Package 'ExtremeBounds'

June 20, 2023

Type Package

Title Extreme Bounds Analysis (EBA)

Version 0.1.7

Date 2023-06-23

Author Marek Hlavac <mhlavac@alumni.princeton.edu>

Maintainer Marek Hlavac <mhlavac@alumni.princeton.edu>

Description An implementation of Extreme Bounds Analysis (EBA), a global sensitivity analysis that examines the robustness of determinants in regression models. The package supports both Leamer's and Sala-i-Martin's versions of EBA, and allows users to customize all aspects of the analysis.

License GPL (>= 2)

Imports Formula, graphics, methods, stats, utils

LazyLoad yes

Collate 'ExtremeBounds-internal.R' 'ExtremeBounds.R'

NeedsCompilation no

Repository CRAN

Date/Publication 2023-06-20 21:50:02 UTC

R topics documented:

---

eba   *Extreme Bounds Analysis*

Description

eba is used to perform extreme bounds analysis (EBA), a global sensitivity test that examines the robustness of the association between a dependent variable and a variety of possible determinants. The eba function performs a demanding version of EBA, proposed by Leamer (1985), that focuses on the upper and lower extreme bounds of regression estimates, as well as a more flexible version proposed by Sala-i-Martin (1997). Sala-i-Martin's EBA considers the entire distribution of regression coefficients. For Sala-i-Martin's version of extreme bounds analysis, eba estimates results for both the normal model (in which regression coefficients are assumed to be normally distributed across models) and the generic model (where no such assumption is made).

Usage

```
eba(formula = NULL, data,
    y = NULL, free = NULL, doubtful = NULL, focus = NULL,
    k = 0:3, mu = 0, level = 0.95, vif = NULL, exclusive = NULL,
    draws = NULL, reg.fun = lm, se.fun = NULL, include.fun = NULL,
    weights = NULL, ...)
```

Arguments

formula	a formula that specifies the EBA model that the function will run. Most generally, the formula is of the following format: y ~ free | focus | (additional) doubtful. See the arguments y, free, doubtful and focus below for a more detailed description of what each of these variable categories means. Note that one can also specify that all doubtful variables are of interest (i.e., all are 'focus' variables): y ~ free | focus. Finally, the user can also specify a model with no free variables, in which all doubtful variables are considered to be 'focus': y ~ focus.
data	a data frame containing the variables used in the extreme bounds analysis.
y	a character string that specifies the dependent variable.
free	a character vector that specifies the 'free' variables to be used in the analysis. These variables are included in each regression model.
doubtful	a character vector that specifies the 'doubtful' variables to be used in the analysis. These variables will be included, in various combinations, in the estimated regression models.
focus	a character vector that specifies the 'focus' variables of the extreme bounds analysis. These are the variables whose robustness the user wants to test. The focus variables must be a subset of the variables included in the argument doubtful. Since these are the variables of interest, eba will only run regressions with doubtful variable combinations that contain at least one focus variable.

