Package ‘bunching’

October 12, 2022

**Type**  Package

**Title**  Estimate Bunching

**Version**  0.8.6

**Description**  Implementation of the bunching estimator for kinks and notches.

- Allows for flexible estimation of counterfactual (e.g. controlling for round number bunching, accounting for other bunching masses within bunching window, fixing bunching point to be minimum, maximum or median value in its bin, etc.).
- It produces publication-ready plots in the style followed since Chetty et al. (2011) <doi:10.1093/qje/qjr013>, with lots of functionality to set plot options.

**URL**  https://github.com/mavpanos/bunching

**BugReports**  https://github.com/mavpanos/bunching/issues

**License**  MIT + file LICENSE

**Encoding**  UTF-8

**LazyData**  true

**Depends**  BB (>= 2014.10-1), dplyr (>= 0.8.1), ggplot2 (>= 3.1.1), R (>= 2.10), stats, tidyr (>= 0.8.2)

**RoxygenNote**  7.2.1

**Suggests**  knitr, rmarkdown, testthat

**VignetteBuilder**  knitr, rmarkdown

**NeedsCompilation**  no

**Author**  Panos Mavrokonstantis [aut, cre]

**Maintainer**  Panos Mavrokonstantis <mavpanos@gmail.com>

**Repository**  CRAN

**Date/Publication**  2022-08-24 16:52:34 UTC
bin_data

**Description**

Create data frame of binned counts

**Usage**

```r
bin_data(z_vector, binv = "median", zstar, binwidth, bins_l, bins_r)
```

**Arguments**

- `z_vector` a numeric vector of (unbinned) data.
- `binv` a string setting location of zstar within its bin ("min", "max" or "median" value). Default is median.
- `zstar` a numeric value for the the bunching point.
- `binwidth` a numeric value for the width of each bin.
- `bins_l` number of bins to left of zstar to use in analysis.
- `bins_r` number of bins to right of zstar to use in analysis.

**Value**

`bin_data` returns a data frame with bins and corresponding frequencies (counts).

**See Also**

`bunchit`
bunching

Examples

data(bunching_data)
binned_data <- bin_data(z_vector = bunching_data$kink, zstar = 10000,
binwidth = 50, bins_l = 20, bins_r = 20)
head(binned_data)

bunching: Analyze bunching at a kink or notch

Description

The bunching package implements the bunching estimator in settings with kinks or notches. Given a numeric vector, it allows the user to estimate bunching at a particular location in the vector’s distribution, and returns a rich set of results. Important features include functionality for controlling for (different levels of) round numbers, controlling for other bunching points in the bunching bandwidth, and splitting bins using the bunching point as the minimum, median or maximum in its bin for robustness analysis. It estimates standard errors using residual-based bootstrapping, and returns estimated elasticities using both reduced-form and parametric specifications. Besides estimation, it produces bunching plots in the style of Chetty et al. (2011) with lots of functionality for editing the plot’s appearance.

Main functions

bunching has two main functions:

- bunchit is the main function that runs all the analysis.
- plot_hist is a tool for exploratory visualization prior to estimating bunching. It can be used to decide how to choose the appropriate binwidth, bandwidth, the number around the bunching point to include in the bunching region, the polynomial order, whether to control for round numbers and other fixed effects in the bandwidth.

See Also

bunchit, plot_hist

bunching_data Simulated data for bunching examples.

Description

A dataset containing two simulated vectors of about 27,500 observations.

Usage

bunching_data
Format

A data frame with 27510 rows and 2 variables:

- **kink_vector** simulated earnings vector, suitable for examples of bunching at kinks.
- **notch_vector** simulated earnings vector, suitable for examples of bunching at notches.

See Also

bunching, bunchit

---

**bunchit**

**Bunching Estimator**

**Description**

Implement the bunching estimator in a kink or notch setting.

