shows that truncating to 20% provides an even better fit. Figure 6d shows that for subsamples with 50% of the data and less provides an estimate that is very close to the truth of 0.5.

8 Concluding remarks

Our new `bunching` package provides a suite of estimation techniques that allow researchers to tailor their estimation of the bunching elasticity to different assumptions. These estimation methods include non-parametric bounds and semi-parametric censored models with covariates. The non-parametric bounds are the least restrictive method and also nest estimators from the previous literature. These techniques have wide applicability, because piecewise-linear budget constraints are common across fields, from public finance and labor economics, to industrial organization and accounting.

9 Acknowledgements

The views expressed in this paper represent the views of the authors and does not indicate concurrence either by the Board of Governors of the Federal Reserve System or other members of the Federal Reserve System. We gratefully acknowledge the contributions of Andrey Ampilogov. Michael A. Navarrete provided excellent research assistance. Bertanha acknowledges financial support received while visiting the Kenneth C. Griffin Department of Economics, University of Chicago.

10 References

Allen, E. J., P. M. Dechow, D. G. Pope, and G. Wu. 2017. Reference-Dependent Preferences: Evidence from Marathon Runners. *Management Science* 63(6): 1657–1672.

Bertanha, M., A. H. McCallum, and N. Seegert. 2021. Better Bunching, Nicer Notching. Finance and Economics Discussion Series 2021-002, Board of Governors of the Federal Reserve System, Washington DC. https://doi.org/10.17016/FEDS.2021.002.

Caetano, C. 2015. A Test of Exogeneity Without Instrumental Variables in Models With Bunching. *Econometrica* 83(4): 1581–1600. https://onlinelibrary.wiley.com/doi/abs/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. https://doi.org/10.17016/FEDS.2020.036.

———. 2020b. Correcting for Endogeneity in Models with Bunching. Finance and Economics Discussion Series 2020-080, Board of Governors of the Federal Reserve System (U.S.). https://ideas.repec.org/p/fip/fedgfe/2020-80.html.

Caetano, G., J. Kinsler, and H. Teng. 2019. Towards causal estimates of children's time allocation on skill development. *Journal of Applied Econometrics* 34(4): 588–605. https://onlinelibrary.wiley.com/doi/abs/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(1): 343–394.

Cengiz, D., A. Dube, A. Lindner, and B. Zipperer. 2019. The Effect of Minimum Wages on Low-wage Jobs. *Quarterly Journal of Economics* 134(3): 1405–1454.

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(2): 749–804.

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(7): 2683–2721. http://www.aeaweb.org/articles?id=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(3): 382–423. http://www.aeaweb.org/articles?id=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.

Garicano, L., C. Lelarge, and J. Van Reenan. 2016. Firm Size Distortions and the Productivity Distribution: Evidence from France. *American Economic Review* 106(11): 3439–3479. https://ideas.repec.org/a/aea/aecrev/v106y2016i11p3439-79.html.

Ghanem, D., S. Shen, and J. Zhang. 2019. A Censored Maximum Likelihood Approach to Quantifying Manipulation in China's Air Pollution Data. Working paper, University of California - Davis.

Grossman, D., and U. Khalil. 2020. Neighborhood networks and program participation. *Journal of Health Economics* 70: 102257. http://www.sciencedirect.com/science/article/pii/S0167629618306830.

Ito, K. 2014. Do Consumers Respond to Marginal or Average Price? Evidence from Nonlinear Electricity Pricing. *American Economic Review* 104(2): 537–563.

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(2): 319–336.

Jales, H. 2018. Estimating the effects of the minimum wage in a developing country: A density discontinuity design approach. *Journal of Applied Econometrics* 33(1): 29–51. https://onlinelibrary.wiley.com/doi/abs/10.1002/jae.2586.

Jales, H., and Z. Yu. 2017. Identification and Estimation Using a Density Discontinuity Approach. In *Regression Discontinuity Designs: Theory and Applications*, ed. M. D. Cattaneo and J. C. Escanciano, 29–72. Vol. 38. Emerald Publishing Limited. https://www.emerald.com/insight/content/doi/10.1108/S0731-905320170000038003/full/html.

Khalil, U., and N. Yildiz. 2020. A Test of the Selection-on-Observables Assumption Using a Discontinuously Distributed Covariate. Technical report, Monash University. https://www.dropbox.com/s/o9bgdua6kcut7a8/selnonobsvbls200715.pdf?dl=0.

Kleven, H. J. 2016. Bunching. *Annual Review of Economics* 8: 435–464.

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(2): 669–723.

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(2): 214–57.

Saez, E. 2010. Do Taxpayers Bunch at Kink Points? *American Economic Journal: Economic Policy* 2(3): 180–212.

Sallee, J. M., and J. Slemrod. 2012. Car Notches: Strategic Automaker Responses to Fuel Economy Policy. *Journal of Public Economics* 96(11): 981–999. https://ideas.repec.org/a/eee/pubeco/v96y2012i11p981-999.html.

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 senior research assistant in the International Finance Division of the Board of Governors of the Federal Reserve System. She received her B.A. in Economics from William & Mary in 2019.

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