k	a vector of integers that specifies the number of doubtful variables that will be included in each estimated regression model in addition to the focus variable. Following Levine and Renelt (1992), the default is set to `0:3`, meaning that up to three additional doubtful variables will be included in each model on top of the focus variable.
mu	a named vector of numeric values that specifies regression coefficients under the null hypothesis. The names of the vector's elements indicate which variable the null hypothesis coefficients belong to. These null hypothesis coefficient values will be used in all hypothesis testing. Alternatively, the argument `mu` can be a single numeric value that will set the null hypothesis values for all variables' coefficients. By default, `mu` will be equal to zero for all examined variables, as is standard in most applications of extreme bounds analysis.
level	a numeric value between 0 and 1 that indicates the confidence level to be used in determining the robustness/fragility of determinants.
vif	a numeric value that sets the maximum limit on a coefficient's variance inflation factor (VIF), a rule-of-thumb indicator of multicollinearity. Only coefficient estimates whose VIF does not exceed the limit will be considered in the analysis. If `NULL` (default), no limit on the VIF is imposed.
exclusive	a list of character vectors, or a formula with sets of mutually exclusive variables separated by `|`. Each character vector (or formula component) specifies a set of mutually exclusive doubtful variables. These variables will never be included in the same regression model. Specifying which doubtful variables may not be included together in the same model can help alleviate concerns about regressor multicollinearity, and can also be useful when several doubtful variables measure the same substantive concept.
draws	a positive integer value that specifies how many regressions eba should be run. These regressions will be randomly drawn (without replacement, and each with equal probability) from the full set of doubtful variable combinations (that, of course, contain the variables specified in `focus`). Such a random draw can be useful when estimating the full set of regressions would require too much time. If NULL (default), there will be no random sampling of regression model and all combinations will be estimated.
reg.fun	a function that estimates the desired regression model. The function must accept arguments `formula` and `data` in the same way that the standard functions `lm` and `glm` do. Additional arguments can be passed on via the . . . argument. In this way, the user can make eba estimate, say, a logistic or probit regression by setting `reg.fun = glm` and passing on the appropriate values for `glm`'s `family` argument through eba's . . . argument. By default, an Ordinary Least Squares (OLS) regression is performed via the `lm` function.
se.fun	a function that calculates the standard errors for regression coefficient estimates. The function must accept the regression model object as its first argument, and must return a numeric vector with element names that identify the corresponding regressors.
include.fun	a function that determines whether the results from a particular regression model will be included in the analysis. The function must accept the regression model object as its first argument, and must return a logical value. Only regression

models for which the function returns a value of TRUE will be included in the extreme bounds analysis.

weights   a character string or a function that specifies what weights will be applied to the results from each estimated regression model. The default value of NULL means that each model will have an equal weight. If the argument is set to "adj.r.squared", "lri", "r.squared", the regression results will be weighted based on the adjusted R-squared, the likelihood ratio index (McFadden, 1974), or the R-squared, respectively.

...   additional arguments that will be passed on to the regression function specified by reg.fun.

Details

If the argument focus is NULL, it is populated by the content of doubtful. Conversely, if doubtful is NULL, it will be filled in with values from focus. It is thus sufficient to specify only one of doubtful or focus to test the robustness of all doubtful variables.

The character strings in arguments y, free, doubtful, focus and exclusive can contain model formula operators described in [formula](such as :, *, ^, %in%), as well as the function [I](https://). In addition, the variables in character strings can be enclosed within other functions: "log(x)", for instance, represents the natural logarithm of x.

The summary object obtained from the regression function specified in argument reg.fun should contain a coefficients matrix component. eba will collect the coefficient estimates, standard errors, test statistics and p-values from the first, second, third and fourth columns of the coefficients matrix, respectively. The number of observations is equal to length(x$residuals), where x is the regression model object.

The calculation of weights based on McFadden's likelihood ratio index (see argument weights) relies on the generic accessor function logLik. If weights are based on the regression's R-squared and adjusted R-squared, eba obtains the values of these statistics from the model object's components r.squared and adj.r.squared, respectively.

Value

eba returns an object of class "eba". The corresponding summary function (i.e., summary.eba) returns the same object.

An object of class "eba" is a list containing the following components:

bounds   a data frame with the results of the extreme bounds analysis. The data frame bounds contains the following columns:

- type: type of reported variable - either "free" or "focus".
- mu: the regression coefficient under the null hypothesis.
- beta.below.mu: proportion of estimated regression coefficients whose value is less than mu.
- beta.above.mu: proportion of estimated regression coefficients whose value is greater than mu.
- beta.significant: proportion of regression models in which the estimated coefficient is statistically significantly different from mu.