**Usage**

```r
bunchit(
  z_vector,
  binv = "median",
  zstar,
  binwidth,
  bins_l,
  bins_r,
  poly = 9,
  bins_excl_l = 0,
  bins_excl_r = 0,
  extra_fe = NA,
  rn = NA,
  n_boot = 100,
  correct = TRUE,
  correct_above_zu = FALSE,
  correct_iter_max = 200,
  t0,
  t1,
  notch = FALSE,
  force_notch = FALSE,
  e_parametric = FALSE,
  e_parametric_lb = 1e-04,
  e_parametric_ub = 3,
  seed = NA,
  p_title = "",
  p_xtitle = deparse(substitute(z_vector)),
  p_ytitle = "Count",
  p_title_size = 11,
)```

```
p_axis_title_size = 10,
p_axis_val_size = 8.5,
p_miny = 0,
p_maxy = NA,
p_ybreaks = NA,
p_freq_color = "black",
p_cf_color = "maroon",
p_zstar_color = "red",
p_grid_major_y_color = "lightgrey",
p_freq_size = 0.5,
p_freq_msize = 1,
p_cf_size = 0.5,
p_zstar_size = 0.5,
p_b = FALSE,
p_e = FALSE,
p_b_e_xpos = NA,
p_b_e_ypos = NA,
p_b_e_size = 3,
p_domregion_color = "blue",
p_domregion_ltype = "longdash"
)

Arguments

z_vector  a numeric vector of (unbinned) data.
binv  a string setting location of zstar within its bin ("min", "max" or "median" value). Default is median.
zstar  a numeric value for the the bunching point.
binwidth  a numeric value for the width of each bin.
bins_l  number of bins to left of zstar to use in analysis.
bins_r  number of bins to right of zstar to use in analysis.
poly  a numeric value for the order of polynomial for counterfactual fit. Default is 9.
bins_excl_l  number of bins to left of zstar to include in bunching region. Default is 0.
bins_excl_r  number of bins to right of zstar to include in bunching region. Default is 0.
extra_fe  a numeric vector of bin values to control for using fixed effects. Default includes no controls.
rn  a numeric vector of (up to 2) round numbers to control for. Default includes no controls.
n_boot  number of bootstrapped iterations. Default is 100.
correct  implements correction for integration constraint. Default is TRUE.
correct_above_zu  if integration constraint correction is implemented, should counterfactual be shifted only above zu (upper bound of exclusion region)? Default is FALSE (i.e. shift from above zstar).
correct_iter_max
maximum iterations for integration constraint correction. Default is 200.

t0
numeric value setting the marginal (average) tax rate below zstar in a kink (notch) setting.

t1
numeric value setting the marginal (average) tax rate above zstar in a kink (notch) setting.

notch
whether analysis is for a kink or notch. Default is FALSE (kink).

force_notch
whether to enforce user’s choice of zu (upper limit of bunching region) in a notch setting. Default is FALSE (zu set by setting bunching equal to missing mass).

e_parametric
whether to estimate elasticity using parametric specification (quasi-linear and iso-elastic utility function). Default is FALSE (which estimates reduced-form approximation).

e_parametric_lb
lower bound for elasticity estimate’s solution using parametric specification in notch setting. Default is 1e-04.

e_parametric_ub
upper bound for elasticity estimate’s solution using parametric specification in notch setting. Default is 3.

seed
a numeric value for bootstrap seed (random re-sampling of residuals). Default is NA.

p_title
plot’s title. Default is empty.

p_xtitle
plot’s x_axis label. Default is the name of z_vector.

p_ytitle
plot’s y_axis label. Default is "Count".

p_title_size
size of plot’s title. Default is 11.

p_axis_title_size
size of plot’s axes’ title labels. Default is 10.

p_axis_val_size
size of plot’s axes’ numeric labels. Default is 8.5.

p_miny
plot’s minimum y_axis value. Default is 0.

p_maxy
plot’s maximum y_axis value. Default is optimized internally.

p_ybreaks
a numeric vector of y-axis values at which to add horizontal line markers in plot. Default is optimized internally.

p_freq_color
plot’s frequency line color. Default is "black".

p_cf_color
plot’s counterfactual line color. Default is "maroon".

p_zstar_color
plot’s bunching region marker lines color. Default is "red".

p_grid_major_y_color
plot’s y-axis major grid line color. Default is "lightgrey".

p_freq_size
plot’s frequency line thickness. Default is 0.5.

p_freq_msize
plot’s frequency line marker size. Default is 1.

p_cf_size
plot’s counterfactual line thickness. Default is 0.5.
bunchit

- `p_zstar_size`: plot's bunching region marker line thickness. Default is 0.5.
- `p_b`: whether plot should also include the bunching estimate. Default is FALSE.
- `p_e`: whether plot should also include the elasticity estimate. Only shown if `p_b` is TRUE. Default is FALSE.
- `p_b_e_xpos`: plot's x-axis coordinate of bunching/elasticity estimate. Default is set internally.
- `p_b_e_ypos`: plot's y-axis coordinate of bunching/elasticity estimate. Default is set internally.
- `p_b_e_size`: size of plot's printed bunching/elasticity estimate. Default is 3.
- `p_domregion_color`: plot's dominated region marker line color in notch setting. Default is "blue".
- `p_domregion_ltype`: line type for the vertical line type marking the dominated region (zD) in the plot for notch settings. Default is "longdash".

