Package: bunchr (via r-universe)

September 11, 2024

Type Package

Title Analyze Bunching in a Kink or Notch Setting

Version 1.2.0

Maintainer Itai Trilnick <itai.trilnick@berkeley.edu>

Description View and analyze data where bunching is expected. Estimate counter- factual distributions. For earnings data, estimate the compensated elasticity of earnings w.r.t. the net-of-tax rate.

URL http://github.com/trilnick/bunchr

BugReports http://github.com/trilnick/bunchr/issues

License MIT + file LICENSE

Encoding UTF-8

LazyData TRUE

RoxygenNote 7.2.3

Depends R (>= 3.3.1)

Imports stats, graphics, utils, shiny (>= 0.10.2)

Suggests knitr, rmarkdown, testthat, roxygen2

VignetteBuilder knitr

Repository https://trilnick.r-universe.dev

RemoteUrl https://github.com/trilnick/bunchr

RemoteRef HEAD

RemoteSha aa7305efceab4cabfd8e6f12bee6aa62f7fb6d33

Contents

---

bunch *Bunching Analysis*

Description

Given a kinked or notched budget set, this function gets a vector of earnings and analyzes bunching. The bunchr package has two main useful functions:

Usage

```
bunch(earnings, zstar, t1, t2, Tax = 0, cf_start = NA, cf_end = NA,
  exclude_before = NA, exclude_after = NA, force_after = FALSE,
  binw = 10, poly_size = 7, convergence = 0.01, max_iter = 100,
  correct = TRUE, select = TRUE, draw = TRUE, nboots = 0, seed = NA,
  progress = FALSE, title = "Bunching Visualization",
  varname = "Earnings")
```

Arguments

`earnings`	Vector of earnings, hopefully a very large one.
`zstar`	Place of kink (critical earning point).
`t1`	Marginal tax rate before kink.
`t2`	Marginal tax rate after kink.
`Tax`	"Penalty" tax for crossing zstar.
`cf_start`	Number of bins before the kink bin where counter-factual histogram should start.
`cf_end`	Number of bins after the kink bin where counter-factual histogram should start.
`exclude_before`	Number of excluded bins before the kink bin.
`exclude_after`	Number of excluded bins after the kink bin.
`force_after`	For notch analysis, should bunch be forced to use of the provided *exclude_after* for the end of the bunching, rather than trying to find the bin where the sum of the integral is zero? See details at `notch_estimator` documentation.
`binw`	Bin width.
`poly_size`	Order of polynomial used to calculate counter-factual histogram.
`convergence`	Minimal rate of change of bunching estimate to stop iterations.
`max_iter`	Maximum number of iterations for bunching estimates.

`correct`	Should the counter-factual histogram be corrected to compensate for shifting left because of the notch? See details.
`select`	Should model selection be used to find counter-factual histogram? See details.
`draw`	Should a graph be drawn?
`nboots`	how many bootstraps should be run?
`seed`	specify seed for bootstraps (earnings sampling).
`progress`	Should a progress bar be desplayed?
`title`	Title for Plot output
`varname`	Name for running variable, to be desplayed in the plot

Details

`bunch` checks if the specification has a kink, i.e. if the Tax parameter is greater than zero. If so, it applies `notch_estimator`. Otherwise, it applies `kink_estimator`. Additionally, `bunch` can bootstrap by sampling the earnings vector, returning a vector with the estimated elasticities.

Value

`bunch` returns a list comprising of the parameters returned by `kink_estimator` and `notch_estimator`. If bootstraps were asked for, bootstrapped values are added to the list. Drawing of histograms is suppressed when running the bootsraps.

See Also

`kink_estimator`, `notch_estimator`

Examples

```
# analyzing a kink
ability_vec <- 4000 * rbeta(100000, 2, 5)
earning_vec <- sapply(ability_vec, earning_fun, 0.2, 0, 0.2, 0, 1000)
# bunch_viewer(earning_vec, 1000, 20, 20, 1, 1, binw = 20)
estim <- bunch(earning_vec, 1000, 0, 0.2, Tax = 0, 20, 20, 1, 1,
binw = 20, draw=TRUE, nboots = 0, seed = 16)
estim$e

# analyzing a notch
earning_vec <- sapply(ability_vec, earning_fun, 0.2, 0.2, 0.2, 500, 1000)
bunch_viewer(earning_vec, 1000, 10, 40, 2, 22, binw = 50)
estim <- bunch(earning_vec, 1000, 0.2, 0.2, Tax = 500, 10, 40, 2, 22,
binw = 50, draw = FALSE, nboots = 0, seed = 16)
estim$e
```

bunchApp — *Run bunchApp: an Interactive Bunching Simulation*

Description

bunchApp is an interactive simulator for bunching analysis. It is meant to serve as a tool for understanding bunching analysis in general, and the use of bunchr for data analysis. This app is opened on a separate window.