- beta.significant.below.mu: proportion of estimated regression coefficients that are both statistically significantly different from and whose value is less than mu.
- beta.significant.above.mu: proportion of estimated regression coefficients that are both statistically significantly different from and whose value is greater than mu.
- leamer.lower: Leamer's lower extreme bound at the specified confidence level.
- leamer.upper: Leamer's upper extreme bound at the specified confidence level.
- leamer.robust: logical value indicating whether the variable is robust based on Leamer's extreme bounds analysis. If leamer.lower and leamer.upper have the same sign, the value will be TRUE. If they have opposite signs, leamer.robust will be FALSE.
- cdf.mu.normal: the value of the cumulative density function at CDF(mu) - i.e., the proportion of coefficients that are estimated to be lower or equal to mu - based on Sala-i-Martin's EBA that assumes that regression coefficients are normally distributed across the estimated models. Weights specified by eba's argument weights are applied.
- cdf.above.mu.normal: equal to 1 − cdf.mu.normal. This value represents the proportion of coefficients that are estimated to be greater than mu, based on Sala-i-Martin's EBA that assumes that regression coefficients are normally distributed across the estimated models. Weights specified by eba's argument weights are applied.
- cdf.mu.generic: the value of the cumulative density function at CDF(mu) based on Sala-i-Martin's EBA that does not assume any particular distribution of regression coefficients across the estimated models. Weights specified by eba's argument weights are applied.
- cdf.above.mu.generic: equal to 1 − cdf.mu.generic. This value represents the proportion of coefficients that are estimated to be greater than mu, based on Sala-i-Martin's EBA that does not assume any particular distribution of regression coefficients across the estimated models. Weights specified by eba's argument weights are applied.

call   the matched call.

coefficients a list that contains data frames with selected quantities of interest that emerge from the extreme bounds analysis. This list can also be extracted by calling the generic accessor function coefficients on the "eba" object. The list coefficients contains the following data frame components:

- cdf.generic.unweighted: the CDF(mu) and (1-CDF(mu)) based on Sala-i-Martin's generic EBA that does not assume any distribution of regression coefficients across models. Each regression model receives an equal weight.
- cdf.generic.weighted: the CDF(mu) and (1-CDF(mu)) based on Sala-i-Martin's generic EBA that does not assume any distribution of regression coefficients across models. Individual regression models receive a weight specified by the argument weights.

- min: the value of the lowest regression coefficient across the estimated models, along with additional statistics.
- max: the value of the highest regression coefficient across the estimated models, along with additional statistics.
- mean: the mean of the estimated regression coefficients, standard errors and variances. Each regression model receives an equal weight. Note that the mean of the variances will generally not be equal to the square of standard errors.
- weighted.mean: the weighted mean of the estimated regression coefficients, standard errors and variances. Individual regression models receive a weight specified by the argument weights. Note that the weighted mean of the variances will generally not be equal to the square of standard errors.
- median: the value of the median regression coefficient across the estimated models, along with additional statistics. If no unambiguous median value exists, NA is reported.
- median.lower: the value of the median regression coefficient across the estimated models, along with additional statistics. If no unambiguous median value exists, the lower of the two 'potential median' coefficients is reported.
- median.higher: the value of the median regression coefficient across the estimated models, along with additional statistics. If no unambiguous median value exists, the higher of the two 'potential median' coefficients is reported.
- min.ci.lower: the minimum value of the lower bound of the confidence interval (at the requested confidence level) across the estimated models, along with additional statistics. This value represents the lower extreme bound in Leamer's EBA.
- max.ci.upper: the maximum value of the upper bound of the confidence interval (at the requested confidence level) across the estimated models, along with additional statistics. This value represents the upper extreme bound in Leamer's EBA.