Details

bunchit implements the bunching estimator in both kink and notch settings. It bins a given numeric vector, fits a counterfactual density, and estimates the bunching mass (normalized and not), the elasticity and the location of the marginal buncher. In the case of notches, it also finds the dominated region and estimates the fraction of observations located in it.

Value

bunchit returns a list of results, both for visualizing and for further analysis of the data underlying the estimates. These include:

- `plot`: The bunching plot.
- `data`: The binned data used for estimation.
- `cf`: The estimated counterfactuals.
- `B`: The estimated excess mass (not normalized).
- `B_vector`: The vector of bootstrapped B's.
- `B_sd`: The standard deviation of B_vector.
- `b`: The estimated excess mass (normalized).
- `b_vector`: The vector of bootstrapped b's.
- `b_sd`: The standard deviation of b_vector.
- `e`: The estimated elasticity.
- `e_vector`: The vector of bootstrapped elasticities (e).
- `e_sd`: The standard deviation of e_vector.
- `alpha`: The estimated fraction of bunchers in dominated region (notch case).
- `alpha_vector`: The vector of bootstrapped alphas.
- `alpha_sd`: The standard deviation of alpha_vector.
- `model_fit`: The model fit on the actual (i.e. not bootstrapped) data.
- `zD`: The value demarcating the dominated region (notch case).
zD_bin  The bin above \texttt{zstar} demarcating the dominated region (notch case).

\texttt{zU}\_bin  The location of \texttt{zU} (upper range of excluded region) as estimated from notch setting by setting \texttt{force\_notch = FALSE}.

\texttt{marginal\_buncher}  The location (\texttt{z value}) of the marginal buncher.

\texttt{marginal\_buncher\_vector}  The vector of bootstrapped \texttt{marginal\_buncher} values.

\texttt{marginal\_buncher\_sd}  The standard deviation of \texttt{marginal\_buncher\_vector}.

\textbf{See Also}

\texttt{plot_hist}

\textbf{Examples}

## Not run:
# First, load the example data
data(bunching\_data)

# Example 1: Kink with integration constraint correction
kink1 <- bunchit(z\_vector = bunching\_data\$kink, zstar = 10000, binwidth = 50,  
bins\_l = 20, bins\_r = 20, poly = 4, t0 = 0, t1 = .2,  
p\_b = TRUE, seed = 1)
kink1\$plot
kink1\$b
kink1\$b\_sd

# Example 2: Kink with diffuse bunching
bpoint <- 10000; binwidth <- 50
kink2\_vector <- c(bunching\_data\$kink\_vector,  
rep(bpoint - binwidth,80), rep(bpoint - 2*binwidth,190),  
rep(bpoint + binwidth,80), rep(bpoint + 2*binwidth,80))
kink2 <- bunchit(z\_vector = kink2\_vector, zstar = 10000, binwidth = 50,  
bins\_l = 20, bins\_r = 20, poly = 4, t0 = 0, t1 = .2,  
bins\_excl\_l = 2, bins\_excl\_r = 2, correct = FALSE,  
p\_b = TRUE, seed = 1)
kink2\$plot

# Example 3: Kink with further bunching at other level in bandwidth
kink3\_vector <- c(bunching\_data\$kink\_vector, rep(10200,540))
kink3 <- bunchit(kink3\_vector, zstar = 10000, binwidth = 50,  
bins\_l = 40, bins\_r = 40, poly = 6, t0 = 0, t1 = .2,  
correct = FALSE, p\_b = TRUE, extra\_fe = 10200, seed = 1)
kink3\$plot

