

Functions Expost IOp

Pedro Salas and Paolo Brunori

July 2023

Documentation for the R functions used to estimate expost IOp. They are defined to be used for the GEOM database.¹

1 Libraries

Libraries are updated on the 1st June, 2023.

¹Note: Some functions call a weights argument. If you do not want to use weights, use a vector of value 1.

Library	Resource
trtf	https://cran.r-project.org/web/packages/trtf/index.html
tram	https://cran.r-project.org/web/packages/tram/index.html
dineq	https://cran.r-project.org/web/packages/dineq/index.html
lattice	https://cran.r-project.org/web/packages/lattice/index.html
latticeExtra	https://cran.r-project.org/web/packages/latticeExtra/index.html
multcomp	https://cran.r-project.org/web/packages/multcomp/index.html
memisc	https://cran.r-project.org/web/packages/memisc/index.html
Matrix	https://cran.r-project.org/web/packages/Matrix/index.html
colorspace	https://cran.r-project.org/web/packages/colorspace/index.html
grid	https://cran.r-project.org/src/contrib/Archive/grid/
libcoin	https://cran.r-project.org/web/packages/libcoin/index.html
inum	https://cran.r-project.org/web/packages/inum/index.html
partykit	https://cran.r-project.org/web/packages/partykit/index.html
ATR	https://cran.r-project.org/web/packages/ATR/index.html
mlt	https://cran.r-project.org/web/packages/mlt/index.html
modi	https://cran.r-project.org/web/packages/modi/index.html
resample	https://cran.r-project.org/web/packages/resample/index.html
scales	https://cran.r-project.org/web/packages/scales/index.html
forcats	https://cran.r-project.org/web/packages/forcats/index.html
ggsankey	https://github.com/davidsjoberg/ggsankey
metan	https://cran.r-project.org/web/packages/metan/index.html
RColorBrewer	https://cran.r-project.org/web/packages/RColorBrewer/index.html
gtools	https://cran.r-project.org/web/packages/gtools/index.html

2 Code

2.1 Tune Trafotree

`tune_trafotree(data, model, folds = 5, minorder = 2, maxorder = 10, mindiff = 0.001, plot = TRUE)`

- data: name of dataframe.
- model: formula, e.g., `dependent ~ independents`.
- folds: number of folds for the cross-validation. Default value: 5.
- minorder: minimum order to search. Default value: 2.
- maxorder: maximum order to search. Default value: 10.
- mindiff: relative improvement in the out of sample log-likelihood to select the order. For instance, 0.01 means an improvement of 1%. Default value: 0.001.
- plot: if TRUE, get a plot of the tuning.

The function returns four objects:

- order: order tuned.
- loglik: log-likelihood of the order tuned.
- res: results generated in the tuning.
- plot: object containing the plot, if plot = TRUE.

2.2 Get Trafotree

`get_trtree(data, model, dep, order = 5, maxd = Inf, mincri = 0.99, minbu = 50, centiles = 99, rel.ineq = TRUE, lenv = TRUE, share.lenv = 0.1)`

- data: name of dataframe.
- model: formula, e.g., dependent independents.
- dep: name of the dependent variable (with quotation marks, e.g., "income").
- order: order tuned for the Bernstein Polynomial. Default = 5.
- mincri: mincriterion (1-alpha) of each tree. Default value = 0.99.
- minbu: minbucket of each tree. Default value = 50.
- centiles: number of centiles to set tranches. Default = 99.
- rel.ineq: If TRUE, y_{tilde} is delivered to be estimated with relative inequality. If FALSE, y_{tilde} is delivered to be estimated with the variance.
- lenv: Estimate Lower Envelope. Default = TRUE.
- share.lenv: Estimate Robust Lower Envelope with the bottom XX%. Default XX = 0.1 (10%).

