Package: reverseR (via r-universe)

October 21, 2026

Type Package

LazyLoad yes

LazyData yes

Title Linear Regression Stability to Significance Reversal

Version 0.2

Date 2024-09-24

Maintainer Andrej-Nikolai Spiess <draspiess@gmail.com>

Description Tests linear regressions for significance reversal through leave-one(multiple)-out.

License GPL (>= 2)

Depends R (>= 3.5.0)

Imports methods, boot, boot.pval, L1pack, quantreg, isotree, robustbase

NeedsCompilation no

Config/pak/sysreqs libcairo2-dev cmake libfontconfig1-dev libfreetype6-dev libfribidi-dev make libharfbuzz-dev libicu-dev libjpeg-dev libpng-dev libtiff-dev libuv1-dev libwebp-dev libxml2-dev libssl-dev libnode-dev

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

Date/Publication 2024-12-06 10:38:38 UTC

RemoteUrl https://github.com/anspiess/reverser

RemoteRef HEAD

RemoteSha dd11442d0c6746dcde3280455c2a07dd64a3d210

Contents

bootLM 2
Influence plots 3
jackLM 5
lmExact 6
lmInfl 8

pcomp 12
PNAS2015 13
regionInfl 14
rpLM 15

Index **17**

`bootLM` *Nonparametric/Parametric bootstrap linear model*

Description

Nonparametric and parametric bootstrap (sampling cases, residuals or distributions with replacement) method for parameter estimation and confidence interval of a linear model.

Usage

```
bootLM(model, type = c("cases", "residuals", "residuals2", "parametric"),
       R = 10000, alpha = 0.05, ret.models = FALSE)
```

Arguments

`model`	an `lm` model.
`type`	what to bootstrap. See "Details".
`R`	number of bootstrap samples.
`alpha`	the α-level to use as the threshold border.
`ret.models`	logical. If `TRUE`, the `R` models are returned as a list.

Details

If `type = "cases"`, for all (x_i, y_i) datapoints, linear models are created by sampling `R` times - with replacement - from $n \in \{1 \dots N\}$ and building models $Y_n = X_n\beta + \varepsilon$. This is also known as the .632-bootstrap, because the samples will, on average, contain $1 - e^{-1} = 0.632$ unique elements. If `type = "residuals"`, for all residuals ($r_i = y_i - \hat{y}_i$), linear models are created by sampling `R` times - with replacement - from $n \in (1 \dots N)$ and building models $\hat{Y}_i + r_n = X_i\beta + \varepsilon$. If `type = "residuals2"` is selected, scaled and centered residuals $r_n = \frac{r_i}{\sqrt{1-h_{ii}}} - \bar{r}$ according to Davison & Hinkley are used. In the `"parametric"` bootstrap, n values drawn from a normal distribution $j_n \in \mathcal{N}(0, \sigma)$, where $\sigma = \sqrt{\frac{\sum(r_i)^2}{n-p}}$, are added to the fitted values, and linear models are created $\hat{Y}_i + j_n = X_i\beta + \varepsilon$. Parameter estimates are obtained from each sampling, from which the average $\overline{P_n}$ and standard error $\hat{\sigma}$ is calculated as well as a quantile based confidence interval. *p*-values are calculated through inversion of the confidence interval.

Value

A dataframe containing the estimated coefficients, their standard error, lower an upper confidence values and *p*-values. If `ret.models = TRUE` a list with all `R` models is returned.

Author(s)

Andrej-Nikolai Spiess

References

An Introduction to the Bootstrap.
Efron B, Tibshirani R.
Chapman & Hall (1993).

The Bootstrap and Edgeworth Expansion.
Hall P.
Springer, New York (1992).

Modern Statistics with R.
Thulin M.
Eos Chasma Press, Uppsala (2021).

Bootstrap methods and their application.
Davison AC, Hinkley DV.
Cambridge University Press (1997).

Examples

```
## Example with single influencer (#18) and insignificant model (p = 0.115),
## using case bootstrap.
set.seed(123)
a <- 1:20
b <- 5 + 0.08 * a + rnorm(20, 0, 1)
LM <- lm(b ~ a)
bootLM(LM, R = 100)

## using residuals bootstrap.
bootLM(LM, R = 100, type = "residuals")
```

`Influence plots` *Two diagnostic plots for checking p-value influencers*

Description

Two different plot types that visualize *p*-value influencers.