# Example 4: Kink with round number bunching
rn1 <- 500; rn2 <- 250
bpoint <- 10000
kink4\_vector <- c(bunching\_data\$kink\_vector,  
rep(bpoint + rn1, 270),  
rep(bpoint + rn2, 270),  
rep(bpoint - rn1, 270),  
rep(bpoint - rn2, 270))
kink4 <- bunchit(kink4\_vector, zstar = 10000, binwidth = 50,  
bins\_l = 20, bins\_r = 20, poly = 4, t0 = 0, t1 = .2,  
correct = FALSE, p\_b = TRUE, seed = 1)
kink4\$plot
rep(bpoint + 2*rn1, 230),
rep(bpoint - rn1, 260),
rep(bpoint - 2*rn1, 275),
rep(bpoint + rn2, 130),
rep(bpoint + 3*rn2, 140),
rep(bpoint - rn2, 120),
rep(bpoint - 3*rn2, 135))

kink4 <- bunchit(z_vector = kink4_vector, zstar = bpoint, binwidth = 50,
bins_l = 20, bins_r = 20, poly = 6, t0 = 0, t1 = .2,
correct = FALSE, p_b = TRUE, p_e = TRUE, p_freq_msize = 1.5,
p_b_e_ypos = 880, nn = c(250, 500), seed = 1)

kink4$plot

# Example 5: Notch
notch <- bunchit(z_vector = bunching_data(notch_vector, zstar = 10000, binwidth = 50,
bins_l = 40, bins_r = 40, poly = 5, t0 = 0.18, t1 = .25,
correct = FALSE, notch = TRUE, p_b = TRUE, p_b_e_xpos = 8900,
n_boot = 0)

notch$plot

## End(Not run)

domregion

**Dominated Region**

**Description**

Estimate z (the value of z_vector) that demarcates the upper bound of the dominated region (in notch settings only).

**Usage**

`domregion(zstar, t0, t1, binwidth)`

**Arguments**

- `zstar`: a numeric value for the the bunching point.
- `t0`: numeric value setting the marginal (average) tax rate below zstar in a kink (notch) setting.
- `t1`: numeric value setting the marginal (average) tax rate above zstar in a kink (notch) setting.
- `binwidth`: a numeric value for the width of each bin.

**Value**

domregion returns a list with the following objects related to the dominated region (in notch settings only):

- **zD**: The level of z that demarcates the upper bound of the dominated region.
- **zD_bin**: The value of the bin which zD falls in.
**do_bootstrap**

**Description**

Estimate bunching on bootstrapped samples, using residual-based bootstrapping with replacement.

**Usage**

```r
do_bootstrap(
  zstar,  
  binwidth, 
  firstpass_prep, 
  residuals, 
  n_boot = 100,  
  correct = TRUE, 
  correct_iter_max = 200,  
  notch = FALSE, 
  zD_bin = NA,  
  seed = NA
)
```

**Arguments**

- `zstar` a numeric value for the the bunching point.
- `binwidth` a numeric value for the width of each bin.
- `firstpass_prep` (binned) data that includes all variables necessary for fitting the model.
- `residuals` residuals from (first pass) fitted bunching model.
- `n_boot` number of bootstrapped iterations. Default is 100.
- `correct` implements correction for integration constraint. Default is TRUE.
- `correct_iter_max` maximum iterations for integration constraint correction. Default is 200.
- `notch` whether analysis is for a kink or notch. Default is FALSE (kink).
- `zD_bin` the bin marking the upper end of the dominated region (notch case).
- `seed` a numeric value for bootstrap seed (random re-sampling of residuals). Default is NA.

**Examples**

```r
domregion(zstar = 10000, t0 = 0, t1 = 0.2, binwidth = 50)
```

**See Also**

`bunchit`
do_correction returns a list with the following bootstrapped estimates:

- **b_vector**: A vector with the bootstrapped normalized excess mass estimates.
- **b_sd**: The standard deviation of the bootstrapped b_vector.
- **B_vector**: A vector with the bootstrapped excess mass estimates (not normalized).
- **B_sd**: The standard deviation of the bootstrapped B_vector.
- **marginal_buncher_vector**: A vector with the bootstrapped estimates of the location (z value) of the marginal buncher.
- **marginal_buncher_sd**: The standard deviation of the bootstrapped marginal_buncher_vector.
- **alpha_vector**: A vector with the bootstrapped estimates of the fraction of bunchers in the dominated region (only in notch case).
- **alpha_vector_sd**: The standard deviation of the bootstrapped alpha_vector.