mu	a named vector of regression coefficients under the null hypothesis for each variable.
level	a number between 0 and 1 that indicates the confidence level for hypothesis testing.
ncomb	total number of doubtful variable combinations that include at least one focus variable.
nreg	total number of regressions that were estimated as part of the extreme bounds analysis. When draws is NULL (i.e., no random sampling of regression models is requested), ncomb and nreg will be equal.
nreg.variable	a named vector containing the the number of estimated regressions that included each variable.
ncoef.variable	a named vector containing the the number of estimated coefficients that were used in the extreme bounds analysis. This number can differ from nreg.variable when vif or include.fun is specified.
regressions	a list that contains estimation results for each regression that was run as part of the extreme bounds analysis. This list contains several components which

store quantities such as coefficient or standard error estimates for each of the estimated regressions. Each of these components is a matrix whose number of rows corresponds to the total number of regressions (equal to `nreg`) and whose columns represent individual regressors. In each of the component matrices, results from a particular regression model will be included in the same row. The list `regressions` contains the following components:

- `beta`: regression coefficients.
- `se`: standard errors.
- `var`: variances of the regression coefficients. This value is equal to the square of `se`.
- `t`: test statistics (typically t- or z-statistics; might depend on the regression function used - see argument `reg.fun`).
- `p`: p-values.
- `ci.lower`: lower bound of the confidence interval for the requested confidence level (see argument `level`).
- `ci.upper`: upper bound of the confidence interval for the requested confidence level (see argument `level`).
- `nobs`: number of observations.
- `vif`: variance inflation factor (VIF).
- `vif.satisfied`: a logical value that indicates whether the variance inflation factor is within the set maximum limit (see argument `vif`).
- `formula`: a character string containing the model formula, a symbolic description of the model that was fitted.
- `weight`: a numerical value that represents the weight that is given to a particular regression model in the extreme bounds analysis.
- `cdf.mu.generic`: the value of cumulative distribution function of the regression coefficients at the value of `mu`. The coefficients are assumed to be distributed normally with a mean given by the regression model's coefficient estimate and with a standard deviation given by the estimated standard error. This value is used in estimating the generic version of Sala-i-Martin's EBA.
- `include`: a logical value (TRUE or FALSE) that indicates whether the corresponding model's estimation results are included in the extreme bounds analysis (based on the argument `include.fun`). A value of `NA` that the corresponding regression-variable combination is not included in the analysis (either not part of the variable combination, or omitted due to multicollinearity).
- `nomit`: number of regressions that have been omitted from the analysis, typically due to perfect multicollinearity. They can be found by looking for the regressions in which all `beta` values are equal to `NA`.

Please cite as:

Hlavac, Marek (2016). ExtremeBounds: Extreme Bounds Analysis in R. *Journal of Statistical Software*, 72(9), 1-22. doi: 10.18637/jss.v072.i09.

Author(s)

Marek Hlavac < mhlavac at alumni.princeton.edu >
Research Fellow, Central European Labour Studies Institute (CELSI), Bratislava, Slovakia

References

McFadden, Daniel L. (1974). Conditional Logit Analysis of Qualitative Choice Behavior. In: P. Zarembka (Ed.), *Frontiers in Econometrics*, Academic Press: New York, 105-142.

Leamer, Edward E. (1985). Sensitivity Analysis Would Help. *American Economic Review*, 57(3), 308-313.

Levine, Ross, and David Renelt. (1992). A Sensitivity Analysis of Cross-Country Growth Regressions. *American Economic Review*, 82(4), 942-963.

Sala-i-Martin, Xavier. (1997). I Just Ran Two Million Regressions. *American Economic Review*, 87(2), 178-183. doi:10.3386/w6252.

See Also

hist.eba, print.eba

Examples

```
# perform Extreme Bounds Analysis

eba.results <- eba(formula = mpg ~ wt | hp + gear | cyl + disp + drat + qsec + vs + am + carb,
                   data = mtcars[1:10, ], exclusive = ~ cyl + disp + hp | am + gear)

# The same result can be achieved by running:
# eba.results <- eba(data = mtcars[1:10, ], y = "mpg", free = "wt",
#                    doubtful = c("cyl", "disp", "hp", "drat", "qsec",
#                                 "vs", "am", "gear", "carb"),
#                    focus = c("hp", "gear"),
#                    exclusive = list(c("cyl", "disp", "hp"),
#                                     c("am", "gear")))

# print out results
print(eba.results)

# create histograms
hist(eba.results, variables = c("hp","gear"),
     main = c("hp" = "Gross horsepower", "gear" = "Number of forward gears"))
```