1. `inflPlot`: plots the linear regression, marks the reverser(s) in darkred and displays trend lines for the full and leave-reversers-out data set (black and darkred, respectively).
2. `pvalPlot`: plots the *p*-values for each leave-one-out data point and displays the (log) *p*-values as an index plot with reverser points in darkred, together with the α-border as defined in `lmInfl` and the original models' *p*-value.

Usage

```
inflPlot(infl, measure, ...)
pvalPlot(infl, ...)
```

Arguments

`infl`	an object obtained from `lmInfl`.
`measure`	which influence measure to use, see 'Details'.
`...`	other plotting parameters.

Details

The influence `measure`'s to use are those listed in `lmInfl`, with the following syntax: `"dfb.Slope"`, `"dffit"`, `"cov.r"`, `"cook.d"`, `"hat"`, `"sR"`, `"hadi"`, `"cdr"`, `"Si"`.

Value

The corresponding plot.

Author(s)

Andrej-Nikolai Spiess

References

Regression diagnostics: Identifying influential data and sources of collinearity.
Belsley DA, Kuh E, Welsch RE.
John Wiley, New York (2004).

Applied Regression Analysis: A Research Tool.
Rawlings JO, Pantula SG, Dickey DA.
Springer; 2nd Corrected ed. 1998. Corr. 2nd printing 2001.

Applied Regression Analysis and Generalized Linear Models.
Fox J.
SAGE Publishing, 3rd ed, 2016.

Residuals and Influence in Regression.
Cook RD & Weisberg S.
Chapman & Hall, 1st ed, New York, USA (1982).

Examples

```
set.seed(123)
a <- 1:20
b <- 5 + 0.08 * a + rnorm(20, 0, 1)
LM1 <- lm(b ~ a)
res1 <- lmInfl(LM1)
inflPlot(res1)
pvalPlot(res1)
```

`jackLM` *Jackknife linear model*

Description

Jackknife (Leave-One-Out) method for parameter estimation and confidence interval of a linear model, according to Quenouille (1956).

Usage

```
jackLM(model, alpha = 0.05)
```

Arguments

`model`	an `lm` model.
`alpha`	the α-level to use as the threshold border.

Details

For all (x_i, y_i) datapoints, a linear model is created by leaving out each entry successively, $Y_{-i} = X_{-i}\beta + \varepsilon$. Pseudovalues from obtained and original coefficients are then created, $P_{-i} = (N \cdot \beta) - ((N-1) * \beta_{-i})$, from which the average $\overline{P_{-i}}$ and standard error $\frac{\sigma}{\sqrt{N}}$ is calculated to obtain the classical confidence interval $\overline{X}_n \pm t_{\alpha,\nu}\frac{S_n}{\sqrt{n}}$.

Value

A dataframe containg the estimated coefficients, their standard error, lower an upper confidence values and *p*-values.

Author(s)

Andrej-Nikolai Spiess

References

Notes on bias in estimation.
Quenouille MH.
Biometrika, **43**, 1956, 353-36l.

Examples

```
## Example with single influencer (#18) and insignificant model (p = 0.115).
## Jackknife estimates are robust w.r.t. outlier #18.
set.seed(123)
a <- 1:20
b <- 5 + 0.08 * a + rnorm(20, 0, 1)
LM1 <- lm(b ~ a)
jackLM(LM1)
```

`lmExact` *Create random values that deliver linear regressions with exact parameters*

Description

Takes self-supplied x/y values or x/random values and transforms these as to deliver linear regressions $y = \beta_0 + \beta_1 x + \varepsilon$ (with potential replicates) with either

1) exact slope β_1 and intercept β_0,
2) exact *p*-value and intercept β_0, or
3) exact R^2 and intercept β_0.

Intended for testing and education, not for cheating ! ;-)

Usage

```
lmExact(x = 1:20, y = NULL, ny = 1, intercept = 0, slope = 0.1, error = 0.1,
        seed = NULL, pval = NULL, rsq = NULL, plot = TRUE, verbose = FALSE, ...)
```

Arguments

`x`	the predictor values.
`y`	`NULL`. A possible vector of y values with `length(x)`.
`ny`	the number of replicate response values per predictor value.
`intercept`	the desired intercept β_0.
`slope`	the desired slope β_1.
`error`	if a single value, the standard deviation σ for sampling from a normal distribution, or a user-supplied vector of length `x` with random deviates.
`seed`	optional. The random generator seed for reproducibility.
`pval`	the desired *p*-value of the slope.
`rsq`	the desired R^2.
`plot`	logical. If `TRUE`, the linear regression is plotted.
`verbose`	logical. If `TRUE`, a summary is printed to the console.
`...`	other arguments to `lm` or `plot`.