**See Also**

bunchit, prep_data_for_fit

**Examples**

data(bunching_data)
binned_data <- bin_data(z_vector = bunching_data$kink, zstar = 10000,
    binwidth = 50, bins_l = 20, bins_r = 20)
prepped_data <- prep_data_for_fit(binned_data, zstar = 10000, binwidth = 50,
    bins_l = 20, bins_r = 20, poly = 4)
firstpass <- fit_bunching(prepped_data$data_binned,
    prepped_data$model_formula,
    binwidth = 50)
residuals_for_boot <- fit_bunching(prepped_data$data_binned,
    prepped_data$model_formula,
    binwidth = 50)$residuals
boot_results <- do_bootstrap(zstar = 10000, binwidth = 50,
    firstpass_prep = prepped_data,
    residuals = residuals_for_boot,
    seed = 1)

boot_results$b_sd

---

**Integration Constraint Correction**

**Description**

Implements the correction for the integration constraint.
do_correction

Usage

do_correction(
  zstar,
  binwidth,
  data_prepped,
  firstpass_results,
  correct_iter_max = 200,
  notch = FALSE,
  zD_bin = NA
)

Arguments

zstar         a numeric value for the the bunching point.
binwidth      a numeric value for the width of each bin.
data_prepped  (binned) data that includes all variables necessary for fitting the model.
firstpass_results  initial bunching estimates without correction.
correct_iter_max  maximum iterations for integration constraint correction. Default is 200.
notch         whether analysis is for a kink or notch. Default is FALSE (kink).
zD_bin        the bin marking the upper end of the dominated region (notch case).

Value

do_correction returns a list with the data and estimates after correcting for the integration constraint, as follows:

data         The dataset with the corrected counterfactual.
coefficients The coefficients of the model fit on the corrected data.
b_corrected  The normalized excess mass, corrected for the integration constraint.
B_corrected  The excess mass (not normalized), corrected for the integration constraint.
c0_corrected The counterfactual at zstar, corrected for the integration constraint.
marginal_buncher_corrected The location (z value) of the marginal buncher, corrected for the integration constraint.
alpha_corrected The estimated fraction of bunchers in the dominated region, corrected for the integration constraint (only in notch case).

See Also

bunchit, fit_bunching
Examples

```r
data(bunching_data)
binned_data <- bin_data(z_vector = bunching_data$kink, zstar = 10000,
                         binwidth = 50, bins_l = 20, bins_r = 20)
prepped_data <- prep_data_for_fit(binned_data, zstar = 10000, binwidth = 50,
                                  bins_l = 20, bins_r = 20, poly = 4)
firstpass <- fit_bunching(prepped_data$data_binned,
                          prepped_data$model_formula,
                          binwidth = 50)
corrected <- do_correction(zstar = 10000, binwidth = 50,
                           data_prepped = prepped_data$data_binned,
                           firstpass_results = firstpass)
paste0("Without correction, b = ", firstpass$b_estimate)
paste0("With correction, b = ", round(corrected$b_corrected,3))
```

---

elasticity

### Description

Estimate elasticity from single normalized bunching observation.

### Usage

```r
elasticity(
  beta,  # normalized excess mass.
  binwidth,  # a numeric value for the width of each bin.
  zstar,  # a numeric value for the the bunching point.
  t0,  # numeric value setting the marginal (average) tax rate below zstar in a kink (notch) setting.
  t1,  # numeric value setting the marginal (average) tax rate above zstar in a kink (notch) setting.
  notch = FALSE,  # whether analysis is for a kink or notch. Default is FALSE (kink).
  e_parametric = FALSE,  # whether to use a parametric approach.
  e_parametric_lb = 1e-04,  # lower bound for e_parametric.
  e_parametric_ub = 3)  # upper bound for e_parametric.
```
fit_bunching

Value

elasticity returns the estimated elasticity. By default, this is based on the reduced-form approximation. To use the parametric equivalent, set e_parametric to TRUE.

See Also

bunchit

Examples

elasticity(beta = 2, binwidth = 50, zstar = 10000, t0 = 0, t1 = 0.2)

Description

Fit bunching model to (binned) data and estimate excess mass.

Usage

fit_bunching(thedata, themodelformula, binwidth, notch = FALSE, zD_bin = NA)

Arguments

thedata (binned) data that includes all variables necessary for fitting the model.

themodelformula formula to fit.

binwidth a numeric value for the width of each bin.

notch whether analysis is for a kink or notch. Default is FALSE (kink).

zD_bin the bin marking the upper end of the dominated region (notch case).
marginal_buncher

Value

fit_bunching returns a list of the following results:

- **coefficients**: The coefficients from the fitted model.
- **residuals**: The residuals from the fitted model.
- **cf_density**: The estimated counterfactual density.
- **bunchers_excess**: The estimate of the excess mass (not normalized).
- **cf_bunchers**: The counterfactual estimate of counts in the bunching region.
- **b_estimate**: The estimate of the normalized excess mass.
- **bins_bunchers**: The number of bins in the bunching region.
- **model_formula**: The model formula used for fitting.
- **B_zl_zstar**: The count of bunchers in the bunching region below and up to zstar.
- **B_zstar_zu**: The count of bunchers in the bunching region above zstar.
- **alpha**: The estimated fraction of bunchers in the dominated region (only in notch case.)
- **zD_bin**: The value of the bin which zD falls in.