The function returns six objects:

- trafodata: data frame containing variable " y_{tilde} ", the smoothed dependent variable. It also includes the original data and the intermediate variables used to estimate IOp.
- tree: transformation tree object, that can be used for plots, own predictions and any other task.
- qtl: quantiles used in the data.
- tr: matrix containing the interpolations of the Bernstein Polynomial, used to estimate IOp.
- eop: estimate of lower envelope (area).
- eopx: estimate of robust lower envelope (area).

Note: The function sets testtype = "Bonferroni".

2.3 Get Plot with nice nodes

`plot(object, terminal_panel = node_dense(colordens = "E0112B"))`

- `object`: trafotree object.
- `terminal_panel`: how terminal nodes should be shown. Default `node_dense`, showing the densities of log incomes.
- `colordens`: color used to fill the densities. Default: "E0112B", the corporate red color used by III/LSE.

2.4 Get Log Interpolations

`get_tr(data, dep, types, order = 5, centiles = 99, rel.lineq = TRUE, lenv = TRUE, share_lenv = TRUE)`

- `data`: name of dataframe.
- `dep`: name of the dependent variable (with quotation marks, e.g., "income").
- `types`: name of type variable (with quotation marks, e.g., "types").
- `order`: order of the Bernstein Polynomial. Default: 5.
- `centiles`: number of centiles to interpolate the distribution. Default value: 99.
- `rel.lineq`: If TRUE, `y_tilde` is delivered to be estimated with relative inequality. If FALSE, `y_tilde` is delivered to be estimated with the variance.
- `lenv`: Estimate Lower Envelope. Default = TRUE.
- `share_lenv`: Estimate Robust Lower Envelope with the bottom XX%. Default XX = 0.1 (10%).

The function returns three objects:

- `trafodata`: dataset with all new variables from log interpolation.
- `qtl`: quantiles used in the data.
- `tr`: matrix containing the interpolations of the Bernstein Polynomial, used to estimate IOP.

2.5 Arrange colors for Plots

`colplot(data, dep, types, grouping_var)`

- `data`: name of dataframe.
- `dep`: name of the dependent variable (with quotation marks, e.g., "income").
- `types`: name of type variable (with quotation marks, e.g., "types").
- `grouping_var`: name of variable that groups colors (with quotation marks, e.g., "types").
- `palette`: palette of colours to be used in the plot

The function returns three objects:

- `plot_data` : dataset that can be used to plot, with the modified variables and labels.
- `col`: vector of colors.
- `n_types`: number of types.

2.6 Plot ECDFs

`plot_ponytail(data, dep, group, fill, quart_var, tr, nti, col, limit_x, x_lab = "", y_lab = "", labs = "", legend_pos = "", title = "")`

- `data`: name of dataframe.
- `dep`: name of the dependent variable (with quotation marks, e.g., "income").
- `group`: name of grouping variable (with quotation marks, e.g., "types").
- `fill`: name of filling variable (with quotation marks, e.g., "types").
- `quart_var`: object with quantiles.
- `tr`: object with tr matrix.
- `nti`: number of types.
- `col`: vector of colors.
- `limit_x`: range of x axis.
- `x_lab` — `y_lab` — `labs`: labels for x axis, y axis and other labels.
- `legend_pos`: position of legend.
- `title`: title of the plot.

The function returns a plot with the ECDFs and the interpolations.

2.7 Plot Densities

`plot_mountain(data, dep, group, fill, col, limit_x, x_lab = "", y_lab = "", labs = "", legend_pos = "bottom", title = "")`

- `data`: name of dataframe.
- `dep`: name of the dependent variable (with quotation marks, e.g., "income").
- `group`: name of grouping variable (with quotation marks, e.g., "types").
- `fill`: name of filling variable (with quotation marks, e.g., "types").
- `col`: vector of colors.
- `limit_x`: range of x axis.
- `x_lab` — `y_lab` — `labs`: labels for x axis, y axis and other labels.
- `legend_pos`: position of legend. Default: "bottom".
- `title`: title of the plot.

The function returns a plot with the density as a mixture of type-specific densities.