Details

For case **1)**, the `error` values are added to the exact $(x_i, \beta_0 + \beta_1 x_i)$ values, the linear model $y_i = \beta_0 + \beta_1 x_i + \varepsilon$ is fit, and the residuals $y_i - \hat{y}_i$ are re-added to $(x_i, \beta_0 + \beta_1 x_i)$.
For case **2)**, the same as in **1)** is conducted, however the slope delivering the desired *p*-value is found by an optimizing algorithm.
Finally, for case **3)**, a QR reconstruction, rescaling and refitting is conducted, using the code found

under 'References'.

If y is supplied, changes in slope, intercept and p-value will deliver the sames residuals as the linear regression through x and y. A different R^2 will change the response value structure, however.

Value

A list with the following items:

lm	the linear model of class lm.
x	the predictor values.
y	the (random) response values.
summary	the model summary for quick checking of obtained parameters.

Using both x and y will give a linear regression with the desired parameter values when refitted.

Author(s)

Andrej-Nikolai Spiess

References

For method **3**):
http://stats.stackexchange.com/questions/15011/generate-a-random-variable-with-a-defined-correlation-to-an-existing-variable.

Examples

```
## No replicates, intercept = 3, slope = 0.2, sigma = 2, n = 20.
res1 <- lmExact(x = 1:20, ny = 1, intercept = 3, slope = 2, error = 2)

## Same as above, but with 3 replicates, sigma = 1,  n = 20.
res2 <- lmExact(x = 1:20, ny = 3, intercept = 3, slope = 2, error = 1)

## No replicates, intercept = 2 and p-value = 0.025, sigma = 3, n = 50.
## => slope = 0.063
res3 <- lmExact(x = 1:50, ny = 1, intercept = 2, pval = 0.025, error = 3)

## 5 replicates, intercept = 1, R-square = 0.85, sigma = 2, n = 10.
## => slope = 0.117
res4 <- lmExact(x = 1:10, ny = 5, intercept = 1, rsq = 0.85, error = 2)

## Heteroscedastic (magnitude-dependent) noise.
error <- sapply(1:20, function(x) rnorm(3, 0, x/10))
res5 <- lmExact(x = 1:20, ny = 3, intercept = 1, slope = 0.2,
                error = error)

## Supply own x/y values, residuals are similar to an
## initial linear regression.
X <- c(1.05, 3, 5.2, 7.5, 10.2, 11.7)
```

```
set.seed(123)
Y <- 0.5 + 2 * X + rnorm(6, 0, 2)
res6 <- lmExact(x = X, y = Y, intercept = 1, slope = 0.2)
all.equal(residuals(lm(Y ~ X)), residuals(res6$lm))
```

`lmInfl` *Checks and analyzes leave-one-out (LOO) p-values and a variety of influence measures in linear regression*

Description

This function calculates leave-one-out (LOO) *p*-values for all data points and identifies those resulting in "significance reversal", i.e. in the *p*-value of the model's slope traversing the user-defined α-level. It also extends the classical influence measures from `influence.measures` with a few newer ones (*e.g*, 'Hadi's measure', 'Coefficient of determination ratio' and 'Pena's Si') within an output format where each outlier is marked when exceeding the measure's specific threshold, as defined in the literature. Belsley, Kuh & Welsch's *dfstat* criterion is also included.

Usage

```
lmInfl(model, alpha = 0.05, cutoff = c("BKW", "R"), verbose = TRUE, ...)
```

Arguments

`model`	the linear model of class `lm`.
`alpha`	the α-level to use as the threshold border.
`cutoff`	use the cutoff-values from Belsley, Kuh & Welsch or the R-internal ones. See 'Details'.
`verbose`	logical. If `TRUE`, results are displayed on the console.
`...`	other arguments to `lm`.

Details

The algorithm
1) calculates the *p*-value of the full model (all points),
2) calculates a LOO-*p*-value for each point removed,
3) checks for significance reversal in all data points and
4) returns all models as well as classical `influence.measures` with LOO-*p*-values, Δp-values, slopes and standard errors attached.