See Also

bunchit, prep_data_for_fit

Examples

data(bunching_data)
binned_data <- bin_data(z_vector = bunching_data$kink, zstar = 10000, binwidth = 50, bins_l = 20, bins_r = 20)
prepped_data <- prep_data_for_fit(binned_data, zstar = 10000, binwidth = 50, bins_l = 20, bins_r = 20, poly = 4)
fitted <- fit_bunching(thedata = prepped_data$data_binned, themodelformula = prepped_data$model_formula, binwidth = 50)

# extract coefficients
fitted$coefficients

---

marginal_buncher  
Marginal Buncher

Description

Calculate location (value of z_vector) of marginal buncher.

Usage

marginal_buncher(beta, binwidth, zstar, notch = FALSE, alpha = NULL)
notch_equation

Arguments

- **beta**: normalized excess mass.
- **binwidth**: a numeric value for the width of each bin.
- **zstar**: a numeric value for the bunching point.
- **notch**: whether analysis is for a kink or notch. Default is FALSE (kink).
- **alpha**: the proportion of individuals in dominated region (in notch setting).

Value

**marginal_buncher** returns the location of the marginal buncher, i.e. \( z_{\text{star}} + Dz_{\text{star}} \).

See Also

**bunchit**

Examples

```
marginal_buncher(beta = 2, binwidth = 50, zstar = 10000)
```

---

**notch_equation**

Notch Equation

Description

Defines indifference condition based on parametric utility function in notch setting. Used to parametrically solve for elasticity.

Usage

```
notch_equation(e, t0, t1, zstar, dzstar)
```

Arguments

- **e**: elasticity.
- **t0**: numeric value setting the marginal (average) tax rate below \( z_{\text{star}} \) in a kink (notch) setting.
- **t1**: numeric value setting the marginal (average) tax rate above \( z_{\text{star}} \) in a kink (notch) setting.
- **zstar**: a numeric value for the bunching point.
- **dzstar**: The distance of the marginal buncher from \( z_{\text{star}} \).

Value

**util_diff** returns the difference in utility between \( z_{\text{star}} \) and \( z_{\text{I}} \) in notch setting.
plot_bunching

See Also

bunchit
elasticity

Examples

notch_equation(e = .04, t0 = 0, t1 = .2, zstar = 10000, dzstar = 50)

Description

Creates the bunching plot.

Usage

plot_bunching(
  z_vector,
  binned_data,
  cf,
  zstar,
  binwidth,
  bins_excl_l = 0,
  bins_excl_r = 0,
  p_title = "",
  p_xtile = deparse(substitute(z_vector)),
  p_ytitle = "Count",
  p_miny = 0,
  p_maxy = NA,
  p_ybreaks = NA,
  p_title_size = 11,
  p_axis_title_size = 10,
  p_axis_val_size = 8.5,
  p_freq_color = "black",
  p_cf_color = "maroon",
  p_zstar_color = "red",
  p_grid_major_y_color = "lightgrey",
  p_freq_size = 0.5,
  p_freq_msize = 1,
  p_cf_size = 0.5,
  p_zstar_size = 0.5,
  p_b = FALSE,
  b = NA,
  b_sd = NA,
  p_e = FALSE,
plot_bunching

```r
e = NA,
e_sd = NA,
p_b_e_xpos = NA,
p_b_e ypos = NA,
p_b_e_size = 3,
t0 = NA,
t1 = NA,
notch = FALSE,
p_domregion_color = NA,
p_domregion_ltype = NA
```

