Package 'bootUR'

July 24, 2020

Type Package

Title Bootstrap Unit Root Tests

Version 0.2.0

Maintainer Stephan Smeekes <s.smeekes@maastrichtuniversity.nl>

Description Set of functions to perform various bootstrap unit root tests for both individual time series (including augmented Dickey-Fuller test and union tests), multiple time series and panel data; see Palm, Smeekes and Urbain (2008) <doi:10.1111/j.1467-9892.2007.00565.x>, Palm, Smeekes and Urbain (2011) <doi:10.1016/j.jeconom.2010.11.010>, Moon and Perron (2012) <doi:10.1016/j.jeconom.2012.01.008>, Smeekes and Taylor (2012) <doi:10.1017/S0266466611000387> and Smeekes (2015) <doi:10.1111/jtsa.12110> for key references.

Depends R (>= 3.5.0)

License GPL (>= 2)

Encoding UTF-8

LazyData true

RoxygenNote 7.1.0

Imports Rcpp, stats

LinkingTo Rcpp, RcppArmadillo, RcppProgress

URL https://github.com/smeekes/bootUR

Suggests knitr, rmarkdown, testthat, ggplot2

VignetteBuilder knitr

NeedsCompilation yes

Author Stephan Smeekes [cre, aut] (<https://orcid.org/0000-0002-0157-639X>), Ines Wilms [aut]

Repository CRAN

Date/Publication 2020-07-24 14:20:02 UTC

R topics documented:

`bFDRtest` *Bootstrap Unit Root Tests with False Discovery Rate control*

Description

Controls for multiple testing by controlling the false discovery rate (FDR), see Moon and Perron (2012) and Romano, Shaikh and Wolf (2008).

Usage

```
bFDRtest(y, level = 0.05, boot = "AWB", B = 1999, l = NULL,
  ar_AWB = NULL, union = TRUE, p_min = 0, p_max = NULL, ic = "MAIC",
  dc = NULL, detr = NULL, ic_scale = TRUE, verbose = FALSE,
  show_progress = FALSE, do_parallel = FALSE, nc = NULL)
```

Arguments

`y`	A T-dimensional vector or a (T x N)-matrix of N time series with T observations to be tested for unit roots. Data may also be in a time series format (e.g. `ts`, `zoo` or `xts`), or a data frame, as long as each column represents a single time series.
`level`	Desired False Discovery Rate level of the unit root tests. Default is 0.05.
`boot`	String for bootstrap method to be used. Options are `"MBB"` Moving blocks bootstrap (Paparoditis and Politis, 2003; Palm, Smeekes and Urbain, 2011); `"BWB"` Block wild bootstrap (Shao, 2011; Smeekes and Urbain, 2014a); `"DWB"` Dependent wild bootstrap (Shao, 2010; Smeekes and Urbain, 2014a; Rho and Shao, 2019);

- "AWB" Autoregressive wild bootstrap (Smeekes and Urbain, 2014a; Friedrich, Smeekes and Urbain, 2020), this is the default;
- "SB" Sieve bootstrap (Chang and Park, 2003; Palm, Smeekes and Urbain, 2008; Smeekes, 2013);
- "SWB" Sieve wild boostrap (Cavaliere and Taylor, 2009; Smeekes and Taylor, 2012).

B
: Number of bootstrap replications. Default is 1999.

l
: Desired 'block length' in the bootstrap. For the MBB, BWB and DWB boostrap, this is a genuine block length. For the AWB boostrap, the block length is transformed into an autoregressive parameter via the formula $0.01^(1/l)$ as in Smeekes and Urbain (2014a); this can be overwritten by setting ar_AWB directly. Default sets the block length as a function of the time series length T, via the rule $l = 1.75T^(1/3)$ of Palm, Smeekes and Urbain (2011).

ar_AWB
: Autoregressive parameter used in the AWB bootstrap method (boot = "AWB"). Can be used to set the parameter directly rather than via the default link to the block length l.

union
: Logical indicator whether or not to use bootstrap union tests (TRUE) or not (FALSE), see Smeekes and Taylor (2012). Default is TRUE.

p_min
: Minimum lag length in the augmented Dickey-Fuller regression. Default is 0.

p_max
: Maximum lag length in the augmented Dickey-Fuller regression. Default uses the sample size-based rule $12(T/100)^{1/4}$.

ic
: String for information criterion used to select the lag length in the augmented Dickey-Fuller regression. Options are: "AIC", "BIC", "MAIC", "MBIC". Default is "MAIC" (Ng and Perron, 2001).

dc
: Numeric vector indicating the deterministic specification. Only relevant if union = FALSE. Options are (combinations of)
0 no deterministics;
1 intercept only;
2 intercept and trend.
If union = FALSE, the default is adding an intercept (a warning is given).

detr
: String vector indicating the type of detrending to be performed. Only relevant if union = FALSE. Options are: "OLS" and/or "QD" (typically also called GLS, see Elliott, Rothenberg and Stock, 1996). The default is "OLS".

ic_scale
: Logical indicator whether or not to use the rescaled information criteria of Cavaliere et al. (2015) (TRUE) or not (FALSE). Default is TRUE.

verbose
: Logical indicator whether or not information on the outcome of the unit root test needs to be printed to the console. Default is FALSE.

show_progress
: Logical indicator whether a bootstrap progress update should be printed to the console. Default is FALSE.

do_parallel
: Logical indicator whether bootstrap loop should be executed in parallel. Parallel computing is only available if OpenMP can be used, if not this option is ignored. Default is FALSE.

nc
: The number of cores to be used in the parallel loops. Default is to use all but one.