ExtremeBounds   *ExtremeBounds: Extreme Bounds Analysis in R*

Description

The package `ExtremeBounds` performs extreme bounds analysis (EBA), a global sensitivity test that examines the robustness of the association between a dependent variable and a variety of possible determinants. It supports a demanding version of EBA, proposed by Leamer (1985), that focuses on the upper and lower extreme bounds of regression estimates, as well as a more flexible version proposed by Sala-i-Martin (1997). Sala-i-Martin's EBA considers the entire distribution of regression coefficients. For Sala-i-Martin's version of extreme bounds analysis, the package `ExtremeBounds` estimates results for both the normal model (in which regression coefficients are assumed to be normally distributed across models) and the generic model (where no such assumption is made).

The most important function is `eba`, which performs the extreme bounds analysis and stores the results in an object of class `"eba"`. This object can then be passed on to the `print.eba` and `hist.eba` functions to obtain, respectively, a printed summary of EBA results and a set of histograms that illustrate the EBA results graphically.

If you have any comments or suggestions, please do not hesitate to contact the author.

Please cite as:

Hlavac, Marek (2016). ExtremeBounds: Extreme Bounds Analysis in R. *Journal of Statistical Software*, 72(9), 1-22. doi: 10.18637/jss.v072.i09.

Author(s)

Marek Hlavac < mhlavac at alumni.princeton.edu >
Research Fellow, Central European Labour Studies Institute (CELSI), Bratislava, Slovakia

See Also

`eba`, `hist.eba`, `print.eba`

---

hist.eba   *Histograms for Extreme Bounds Analysis*

Description

`hist.eba` is used to generate a set of histograms that present the results of extreme bounds analysis graphically. Each histogram illustrates the distribution of regression coefficients across the models estimated in the course of EBA. In addition, function `hist.eba` can overlay each histogram with lines that indicate the value of the regression coefficient assumed under the null hypothesis (argument `mu.show`), as well as with curves that indicate the distribution's kernel density (argument `density.show`) and a normally distributed approximation (argument `normal.show`). Additional formatting options are available.

Usage

```
## S3 method for class 'eba'
hist(x, variables = NULL, col = "gray", freq = FALSE, main = NULL,
     mu.show = TRUE, mu.col = "red", mu.lwd = 2, mu.visible = TRUE,
   density.show = TRUE, density.col = "blue", density.lwd = 2, density.args = NULL,
     normal.show = FALSE, normal.col = "darkgreen", normal.lwd = 2,
     normal.weighted = FALSE, xlim = NULL, ylim = NULL, ...)
```

Arguments

x	an object of class "eba", typically generated by the [eba](#) function.
variables	a character vector that specifies the variables for which histograms are requested. If NULL, histograms for all variables will be produced.
col	a color to be used to fill the histogram bars. Default is "gray".
freq	logical; if TRUE, the histogram graphic is a representation of frequencies, the counts component of the result; if FALSE, probability densities, component density, are plotted (so that the histogram has a total area of one). Defaults to TRUE if and only if breaks are equidistant (and probability is not specified).
main	a named character vector that specifies the histogram title labels for the requested variables. The name of the vector component specifies the variable, while the content of the component itself contains the title label. If the vector's components are not named, variables are labelled in the order that they appear in the argument variables.
mu.show	logical; if TRUE (default), a vertical line for each variable's regression coefficient value assumed under the null hypothesis (specified by the mu argument of the [eba](#) function, with a default of 0) will be drawn over the histograms.
mu.col	a color to be used to draw the vertical line at mu. Default is "red".
mu.lwd	the line width for vertical line at mu. Default is 2.
mu.visible	logical; if TRUE (default), make sure that the histograms' horizontal axes are scaled so that the vertical line at mu is always visible.
density.show	logical; if TRUE, a kernel density curve for the regression coefficients' distribution will be drawn over the histograms. The kernel densities are calculated using the standard density function. Additional arguments can be passed on to the density function via the density.args argument.
density.col	a color to be used to draw the kernel density curve. Default is "blue".
density.lwd	the line width for the kernel density curve. Default is 2.
density.args	a list of additional arguments that will be passed on to the kernel density function.
normal.show	logical; if TRUE, a density curve for the normal distribution function will be drawn over the histograms. For each variable, the distribution's mean and standard error will be the means and standard errors of the corresponding regression coefficients.
normal.col	a color to be used to draw the normal distribution density curve. Default is "darkgreen".

