Package ‘sValues’

July 15, 2018

Type Package

Title Measures of the Sturdiness of Regression Coefficients

Version 0.1.6

Author Carlos Cinelli

Maintainer Carlos Cinelli <carloscinelli@hotmail.com>

Description Implements the s-values proposed by Ed. Leamer.
It provides a context-minimal approach for sensitivity analysis using extreme
bounds to assess the sturdiness of regression coefficients.

Imports ggplot2, reshape2

License GPL-3

Suggests knitr, testthat

VignetteBuilder knitr

LazyData true

BugReports https://github.com/carloscinelli/sValues

RoxygenNote 6.0.1

NeedsCompilation no

Repository CRAN

Date/Publication 2018-07-15 21:30:03 UTC

R topics documented:

sValues-package *sValues: measures of the sturdiness of regression coefficients*

Description

The R package sValues implements the measure of sturdiness of coefficients proposed by Leamer (2014) and discussed in Leamer (2015). The S-values try to provide a sensible framework to assess the sensitivity of coefficient estimates to model ambiguity.

Details

The main function of the package is the `sValues` function.

More information can be found on its help documentation, examples and vignette.

The package also includes an example dataset on economic growth.

References

Leamer, E. (2014). S-values: Conventional context-minimal measures of the sturdiness of regression coefficients. Working Paper

Leamer, E. (2015). S-values and bayesian weighted all-subsets regressions. European Economic Review.

coef.sValues *Extract sValues Model Coefficients/Statistics*

Description

Extract sValues Model Coefficients/Statistics

Usage

```
## S3 method for class 'sValues'
coef(object, type = "default", ...)

betas(object)

t_values(object)

s_values(object)

extreme_bounds(object)
```

Arguments

`object` an object of class `sValues`.

`type` which coefficient/statistic to extract? Current options are "betas", "t_values", "s_values", "extreme_bounds" and "default". See details.

`...` further arguments passed to or from other methods.

Details

For the `coef` function, the default is to extract the beta coefficients, t-values and s-values. You can can get each one of those individually by setting `type` to either "betas", "t_values" or "s_values". You can also get the extreme bounds of the estimates by setting `type` to "extreme_bounds". Finally, you can set `type = "all"` to get everything.

For each option of `coef`, there is an alternative helper function with the same name. That is, `coef(x, "betas")` is equivalent to `betas(x)`, or `coef(x, "extreme_bounds")` is equivalent to `extreme_bounds(x)`.

Value

The function returns a `data.frame` with the estimates for each variable.

See Also

`summary.sValues`.

Examples

```
data(economic_growth)
eg_sv <- sValues(GR6096 ~ ., data = economic_growth)
eg_betas <- coef(eg_sv, "betas")
eg_t_values <- coef(eg_sv, "t_values")
eg_s_values <- coef(eg_sv, "s_values")
eg_ext_bounds <- coef(eg_sv, "extreme_bounds")

# get sturdy estimates for R2 bounds 0.5 - 1
eg_s_values[abs(eg_s_values[3]) > 1, 3, drop = FALSE]
```

`economic_growth` *Economic Growth data*

Description

Sala i Martin's (88 countries) Leamer's (87 countries) Original (139 countries)

Usage

```
economic_growth

economic_growth_original

economic_growth_sala_i_martin
```

Format

An object of class `data.frame` with 87 rows and 68 columns.

`plot.sValues` *Plot method for S-values*

Description

Plot methods for objects of the class `sValues`.

Usage

```
## S3 method for class 'sValues'
plot(x, type = "t_s_plot", ...)
```

Arguments

`x`	an object of class `sValues`.
`type`	the type of the plot. Current options are *t_s_plot* which returns a scatterplot of s-values vs t-values for all coefficients and *beta_plot* which returns a plot of the different estimates for the coefficients.
`...`	additional arguments to be passed to the plot functions. See details.

Details

Additional arguments:

`t_s_plot`

- `R2_bounds`: a numeric vector of length two specifying which R2 bounds range to plot.

`beta_plot`

- `variables`: a character vector specifying which variables to plot. Default is "all".
- `error_bar`: should the error bars be plotted? Default is `FALSE`.
- `ext_bounds_shades`: should shades representing the extreme bounds be plotted? Default is `FALSE`.

Value

It returns a ggplot object with the requested plot.

Examples

```
# growth regressions example
data(economic_growth)
eg_sv <- sValues(GR6096 ~ ., data = economic_growth)
plot(eg_sv, R2_bounds = c(0.5, 1))
plot(eg_sv, R2_bounds = c(0.1, 1))
plot(eg_sv, type = "beta_plot", variable = "OPENDEC1", error_bar = FALSE)
plot(eg_sv, type = "beta_plot", variable = "OPENDEC1", error_bar = TRUE)
```

---

print.sValues *Succinct display of S-values results.*

Description

Succinct display of S-values results.

Usage

```
## S3 method for class 'sValues'
print(x, ..., print.length = 6)
```

Arguments

x	an object of class sValues.
...	further arguments passed to or from other methods.
print.length	how many variables to show in the screen? This is used for pretty printing. The default is 6.

Value

NULL

Examples

```
data(economic_growth)
eg_sv <- sValues(GR6096 ~ ., data = economic_growth)
eg_sv
str(eg_sv)
```

str.sValues *str sValues*

Description

str method for sValues.