Usage

```
bunchApp()
```

Details

This function merely runs the app. It accepts no parameters.

See Also

The machinery behind the simulation: bunch

This simulator is also offered online at https://trilnick.shinyapps.io/bunchapp/.

bunchr — *bunchr: A Package for Bunching Analysis*

Description

The bunchr package is meant to help analyze bunching. Given a vector of earnings (or any other numeric vector), it creates a counter-factual count histogram and calculates the compensated elasticity of earnings w.r.t. the net-of-tax rate.

Main functions

bunchr has three main functions:

- bunch is the main function running the actual analysis.
- bunch_viewer serves as an aid to the second by visualizing some of theuser-specified options without running an analysis. Use it to see what the histogram of your earnings vector looks like when setting specific bin size, where the counter-factual analysis should be done, and the bounds of the excluded area. You can also save the histogram bins and counts.
- bunchApp is an interactive simulator. Use it to explore bunching simulation and estimation of earning elasticity.

See Also

bunch, bunch_viewer

bunch_viewer — *Visualizing a histogram and potential excluded areas*

Description

This function is meant to aid find excluded bins and analysis area for a bunching study. It displays a histogram with borders. Optionally, you can get the actual histogram back. This is convenient, as the kink/notch point is set as the center of a bin.

Usage

```
bunch_viewer(earnings, zstar = NA, cf_start = 10, cf_end = 50,
  exclude_before = 2, exclude_after = 20, binw = NA, trimy = TRUE,
  report = FALSE, title = "Count Histogram", varname = "Running Variable")
```

Arguments

earnings	Vector of earnings, hopefully a very large one
zstar	Place of notch/kink (critical earning point)
cf_start	Number of bins before the kink bin where counter-factual histogram should start.
cf_end	Number of bins after the kink bin where counter-factual histogram should start.
exclude_before	Number of excluded bins before the kink bin.
exclude_after	Number of excluded bins after the kink bin.
binw	Bin width.
trimy	Logical. Should the y-axis be trimmed to better show off-bunching histogram?
report	Should the function return the actual histogram?
title	Title for Plot output
varname	Name for running variable, to be desplayed in the plot

Value

A plot, the actual histogram if report is set to TRUE.

See Also

bunch

Examples

```
ability_vec <- 4000 * rbeta(100000, 2, 5)
earning_vec <- sapply(ability_vec, earning_fun, 0.2, 0.1, 0.2, 0, 1000)
bunch_viewer(earning_vec, 1000, 20, 40, 2, 2, 20, trimy = TRUE, report = FALSE)
```

earning_fun *Finding optimal earning under kinked/notched budget set*

Description

For an agent with quasi-linear iso-elastic utility, find the utility maximizing earning level.

Usage

```
earning_fun(n, elas, t1, t2, Tax, zstar)
```

Arguments

n	Ability of person (earnings with zero tax)
elas	elasticity of earnings w.r.t. net-of-tax rate
t1	Tax rate before notch/kink
t2	Tax rate after notch/kink
Tax	height of notch (zero for pure kink)
zstar	place of notch/kink (critical earning point)

Details

`earn_funciton` is intended to simulate earnings of agents under a kink or notch.

Value

Optimal earning level.

See Also

`util_calc`, `bunch`

Examples

```
earning_fun(1200,0.2,0.1,0.3,100,1000)
```

elas_equalizer *Using elasticity to calculating distance between utility at tangency and at notch point*

Description

Given an elasticity, a budget set, and the earnings of the marginal buncher, calculate the utility at notch point and at marginal buncher's earning, and return the absolute difference. Equating these two utilities helps find the elasticity of the marginal buncher. See equations (3) and (4) at Kelven and Waseem (2013)

Usage

```
elas_equalizer(elas, t1, t2, Tax, zstar, delta_zed, binw)
```

Arguments

elas	elasticity of earnings w.r.t. net-of-tax rate
t1	Tax rate before notch/kink
t2	Tax rate after notch/kink
Tax	Height of notch (zero for pure kink)
zstar	Place of notch/kink (critical earning point)
delta_zed	The notch size
binw	Bin width

Value

Absolute value of utility at $zstar + delta_zed$ minus utility at kink/notch point.