normal.lwd	the line width for the normal distribution density curve. For more detail, see the documentation for `lwd` in [par](#). Default is 2.
normal.weighted	
	logical; If TRUE, the normal distribution density shown by `normal.show` will use the weighted mean and standard error, with each regression model's weight specified by `eba`'s argument `weights`. If FALSE (default), the unweighted mean and standard error will be used (i.e., each regression model receives equal weight).
xlim	the range of x values with sensible defaults.
ylim	the range of y values with sensible defaults.
...	additional arguments that will be passed on to the [hist](#) function that creates the histograms.

Value

`hist.eba` returns an object of class `"hist.eba"`.

An object of class `"hist.eba"` is a list containing the following components:

call	the matched call
histograms	a list of objects of class `histogram` that store information about the histograms produced by `hist.eba`. The histograms are included in the order that was requested and are indexed by the variable name. See the documentation for the default [hist](#) method for more information.

Please cite as:

Hlavac, Marek (2016). ExtremeBounds: Extreme Bounds Analysis in R. *Journal of Statistical Software*, 72(9), 1-22. doi: 10.18637/jss.v072.i09.

Author(s)

Marek Hlavac < mhlavac at alumni.princeton.edu >
Research Fellow, Central European Labour Studies Institute (CELSI), Bratislava, Slovakia

See Also

[density](#), [eba](#), [hist](#), [print.eba](#)

Examples

```
# perform Extreme Bounds Analysis

eba.results <- eba(formula = mpg ~ wt | hp + gear | cyl + disp + drat + qsec + vs + am + carb,
                   data = mtcars[1:10, ], k = 0:2)

# The same result can be achieved by running:
# eba.results <- eba(data = mtcars[1:10, ], y = "mpg", free = "wt",
```

```
#                 doubtful = c("cyl","disp","hp","drat","qsec","vs","am","gear","carb"),
#                      focus = c("hp","gear"), k = 0:2)

# create histograms, keeping the default settings
hist(eba.results)

# re-create histograms with customized settings
hist(eba.results, variables = c("hp","gear"),
     main = c("hp" = "Gross horsepower", "gear" = "Number of forward gears"),
     mu.visible=FALSE, normal.show=TRUE, normal.lwd=1)
```

---

print.eba   *Print Extreme Bounds Analysis Results*

Description

hist.eba prints the results of extreme bounds analysis (EBA; performed by the eba function) and
returns the printed object invisibly (via invisible(x)). The function prints out information about
the distribution and significance of estimated regression coefficients, the results of Leamer's EBA,
as well as those of Sala-i-Martin's EBA (both the normal and generic model).

Usage

```
## S3 method for class 'eba'
print(x, digits = 3, ...)
```

Arguments

x	an object of class "eba", typically generated by the eba function.
digits	number of decimal places to which the output will be rounded.
...	further arguments passed to print.default and print.data.frame functions within print.eba.

Output

print.eba prints the following information in its output:

- Call: the matched call (based on x$call).
- Confidence level: the confidence level for hypothesis testing (x$level).
- Number of combinations: the total number of doubtful variable combinations that contain at least one focus variable (x$ncomb).
- Regressions estimated: the number of regressions that were estimated in the course of EBA (x$nreg). When no random sampling of regression models was requested (i.e., when eba's argument draws is NULL), the number of combinations (above) will equal the number of regressions estimated.