The idea of *p*-value influencers was first introduced by Belsley, Kuh & Welsch, and described as an influence measure pertaining directly to the change in *t*-statistics, that will "show whether the conclusions of hypothesis testing would be affected", termed **dfstat** in [1, 2, 3] or **dfstud** in [4]:

$$\mathrm{dfstat}_{ij} \equiv \frac{\hat{\beta}_j}{s\sqrt{(X'X)^{-1}_{jj}}} - \frac{\hat{\beta}_{j(i)}}{s_{(i)}\sqrt{(X'_{(i)}X_{(i)})^{-1}_{jj}}}$$

where $\hat{\beta}_j$ is the *j*-th estimate, *s* is the residual standard error, *X* is the design matrix and (*i*) denotes the *i*-th observation deleted.
dfstat, which for the regression's slope β_1 is the difference of *t*-statistics

$$\Delta t = t_{\beta 1} - t_{\beta 1(i)} = \frac{\beta_1}{\text{s.e.}(\beta_1)} - \frac{\beta_1(i)}{\text{s.e.}(\beta_1(\text{i}))}$$

is inextricably linked to the changes in *p*-value Δp, calculated from

$$\Delta p = p_{\beta 1} - p_{\beta 1(i)} = 2\left(1 - P_t(t_{\beta 1}, \nu)\right) - 2\left(1 - P_t(t_{\beta 1(i)}, \nu - 1)\right)$$

where P_t is the Student's *t* cumulative distribution function with ν degrees of freedom, and where significance reversal is attained when $\alpha \in [p_{\beta 1}, p_{\beta 1(i)}]$. Interestingly, the seemingly mandatory check of the influence of single data points on statistical inference is living in oblivion: apart from [1-4], there is, to the best of our knowledge, no reference to **dfstat** or Δp in current literature on influence measures.

Cut-off values for the different influence measures are per default (`cutoff = "BKW"`) those defined in Belsley, Kuh & Welsch (1980) and additional literature.

dfbeta slope: $|\Delta\beta 1_i| > 2/\sqrt{n}$ (page 28)
dffits: $|\text{dffits}_i| > 2\sqrt{2/n}$ (page 28)
covratio: $|\text{covr}_i - 1| > 3k/n$ (page 23)
Cook's D: $D_i > Q_F(0.5, k, n-k)$ (Cook & Weisberg, 1982)
leverage: $h_{ii} > 2k/n$ (page 17)
studentized residual: $t_i > Q_t(0.975, n-k-1)$ (page 20)

If (`cutoff = "R"`), the criteria from `influence.measures` are employed:

dfbeta slope: $|\Delta\beta 1_i| > 1$
dffits: $|\text{dffits}_i| > 3\sqrt{(k/(n-k))}$
covratio: $|1 - \text{covr}_i| > 3k/(n-k)$
Cook's D: $D_i > Q_F(0.5, k, n-k)$
leverage: $h_{ii} > 3k/n$

The influence output also includes the following more "recent" measures:
Hadi's measure (column "hadi"):

$$H_i^2 = \frac{h_{ii}}{1-h_{ii}} + \frac{p}{1-h_{ii}}\frac{d_i^2}{(1-d_i^2)}$$

where h_{ii} are the diagonals of the hat matrix (leverages), $p = 2$ in univariate linear regression and $d_i = e_i/\sqrt{\text{SSE}}$, and threshold value $\text{Med}(H_i^2) + 2 \cdot \text{MAD}(H_i^2)$.
Coefficient of Determination Ratio (column "cdr"):

$$\text{CDR}_i = \frac{R^2_{(i)}}{R^2}$$

with $R^2_{(i)}$ being the coefficient of determination without value *i*, and threshold

$$\frac{B_{\alpha,p/2,(n-p-2)/2}}{B_{\alpha,p/2,(n-p-1)/2}}$$

Pena's Si (column "Si"):

$$S_i = \frac{\mathbf{s}_i' \mathbf{s}_i}{p\widehat{\mathrm{var}}(\hat{y}_i)}$$

where $\mathbf{s_i}$ is the vector of each fitted value from the original model, $\hat{y}_i$, subtracted with all fitted values after 1-deletion, $\hat{y}_i - \hat{y}_{i(-1)}, \cdots, \hat{y}_i - \hat{y}_{i(-n)}$, p = number of parameters, and $\widehat{\mathrm{var}}(\hat{y}_i) = s^2 h_{ii}$, $s^2 = (\mathbf{e}'\mathbf{e})/(n-p)$, $\mathbf{e}$ being the residuals. In this package, a cutoff value of 0.9 is used, as the published criterion of $|\mathbf{S_i} - \mathrm{Med}(\mathbf{S})| \geq 4.5\mathrm{MAD}(\mathbf{S})$ seemed too conservative. Results from this function were verified by Prof. Daniel Pena through personal communication.