Usage

```
## S3 method for class 'sValues'
str(object, max.level = 1, ...)
```

Arguments

object	an object of class sValues.
max.level	maximal level of nesting which is applied for displaying nested structures. Default is 1.
...	further arguments passed to or from other methods.

summary.sValues *summary sValues*

Description

For now, this function is equivalent to print.sValues.

Usage

```
## S3 method for class 'sValues'
summary(object, ...)
```

Arguments

object	an object of class sValues.
...	further arguments passed to or from other methods.

sValues *S-values: conventional model ambiguity measures*

Description

The function sValues performs the extreme bound analysis proposed by Leamer (2014) and discussed in Leamer (2015). For further details see the package vignette.

Usage

```
sValues(..., R2_bounds = c(0.1, 0.5, 1), favorites = NULL,
  R2_favorites = NULL, scale = TRUE)

## S3 method for class 'formula'
sValues(formula, data, R2_bounds = c(0.1, 0.5, 1),
  favorites = NULL, R2_favorites = NULL, scale = TRUE, ...)

## S3 method for class 'matrix'
sValues(m, R2_bounds = c(0.1, 0.5, 1), favorites = NULL,
  R2_favorites = NULL, scale = TRUE, ...)

## S3 method for class 'data.frame'
sValues(df, R2_bounds = c(0.1, 0.5, 1),
  favorites = NULL, R2_favorites = NULL, scale = TRUE, ...)
```

Arguments

`...` arguments passed to other methods. The first argument should be a `formula` followed by a `data.frame`; alternatively, as a shortcut, you can omit the `formula` and provide only a `matrix` or a `data.frame`: in that case, the function will automatically consider the first column as the dependent variable and the rest as the independent variables.

`R2_bounds` a numeric vector with two or more R2 bounds to be considered in the analysis. The default values are `c(0.1, 0.5, 1)`, proposed by Leamer (2014).

`favorites` *optional* - a character vector that specifies the "favorite" variables to be used in the analysis. These variables will have different lower and upper R2 bounds as defined in the `R_favorites` argument.

`R2_favorites` *optional* - a numeric vector with two or more R2 bounds for the "favorite" variables.

`scale` should the variables be scaled/standardized to zero mean and unit variance? The default is `TRUE`. If your data is already scaled/standardized you should set this to `FALSE`.

`formula` an object of the class `formula`: a symbolic description of the model to be fitted.

`data` needed only when you pass a formula as first parameter. An object of the class `data.frame` containing the variables used in the analysis.

`m` an object of class `matrix` with the dependent variable in the first column followed by the covariates. The matrix must have column names.

`df` an object of class `data.frame` with the dependent variable in the first column followed by the covariates.

Value

`sValues` returns an object a list of class "sValues" containing the main results of the analysis:

- `info`: a `list` with the general information about the parameters used in the analysis, such as the formula, the data, the bounds and favorite variables.
- `simple`: a `list` with the results of the simple linear regressions for each variable.
- `all`: the results of the linear regression with all variables.
- `bayes`: a `list` with the results of the bayesian regression for each combination of the R2 bounds. Each bayesian regression includes the coefficient estimates, the variance-covariance matrix and the t-values.
- `ext_bounds`: a `list` with the extreme bounds estimates for each combination of the R2 bounds.
- `s_values`: a `data.frame` with the s_values for each combination of the R2 bounds.

References

Leamer, E. (2014). S-values: Conventional context-minimal measures of the sturdiness of regression coefficients. Working Paper

Leamer, E. (2015). S-values and bayesian weighted all-subsets regressions. European Economic Review.

See Also

`coef.sValues` to extract coefficients or statistics;

`print.sValues` for printing;

`summary.sValues` for summaries;

`plot.sValues` for plots.

Examples

```
# growth regressions example
## All variables, No favorites
data(economic_growth)
eg_sv <- sValues(GR6096 ~ ., data = economic_growth)
eg_sv # prints results
plot(eg_sv, R2_bounds = c(0.5, 1))
plot(eg_sv, type = "beta_plot", variable = "P60", error_bar = TRUE)
coefs_eg <- coef(eg_sv) # extract coefficients
coefs_eg

##  only 14 variables
```

```
eg_sv_14 <-  sValues(GR6096 ~GDPCH60L + OTHFRAC + ABSLATIT +
                       LT100CR + BRIT + GOVNOM1 + WARTIME +
                       SCOUT + P60 + PRIEXP70 + OIL +
                       H60 + POP1560 + POP6560, data = economic_growth)
eg_sv_14
coefs_eg_14 <- coef(eg_sv_14)

## With 14 favorites among all variables
favorites <- c("GDPCH60L", "OTHFRAC", "ABSLATIT", "LT100CR",
              "BRIT", "GOVNOM1", "WARTIME", "SCOUT",
              "P60", "PRIEXP70", "OIL", "H60",
              "POP1560", "POP6560")
eg_sv_fav <- sValues(GR6096 ~ ., data = economic_growth, R2_bounds = c(0.5, 1),
                    favorites = favorites, R2_favorites = c(0.4, 0.8))
eg_sv_fav
plot(eg_sv_fav, R2_bounds = c(0.5, 1))
plot(eg_sv_fav, type = "beta_plot", variable = "P60", error_bar = TRUE)
coefs_eg_fav <- coef(eg_sv_fav)
coefs_eg_fav
```

Index