- Number of regressions by variable: the number of regressions estimated, by variable (x$nreg.variable).

- Number of coefficients used by variable: the number of coefficients used in the extreme bounds analysis, by variable (x$ncoef.variable).

Beta coefficients:

- Coef (Wgt Mean): the weighted mean of the estimated regression coefficients. Individual regression models receive a weight specified by eba's argument weights.

- SE (Wgt Mean): the weighted mean of the standard errors on estimated regression coefficients. Individual regression models receive a weight specified by eba's argument weights.

- Min Coef and SE (Min Coef): the value of the lowest regression coefficient across the estimated models and the corresponding standard error.

- Max Coef and SE (Max Coef): the value of the highest regression coefficient across the estimated models and the corresponding standard error.

Distribution of beta coefficients:

- Pct(beta < mu): proportion of estimated regression coefficients whose value is less than mu.

- Pct(beta > mu): proportion of estimated regression coefficients whose value is greater than mu.

- Pct(significant != mu): proportion of regression models in which the estimated coefficient is statistically significantly different from mu.

- Pct(signif & beta < mu): proportion of estimated regression coefficients that are both statistically significantly different from and whose value is less than mu.

- Pct(signif & beta > mu): proportion of estimated regression coefficients that are both statistically significantly different from and whose value is greater than mu.

Leamer's Extreme Bounds Analysis (EBA):

- Lower Extreme Bound: Leamer's lower extreme bound at the specified confidence level.

- Upper Extreme Bound: Leamer's upper extreme bound at the specified confidence level.

- Robust/Fragile?: a character string indicating whether the variable is robust or fragile based on Leamer's extreme bounds analysis.

Sala-i-Martin's Extreme Bounds Analysis (EBA):

- N: CDF(beta <= 0): the value of the cumulative density function at CDF(mu) - i.e., the proportion of coefficients that are estimated to be lower or equal to mu - based on Sala-i-Martin's EBA that assumes that regression coefficients are normally distributed across the estimated models. Weights specified by eba's argument weights are applied.

- N: CDF(beta > 0): the proportion of coefficients that are estimated to be greater than mu, based on Sala-i-Martin's EBA that assumes that regression coefficients are normally distributed across the estimated models. Weights specified by eba's argument weights are applied.

- G: CDF(beta <= 0): the value of the cumulative density function at CDF(mu) based on Sala-i-Martin's EBA that does not assume any particular distribution of regression coefficients across the estimated models. Weights specified by eba's argument weights are applied.

- G: CDF(beta > 0): the proportion of coefficients that are estimated to be greater than mu, based on Sala-i-Martin's EBA that does not assume any particular distribution of regression coefficients across the estimated models. Weights specified by eba's argument weights are applied.

Note that all values of cumulative density functions for Sala-i-Martin's EBA are printed as percentages.

Please cite as:

Hlavac, Marek (2016). ExtremeBounds: Extreme Bounds Analysis in R. *Journal of Statistical Software*, 72(9), 1-22. doi: 10.18637/jss.v072.i09.

Author(s)

Marek Hlavac < mhlavac at alumni.princeton.edu >
Research Fellow, Central European Labour Studies Institute (CELSI), Bratislava, Slovakia

See Also

eba, hist.eba

Examples

```
# perform Extreme Bounds Analysis

eba.results <- eba(formula = mpg ~ wt | hp + gear | cyl + disp + drat + qsec + vs + am + carb,
                   data = mtcars[1:10, ], k = 0:2)

# The same result can be achieved by running:
# eba.results <- eba(data = mtcars[1:10, ], y = "mpg", free = "wt",
#                 doubtful = c("cyl","disp","hp","drat","qsec","vs","am","gear","carb"),
#                    focus = c("hp","gear"), k = 0:2)

# print out results, rounded to 2 decimal places
print(eba.results, digits = 2)
```

Index