Value

A list with the following items:

`origModel`	the original model with all data points.
`finalModels`	a list of final models with the influencer(s) removed.
`infl`	a matrix with the original data, classical `influence.measures`, studentized residuals, leverages, dfstat, LOO-p-values, LOO-slopes/intercepts and their Δ's, LOO-standard errors and R^2s. Influence measures that exceed their specific threshold - see `inflPlot` - will be marked with asterisks.
`raw`	same as `infl`, but with pure numeric data.
`sel`	a vector with the influencers' indices.
`alpha`	the selected α-level.
`origP`	the original model's p-value.

Author(s)

Andrej-Nikolai Spiess

References

For dfstat / dfstud :
Regression diagnostics: Identifying influential data and sources of collinearity.
Belsley DA, Kuh E, Welsch RE.
John Wiley, New York, USA (2004).

Econometrics, 5ed.
Baltagi B.
Springer-Verlag Berlin, Germany (2011).

Growth regressions and what the textbooks don't tell you.
Temple J.
Bull Econom Res, **52**, 2000, 181-205.

Robust Regression and Outlier Detection.
Rousseeuw PJ & Leroy AM.
John Wiley & Sons, New York, NY (1987).

Hadi's measure:
A new measure of overall potential influence in linear regression.
Hadi AS.
Comp Stat & Data Anal, **14**, 1992, 1-27.

Coefficient of determination ratio:
On the detection of influential outliers in linear regression analysis.
Zakaria A, Howard NK, Nkansah BK.
Am J Theor Appl Stat, **3**, 2014, 100-106.

On the Coefficient of Determination Ratio for Detecting Influential Outliers in Linear Regression Analysis.
Zakaria A, Gordor BK, Nkansah BK.
Am J Theor Appl Stat, **11**, 2022, 27-35.

Pena's measure:
A New Statistic for Influence in Linear Regression.
Pena D.
Technometrics, **47**, 2005, 1-12.

Examples

```
## Example #1 with single influencer and significant model (p = 0.0089).
## Removal of #21 results in p = 0.115!
set.seed(123)
a <- 1:20
b <- 5 + 0.08 * a + rnorm(20, 0, 1)
a <- c(a, 25); b <- c(b, 10)
LM1 <- lm(b ~ a)
lmInfl(LM1)

## Example #2 with single influencer and insignificant model (p = 0.115).
## Removal of #18 results in p = 0.0227!
set.seed(123)
a <- 1:20
b <- 5 + 0.08 * a + rnorm(20, 0, 1)
LM2 <- lm(b ~ a)
lmInfl(LM2)

## Example #3 with multiple influencers and significant model (p = 0.0269).
## Removal of #2, #17, #18 or #20 results in crossing p = 0.05!
set.seed(125)
a <- 1:20
b <- 5 + 0.08 * a + rnorm(20, 0, 1)
LM3 <- lm(b ~ a)
lmInfl(LM3)

## Large Example #4 with top 10 influencers and significant model (p = 6.72E-8).
## Not possible to achieve a crossing of alpha with any point despite strong noise.
set.seed(123)
```

```
a <- 1:100
b <- 5 + 0.08 * a + rnorm(100, 0, 5)
LM4 <- lm(b ~ a)
lmInfl(LM4)
```

---

pcomp *Calculates linear regression p-values from a variety of robust regression methods*

Description

This function calculates *p*-values from a variety of methods, specifically:
1) standard linear model
2) standard linear model with highest *p*-influencer removed
3) robust regression with MM-estimators
4) Theil-Sen regression
5) least absolute deviations regression
6) quantile regression
7) weighted regression with isolation forest scores as inverse weights
8) bootstrap linear model, see bootLM
9) jackknife linear model, see jackLM

Usage

```
pcomp(x, y = NULL, R = 1000, alpha = 0.05, ...)
```

Arguments

x	either a linear model of class lm or the regressions *x*-values.
y	the optional *y*-values.
R	the number of bootstrap resamples, see bootLM.
alpha	the α-level for lmInfl.
...	further arguments to be passed to downstream methods.