**Arguments**

- `z_vector`: a numeric vector of (unbinned) data.
- `binned_data`: binned data with frequency and estimated counterfactual.
- `cf`: the counterfactual to be plotted.
- `zstar`: a numeric value for the the bunching point.
- `binwidth`: a numeric value for the width of each bin.
- `bins_excl_l`: number of bins to left of `zstar` to include in bunching region. Default is 0.
- `bins_excl_r`: number of bins to right of `zstar` to include in bunching region. Default is 0.
- `p_title`: plot’s title. Default is empty.
- `p_xtitle`: plot’s x_axis label. Default is the name of `z_vector`.
- `p_ytitle`: plot’s y_axis label. Default is "Count".
- `p_miny`: plot’s minimum y_axis value. Default is 0.
- `p_maxy`: plot’s maximum y_axis value. Default is optimized internally.
- `p_ybreaks`: a numeric vector of y-axis values at which to add horizontal line markers in plot. Default is optimized internally.
- `p_title_size`: size of plot’s title. Default is 11.
- `p_axis_title_size`: size of plot’s axes’ title labels. Default is 10.
- `p_axis_val_size`: size of plot’s axes’ numeric labels. Default is 8.5.
- `p_freq_color`: plot’s frequency line color. Default is "black".
- `p_cf_color`: plot’s counterfactual line color. Default is "maroon".
- `p_zstar_color`: plot’s bunching region marker lines color. Default is "red".
- `p_grid_major_y_color`: plot’s y-axis major grid line color. Default is "lightgrey".
- `p_freq_size`: plot’s frequency line thickness. Default is 0.5.
- `p_freq_msize`: plot’s frequency line marker size. Default is 1.
- `p_cf_size`: plot’s counterfactual line thickness. Default is 0.5.
- `p_zstar_size`: plot’s bunching region marker line thickness. Default is 0.5.
plot_bunching

whether plot should also include the bunching estimate. Default is FALSE.

normalized bunching estimate.

whether plot should also include the elasticity estimate. Only shown if p_b is TRUE. Default is FALSE.

elasticity estimate.

standard deviation of the elasticity estimate.

plot’s x-axis coordinate of bunching/elasticity estimate. Default is set internally.

plot’s y-axis coordinate of bunching/elasticity estimate. Default is set internally.

size of plot’s printed bunching/elasticity estimate. Default is 3.

numeric value setting the marginal (average) tax rate below zstar in a kink (notch) setting.

numeric value setting the marginal (average) tax rate above zstar in a kink (notch) setting.

whether analysis is for a kink or notch. Default is FALSE (kink).

plot’s dominated region marker line color in notch setting. Default is "blue".

line type for the vertical line type marking the dominated region (zD) in the plot for notch settings. Default is "longdash".

Value

plot_bunching returns a plot with the frequency, counterfactual and bunching region demarcated. Can also include the bunching and elasticity estimate if specified.

See Also

bunchit

Examples

data(bunching_data)
binned_data <- bin_data(z_vector = bunching_data$kink, zstar = 10000, binwidth = 50, bins_l = 20, bins_r = 20)
prepped_data <- prep_data_for_fit(binned_data, zstar = 10000, binwidth = 50, bins_l = 20, bins_r = 20, poly = 4)
fitted <- fit_bunching(thedata = prepped_data$data_binned, themodelformula = prepped_data$model_formula, binwidth = 50)
plot_bunching(z_vector = bunching_data$kink_vector, binned_data = prepped_data$data_binned, cf = fitted$cf_density, zstar = 10000, binwidth = 50, bins_excl_l = 0, bins_excl_r = 0, b = 1.989, b_sd = 0.005, p_b = TRUE)
plot_hist

Plot Histogram

Description

Create a binned plot for quick exploration without estimating bunching mass.

Usage

```r
plot_hist(
  z_vector,
  binv = "median",
  zstar,
  binwidth,
  bins_l,
  bins_r,
  p_title = "",
  p_xtitle = "z_name",
  p_ytitle = "Count",
  p_title_size = 11,
  p_axis_title_size = 10,
  p_axis_val_size = 8.5,
  p_miny = 0,
  p_maxy = NA,
  p_ybreaks = NA,
  p_grid_major_y_color = "lightgrey",
  p_freq_color = "black",
  p_zstar_color = "red",
  p_freq_size = 0.5,
  p_freq_msize = 1,
  p_zstar_size = 0.5,
  p_zstar = TRUE
)
```

Arguments

- `z_vector`: a numeric vector of (unbinned) data.
- `binv`: a string setting location of zstar within its bin ("min", "max" or "median" value). Default is median.
- `zstar`: a numeric value for the bunching point.
- `binwidth`: a numeric value for the width of each bin.
- `bins_l`: number of bins to left of zstar to use in analysis.
- `bins_r`: number of bins to right of zstar to use in analysis.
- `p_title`: plot’s title. Default is empty.
- `p_xtitle`: plot’s x_axis label. Default is the name of `z_vector`. 
plot_hist

p_ytitle - plot’s y_axis label. Default is “Count”.

p_title_size - size of plot’s title. Default is 11.

p_axis_title_size - size of plot’s axes’ title labels. Default is 10.

