Compute Optimal Block Length for Stationary and Circular Bootstrap

Description

`b.star` is a function which computes the optimal block length for a continuous variable `data` using the method described in Patton, Politis, and White (2009).

Usage

```
b.star(data,
       Kn = NULL,
       mmax= NULL,
       Bmax = NULL,
       c = NULL,
       round = FALSE)
```

Arguments

`data`	data, an n x k matrix, each column being a data series.
`Kn`	See footnote c, page 59, Politis and White (2004). Defaults to `ceiling(log10(n))`.
`mmax`	See Politis and White (2004). Defaults to `ceiling(sqrt(n))+Kn`.
`Bmax`	See Politis and White (2004). Defaults to `ceiling(min(3*sqrt(n),n/3))`.
`c`	See Politis and White (2004). Defaults to `qnorm(0.975)`.
`round`	whether to round the result or not. Defaults to FALSE.

Details

`b.star` is a function which computes optimal block lengths for the stationary and circular bootstraps. This allows the use of `tsboot` from the boot package to be fully automatic by using the output from `b.star` as an input to the argument `l =` in `tsboot`. See below for an example.

Value

A kx2 matrix of optimal bootstrap block lengths computed from `data` for the stationary bootstrap and circular bootstrap (column 1 is for the stationary bootstrap, column 2 the circular).

Author(s)

Tristen Hayfield hayfield@phys.ethz.ch, Jeffrey S. Racine racinej@mcmaster.ca

References

Patton, A. and D. N. Politis and H. White (2009), “CORRECTION TO "Automatic block-length selection for the dependent bootstrap" by D. Politis and H. White”, Econometric Reviews 28(4), 372-375.

Politis, D. N. and J. P. Romano (1994), “Limit theorems for weakly dependent Hilbert space valued random variables with applications to the stationary bootstrap”, Statistica Sinica 4, 461-476.

Politis, D. N. and H. White (2004), “Automatic block-length selection for the dependent bootstrap”, Econometric Reviews 23(1), 53-70.

Examples

```
set.seed(12345)

# Function to generate an AR(1) series

ar.series <- function(phi,epsilon) {
  n <- length(epsilon)
  series <- numeric(n)
  series[1] <- epsilon[1]/(1-phi)
  for(i in 2:n) {
    series[i] <- phi*series[i-1] + epsilon[i]
  }
  return(series)
}

yt <- ar.series(0.1,rnorm(10000))
b.star(yt,round=TRUE)

yt <- ar.series(0.9,rnorm(10000))
b.star(yt,round=TRUE)
```