Details

This function is meant to provide a swift overview on the sensitivity of the *p*-values to different (mostly robust) linear regression methods, which correlates to a large extent with the presence of influential / outlying data points, see 'Examples'.

Value

A vector of *p*-values from the above mentioned ten methods, in that order.

Author(s)

Andrej-Nikolai Spiess

References

Robust Regression and Outlier Detection.
Rousseeuw PJ & Leroy AM.
1ed (1987), Wiley (NJ, USA).

A rank-invariant method of linear and polynomial regression analysis.
Theil H.
I. Nederl. Akad. Wetensch. Proc, **53**, 1950, 386-392.

Estimates of the regression coefficient based on Kendall's tau.
Sen PK.
J Am Stat Assoc, **63**, 1968, 1379-1389.

Least absolute deviations estimation via the EM algorithm.
Phillips RF.
Statistics and Computing, **12**, 2002, 281-285.

Quantile Regression.
Koenker R.
Cambridge University Press, Cambridge, New York (2005).

Isolation-based anomaly detection.
Liu FT, Ting KM, Zhou ZH.
ACM Transactions on Knowledge Discovery from Data, **6.1**, 2012, 3.

Examples

```
## Example with influencer
## => a few methods indicate significant
## downward drop of the p-value
set.seed(123)
a <- 1:20
b <- 5 + 0.08 * a + rnorm(20, 0, 1)
pcomp(a, b)
```

PNAS2015 *Small dataset from a 2015 PNAS paper*

Description

The data was acquired by digitization of a graph from a 2015 PNAS paper. Contains three datapoints that exert significance reversal.

Usage

```
data(PNAS2015)
```

Author(s)

Andrej-Nikolai Spiess

Examples

```
## See examples in 'lmInfl' and 'lmThresh'.
LM <- lm(y ~ x, data = PNAS2015)
lmInfl(LM)
```

`regionInfl` *Identify regions of significance reversal and influence measure threshold*

Description

Identifies regions of an (univariate) linear model in which a future data point would result in either
a) significance reversal, or
b) any selected influence measure as given in `crit` exceed its threshold value.
This is intended mainly for visual/didactical purposes.

Usage

```
regionInfl(model, div.x = 20, div.y = 20, grid = TRUE, pred.int = TRUE,
       crit = c("P", "dfb.Slope", "dffit", "cov.r", "cook.d", "hat", "hadi",
    "sR", "cdr", "Si"), cex.grid = 0.5, alpha = 0.05, xlim = NULL, ylim = NULL, ...)
```

Arguments

`model`	the linear model of class `lm`.
`div.x`	the number of grid division for the x-axis.
`div.y`	the number of grid division for the y-axis.
`grid`	logical. Show the grid lines on the plot or not.
`pred.int`	logical. Show the 95% prediction interval on the plot or not.
`crit`	the criterion to use. Either `"P"` for significance reversal or any of the influence measures given there.
`cex.grid`	size of the grid points.
`alpha`	the α-level to be set as threshold.
`xlim`	similar to `xlim`, a 2-element vector for the x-axis limits, overrides `fac.x`.
`ylim`	similar to `ylim`, a 2-element vector for the y-axis limits, overrides `fac.y`.
`...`	other parameters to be supplied to `plot` or `lmInfl`.

Details

For a given linear model $y_i = \beta_0 + \beta_1 x_i + \varepsilon$, each (a, b) pair from a grid of values $(a_1 \dots a_j, b_1 \dots b_k)$ is added to the data, and an updated model $(y_i, b_k) = \beta_0 + \beta_1(x_i, a_j) + \varepsilon$ is created. If the updated model's $p \leq \alpha$ or any of the influence measures does not exceed its published threshold, it is plotted in green, otherwise in orange. If `outlier = TRUE`, a possible reverser is eliminated prior to analysis but visualized in the plot.

Value

A plot with the regions marked in orange or green, and the grid matrix (`grid`) including the criterion outcome in 1 (green) or 0 (orange).