p_axis_val_size - size of plot’s axes’ numeric labels. Default is 8.5.

p_miny - plot’s minimum y_axis value. Default is 0.

p_maxy - plot’s maximum y_axis value. Default is optimized internally.

p_ybreaks - a numeric vector of y-axis values at which to add horizontal line markers in plot. Default is optimized internally.

p_grid_major_y_color - plot’s y-axis major grid line color. Default is “lightgrey”.

p_freq_color - plot’s frequency line color. Default is “black”.

p_zstar_color - plot’s bunching region marker lines color. Default is “red”.

p_freq_size - plot’s frequency line thickness. Default is 0.5.

p_freq_msize - plot’s frequency line marker size. Default is 1.

p_zstar_size - plot’s bunching region marker line thickness. Default is 0.5.

p_zstar - whether to show vertical line for zstar. Default is TRUE.

Value

plot_hist returns a list with the following:

plot - the plot of the density without estimating a counterfactual.

data - the binned data used for the plot.

See Also

bunchit

Examples

# visualize a distribution
data(bunching_data)
plot_hist(z_vector = bunching_data$kink_vector,
          binv = "median", zstar = 10000,
          binwidth = 50, bins_l = 40, bins_r = 40)$plot
Description

Prepare binned data and model for bunching estimation.

Usage

```r
prep_data_for_fit(
  data_binned,  # dataframe of counts per bin
  zstar,        # a numeric value for the bunching point.
  binwidth,     # a numeric value for the width of each bin.
  bins_l,       # number of bins to left of zstar to use in analysis.
  bins_r,       # number of bins to right of zstar to use in analysis.
  poly = 9,     # a numeric value for the order of polynomial for counterfactual fit. Default is 9.
  bins_excl_l = 0,  # number of bins to left of zstar to include in bunching region. Default is 0.
  bins_excl_r = 0,  # number of bins to right of zstar to include in bunching region. Default is 0.
  rn = NA,       # a numeric vector of (up to 2) round numbers to control for. Default includes no controls.
  extra_fe = NA,  # a numeric vector of bin values to control for using fixed effects. Default includes no controls.
  correct_above_zu = FALSE  # if integration constraint correction is implemented, should counterfactual be shifted only above zu (upper bound of exclusion region)? Default is FALSE (i.e. shift from above zstar).
)
```

Arguments

- **data_binned**: dataframe of counts per bin
- **zstar**: a numeric value for the bunching point.
- **binwidth**: a numeric value for the width of each bin.
- **bins_l**: number of bins to left of zstar to use in analysis.
- **bins_r**: number of bins to right of zstar to use in analysis.
- **poly**: a numeric value for the order of polynomial for counterfactual fit. Default is 9.
- **bins_excl_l**: number of bins to left of zstar to include in bunching region. Default is 0.
- **bins_excl_r**: number of bins to right of zstar to include in bunching region. Default is 0.
- **rn**: a numeric vector of (up to 2) round numbers to control for. Default includes no controls.
- **extra_fe**: a numeric vector of bin values to control for using fixed effects. Default includes no controls.
- **correct_above_zu**: if integration constraint correction is implemented, should counterfactual be shifted only above zu (upper bound of exclusion region)? Default is FALSE (i.e. shift from above zstar).
Value

data_binned returns a list with the following:

- **data_binned**: The binned data with the extra columns necessary for model fitting, such as indicators for bunching region, fixed effects, etc.

- **model_formula**: The formula used for model fitting.

See Also

- bunchit

Examples

data(bunching_data)
binned_data <- bin_data(z_vector = bunching_data$kink, zstar = 10000, binwidth = 50, bins_l = 20, bins_r = 20)

prepped_data <- prep_data_for_fit(binned_data, zstar = 10000, binwidth = 50, bins_l = 20, bins_r = 20, poly = 4, bins_excl_l = 2, bins_excl_r = 3, rn = c(250,500), extra_fe = 10200)

head(prepped_data$data_binned)
prepped_data$model_formula
Index

* datasets
  bunching_data, 3
bin_data, 2
bunching, 3, 4
bunching_data, 3
bunchit, 2–4, 4, 10–12, 14–17, 19, 21, 23

do_bootstrap, 10
do_correction, 11
domregion, 9
elasticity, 13, 17
fit_bunching, 12, 14
marginal_buncher, 15
notch_equation, 16
plot_bunching, 17
plot_hist, 3, 8, 20
prep_data_for_fit, 11, 15, 22