References

Kleven, H. and Waseem, Mazhar (2013) *Using notches to uncover optimization frictions and structural elasticities: Theory and evidence from Pakistan*, The Quarterly Journal of Economics 128(2)

Examples

```
elas_equalizer(0.2, 0.1, 0.2, 100, 1000, 200, 20)
```

kink_estimator *Analyzing Bunching at a Kink*

Description

Given a kinked budget set, this function gets a vector of earnings and analyzes bunching. This function could be run independently, but best used through the bunch function.

Usage

```
kink_estimator(earnings, zstar, t1, t2, cf_start = NA, cf_end = NA,
  exclude_before = 2, exclude_after = 2, binw = 10, poly_size = 7,
  convergence = 0.01, max_iter = 100, correct = TRUE, select = TRUE,
  draw = TRUE, title = "Bunching Visualization", varname = "Earnings")
```

Arguments

earnings	Vector of earnings, hopefully a very large one.
zstar	Place of kink (critical earning point).
t1	Marginal tax rate before kink.
t2	Marginal tax rate after kink.
cf_start	Number of bins before the kink bin where counter-factual histogram should start.
cf_end	Number of bins after the kink bin where counter-factual histogram should start.
exclude_before	Number of excluded bins before the kink bin.
exclude_after	Number of excluded bins after the kink bin.
binw	Bin width.
poly_size	Order of polynomial used to calculate counter-factual histogram.
convergence	Minimal rate of change of bunching estimate to stop iterations.
max_iter	Maximum number of iterations for bunching estimates.
correct	Should the counter-factual histogram be corrected to compensate for shifting left because of the notch? See details.
select	Should model selection be used to find counter-factual histogram? See details.
draw	Should a graph be drawn?
title	Title for plot output
varname	Name for running variable, to be desplayed in the plot

Details

A histogram is created from the earnings vector, with the kink point zstar as the center of one of the bins.

Correction of the counter-factual is required, as the kink-induced bunching will shift the whole distribution on the right side of the kink to the left. This option follows Chetty *et al* (2009) in correcting for this.

Model selection works using the `step` function from the stats package. It runs backwards from the full polynomial model, trying to find the best explanatory model using the Akaike information criterion.

Value

`kink_estimator` returns a list of the following variables:

`e` Estimated elasticity

`Bn` The sum of total estimated extra bunching in the excluded bins

`b` The rate of extra bunching in the excluded area, divided by the length of area in \$

`data` A data frame with bin mids, counts, counter-factual counts, and excluded dummy

References

Chetty, R., Friedman, J., Olsen, T., Pistaferri, L. (2009) *Adjustment Costs, Firm Responses, and Micro vs. Macro Labor Supply Elasticities: Evidence from Danish Tax Records*, Quarterly Journal of Economics, 126(2).

See Also

`bunch`, `notch_estimator`

Examples

```
ability_vec <- 4000 * rbeta(100000, 2, 5)
earning_vec <- sapply(ability_vec, earning_fun, 0.2, 0, 0.2, 0, 1000)
# bunch_viewer(earning_vec, 1000, 40, 40, 1, 1, binw = 10)
kink_estimator(earning_vec, 1000, 0, 0.2, 40, 40, 1, 1, binw = 10, draw = FALSE)$e
```

`notch_estimator` *Analyzing Bunching at a Notch*

Description

Given a kinked budget set, this function gets a vector of earnings and analyzes bunching. This function could be run independently, but best used through the bunch function.

Usage

```
notch_estimator(earnings, zstar, t1, t2, Tax = 0, cf_start = NA,
  cf_end = NA, exclude_before = NA, exclude_after = NA,
  force_after = FALSE, binw = 10, poly_size = 7, convergence = 0.01,
  max_iter = 100, select = TRUE, draw = TRUE,
  title = "Bunching Visualization", varname = "Earnings")
```

Arguments

`earnings`	Vector of earnings, hopefully a very large one
`zstar`	Place of kink (critical earning point)
`t1`	Tax rate before kink
`t2`	Tax rate after kink
`Tax`	"Penalty" tax for crossing zstar.
`cf_start`	Number of bins before the kink bin where counter-factual histogram should start.
`cf_end`	Number of bins after the kink bin where counter-factual histogram should start.
`exclude_before`	Number of excluded bins before the kink bin.
`exclude_after`	Number of excluded bins after the kink bin.
`force_after`	Should bunch be forced to use of the provided *exclude_after* for the end of the bunching, rather than trying to find the bin where the sum of the integral is zero? See details.
`binw`	Bin width.
`poly_size`	Order of polynomial used to calculate counter-factual histogram.
`convergence`	Minimal rate of change of bunching estimate to stop iterations.
`max_iter`	Maximum number of iterations for bunching estimates.
`select`	Should model selection be used to find counter-factual histogram? See details.
`draw`	Should a graph be drawn?
`title`	Title for plot output
`varname`	Name for running variable, to be desplayed in the plot

Details

A histogram is created from the earnings vector, with the kink point zstar as the center of one of the bins.