Author(s)

Andrej-Nikolai Spiess

Examples

```
## Model with p = 0.014
set.seed(7)
N <- 20
x <- runif(N, 1, 100)
y <- 0.05 * x + rnorm(N, 0, 2)
LM1 <- lm(y ~ x)
summary(LM1)
regionInfl(LM1, crit = "P", div.x = 20, div.y = 20, cex.grid = 1,
           xlim = c(-20, 120), ylim = c(-5, 10))
```

`rpLM` *Calculates the 'replication probability of significance' of an 'lm' object*

Description

This function uses a bootstrap approach to calculate the *replication probability* of significance, which answers the question "if we repeat this linear regression under identical conditions (similar sample size, similar residual variance), what is the probability of observing significance (or non-significance) similar to the original data?".

Usage

```
rpLM(model, alpha = 0.05, R = 10000, plot = TRUE, verbose = TRUE, ...)
```

Arguments

`model`	a linear model of class `lm`.
`alpha`	the α-level to use as the threshold border.
`R`	the number of bootstrap resamples, see `bootLM`.
`plot`	logical. If `TRUE`, a stripchart of the bootstrap *P*-values, the original *P*-value and the α-level is displayed.
`verbose`	logical. If `TRUE`, the analysis steps are written to the console.
`...`	other parameters to be supplied to `bootLM`.

Details

The approach here is along the lines of Boos & Stefanski (2011), which investigated the replication probability of the *P*-value, as opposed to the works of Goodman (1992), Shao & Chow (2002) and Miller (2009), where the effect size is used. In our context, for a given linear model and using a bootstrap approach, the *replication probability* is the proportion of bootstrap *P*-values with $\tilde{P} \leq \alpha$ when the original model is significant, or $\tilde{P} > \alpha$ when not. Hence, we employ the bootstrap to assess the sampling variability of the *P*-value, not the sampling variability of the *P*-value under H_0, as is common, thereby preserving the non-null property of the data.

Bootstrap results are obtained from non-parametric cases bootstrapping ("np.cases"), non-parametric residuals bootstrapping ("np.resid") and parametric residuals bootstrapping ("p.resid"), see bootLM.

Value

A vector with the three different bootstrap results as described above.

Author(s)

Andrej-Nikolai Spiess

References

Ecological Models and Data in R.
Chapter 5: Stochastic simulation and power analysis.
Benjamin M. Bolker.
Princeton University Press (2008).

P-Value Precision and Reproducibility.
Boos DD & Stefanski LA.
Am Stat, **65**, 2011, 213-212.

A comment on replication, p-values and evidence.
Goodman SN.
Stat Med, **11**, 1992, 875-879.

Reproducibility probability in clinical trials.
Shao J & Chow SC.
Stat Med, **21**, 2002, 1727-1742.

What is the probability of replicating a statistically significant effect?
Miller J.
Psych Bull & Review, **16**, 2009, 617-640.

Examples

```
set.seed(125)
a <- 1:20
b <- 5 + 0.08 * a + rnorm(length(a), 0, 1)
LM1 <- lm(b ~ a)
summary(LM1)
rpLM(LM1, R = 100)
```

Index

∗ **linear**
  bootLM, 2
  Influence plots, 3
  jackLM, 5
  lmExact, 6
  lmInfl, 8
  pcomp, 12
  PNAS2015, 13
  regionInfl, 14
  rpLM, 15
∗ **models**
  bootLM, 2
  Influence plots, 3
  jackLM, 5
  lmExact, 6
  lmInfl, 8
  pcomp, 12
  PNAS2015, 13
  regionInfl, 14
  rpLM, 15
∗ **optimize**
  bootLM, 2
  Influence plots, 3
  jackLM, 5
  lmExact, 6
  lmInfl, 8
  pcomp, 12
  PNAS2015, 13
  regionInfl, 14
  rpLM, 15

bootLM, 2, *12*, *15, 16*

inflPlot, *10*
inflPlot *(*Influence plots*)*, 3
Influence plots, 3
influence.measures, *8–10*

jackLM, 5, *12*

lm, *2*, *5, 6*, *8*, *12*, *14, 15*
lmExact, 6
lmInfl, *3*, *4*, 8, *12*, *14*

pcomp, 12
plot, *6*, *14*
PNAS2015, 13
pvalPlot *(*Influence plots*)*, 3

regionInfl, 14
rpLM, 15

xlim, *14*

ylim, *14*