For "unpure" notches, where the marginal tax rate after the notch is different than the one before it, this function disregards the shifting of post-notch distribution to the right, as suggested by Kleven (2016). Asssumption is that the notch effect is much stronger anyway.

Model selection works using the `step` function from the stats package. It runs backwards from the full polynomial model, trying to find the best explanatory model using the Akaike Information Criterion.

By default, `notch_estimator` will try to find the end of the notch, i.e. a histogram bin defining a right-side boundary for a range of an excluded area. An interpolation of the counts inside this range renders an equality between the sum of the "excess" counts, from the left side to the notch point, and the sum of "missing" counts from the notch point to the notch size. `notch_estimator` goes through an iterative process to find a stable right-side boundary, labels it *notch_size* and returns it. However, the user might want to force a visibly detectible end of notch, rather than let `notch_estimator` calculate one. Use this option with caution: the notch size is then used to calculate elasticity. For calculating intensive margin elasticities, excess bunching must all come from other bins. Thus, total sums must be equal and forcing the notch size might not be appropriate. In other settings, e.g. a labor market with extensive margins (entry and exit from labor force), forcing the notch size might be helpful.

Value

`notch_estimator` returns a list of the following variables:

`e` Estimated elasticity

`Bn` The sum of total estimated extra bunching in the area starting at cf_start and through the notch bin (zstar)

`notch_size` Distance between notch bin and bin where the estimated influence of the notch ends, delta_zed

`data` A data frame with bin mids, counts, counter-factual counts, and excluded dummy

References

Kleven, H J (2016). *Bunching*, Annual Review of Economics, 8(1).

See Also

`bunch`, `kink_estimator`

Examples

```
ability_vec <- 4000 * rbeta(100000, 2, 5)
earning_vec <- sapply(ability_vec, earning_fun, 0.2, 0.2, 0.2, 500, 1000)
bunch_viewer(earning_vec, 1000, 15, 30, 2, 21, binw = 50)
notch_estimator(earning_vec, 1000, 0.2, 0.2, 500, 15, 30, 2, 21, binw = 50,
draw = FALSE)$e
```

`util_calc` *Calculating quasi-linear iso-elastic utility*

Description

$$u(z, n, elas, t1, t2, Tax, zstar) =$$

$$z*(1-t1)+[z > zstar]*((z-zstar)*(t2-t1)-Tax)-n/(1+(1/elas))*(z/n)^(1+(1/elas))$$

Usage

```
util_calc(z, n, elas, t1, t2, Tax, zstar)
```

Arguments

z	Earnings
n	Ability of person (earnings with zero tax)
elas	elasticity of earnings w.r.t. net-of-tax rate
t1	Tax rate before notch/kink
t2	Tax rate after notch/kink
Tax	height of notch (zero for pure kink)
zstar	place of notch/kink (critical earning point)

Value

The utility of earning sum z given other parameters.

Examples

```
util_calc(900, 950, 0.2, 0.1, 0.2, 100, 1000)
```

---

util_equalizer — *Calculating distance between utility at tangency and at notch/kink point*

Description

Ability (n) and elasticity (e) determine an agent's earnings and utility. This function determines the tangency point of the agent's utility with the budget line and returns the distance between the utility of earning at that point and the utility of earning at the notch/kink point. This function is mostly used to find the marginal buncher.

Usage

```
util_equalizer(n, elas, t1, t2, Tax, zstar)
```

Arguments

n	Ability of person (earnings with zero tax)
elas	elasticity of earnings w.r.t. net-of-tax rate
t1	Tax rate before notch/kink
t2	Tax rate after notch/kink
Tax	height of notch (zero for pure kink)
zstar	place of notch/kink (critical earning point)

Value

Absolute value of utility at tangency minus utility at kink/notch point.

See Also

util_calc

Examples

```
util_equalizer(1200,0.2,0.1,0.3,100,1000)
```

Index