Details

The false discovery rate FDR is defined as the expected proportion of false rejections relative to the total number of rejections.

See `iADFtest` for details on the bootstrap algorithm and lag selection.

Value

A list with the following components

`rej_H0`	Logical indicator whether the null hypothesis of a unit root is rejected (`TRUE`) or not (`FALSE`);
`FDR_sequence`	Details on the unit root tests: value of the test statistics and critical values.

For the union test (`union = TRUE`), the output is arranged per time series. If `union = FALSE`, the output is arranged per time series, type of deterministic component (`dc`) and detrending method (`detr`).

Errors and warnings

`Error: Resampling-based bootstraps MBB and SB cannot handle missing values.` If the time series in `y` have different starting and end points (and thus some series contain `NA` values at the beginning and/or end of the sample, the resampling-based moving block bootstrap (MBB) and sieve bootstrap (SB) cannot be used, as they create holes (internal missings) in the bootstrap samples. Switch to another bootstrap method or truncate your sample to eliminate `NA` values.

`Warning: SB and SWB bootstrap only recommended for iADFtest; see help for details.` Although the sieve bootstrap methods `"SB"` and `"SWB"` can be used, Smeekes and Urbain (2014b) show that these are not suited to capture general forms of dependence across units, and using them for joint or multiple testing is not valid. This warning thereofre serves to recommend the user to consider a different bootstrap method.

`Warning: Deterministic specification in argument dc is ignored, as union test is applied.` The union test calculates the union of all four combinations of deterministic components (intercept or intercept and trend) and detrending methods (OLS or QD). Setting deterministic components manually therefore has no effect.

`Warning: Detrending method in argument detr is ignored, as union test is applied.` The union test calculates the union of all four combinations of deterministic components (intercept or intercept and trend) and detrending methods (OLS or QD). Setting detrending methods manually therefore has no effect.

References

Chang, Y. and Park, J. (2003). A sieve bootstrap for the test of a unit root. *Journal of Time Series Analysis*, 24(4), 379-400.

Cavaliere, G. and Taylor, A.M.R (2009). Heteroskedastic time series with a unit root. *Econometric Theory*, 25, 1228–1276.

Cavaliere, G., Phillips, P.C.B., Smeekes, S., and Taylor, A.M.R. (2015). Lag length selection for unit root tests in the presence of nonstationary volatility. *Econometric Reviews*, 34(4), 512-536.

Elliott, G., Rothenberg, T.J., and Stock, J.H. (1996). Efficient tests for an autoregressive unit root. *Econometrica*, 64(4), 813-836.

Friedrich, M., Smeekes, S. and Urbain, J.-P. (2020). Autoregressive wild bootstrap inference for nonparametric trends. *Journal of Econometrics*, 214(1), 81-109.

Moon, H.R. and Perron, B. (2012). Beyond panel unit root tests: Using multiple testing to determine the non stationarity properties of individual series in a panel. Journal of Econometrics, 169(1), 29-33.

Ng, S. and Perron, P. (2001). Lag Length Selection and the Construction of Unit Root Tests with Good Size and Power. *Econometrica*, 69(6), 1519-1554,

Palm, F.C., Smeekes, S. and Urbain, J.-P. (2008). Bootstrap unit root tests: Comparison and extensions. *Journal of Time Series Analysis*, 29(1), 371-401.

Palm, F. C., Smeekes, S., and Urbain, J.-.P. (2011). Cross-sectional dependence robust block bootstrap panel unit root tests. *Journal of Econometrics*, 163(1), 85-104.

Paparoditis, E. and Politis, D.N. (2003). Residual-based block bootstrap for unit root testing. *Econometrica*, 71(3), 813-855.

Perron, P. and Qu, Z. (2008). A simple modification to improve the finite sample properties of Ng and Perron's unit root tests. *Economic Letters*, 94(1), 12-19.

Rho, Y. and Shao, X. (2019). Bootstrap-assisted unit root testing with piecewise locally stationary errors. *Econometric Theory*, 35(1), 142-166.

Romano, J.P., Shaikh, A.M., and Wolf, M. (2008). Control of the false discovery rate under dependence using the bootstrap and subsampling. *Test*, 17(3), 417.

Shao, X. (2010). The dependent wild bootstrap. *Journal of the American Statistical Association*, 105(489), 218-235.

Shao, X. (2011). A bootstrap-assisted spectral test of white noise under unknown dependence. *Journal of Econometrics*, 162, 213-224.

Smeekes, S. (2013). Detrending bootstrap unit root tests. *Econometric Reviews*, 32(8), 869-891.

Smeekes, S. and Taylor, A.M.R. (2012). Bootstrap union tests for unit roots in the presence of nonstationary volatility. *Econometric Theory*, 28(2), 422-456.

Smeekes, S. and Urbain, J.-P. (2014a). A multivariate invariance principle for modified wild bootstrap methods with an application to unit root testing. GSBE Research Memorandum No. RM/14/008, Maastricht University

Smeekes, S. and Urbain, J.-P. (2014b). On the applicability of the sieve bootstrap in time series panels. *Oxford Bulletin of Economics and Statistics*, 76(1), 139-151.

See Also

`iADFtest`

Examples

```
# bFDRtest on GDP_BE and GDP_DE
two_series_bFDRtest <- bFDRtest(MacroTS[, 1:2], boot = "MBB", B = 399,  verbose = TRUE)
```

boot_df *Bootstrap augmented Dickey-Fuller Unit Root Test*

Description

This function performs a standard augmented Dickey-Fuller bootstrap unit root test on a single time series.

Usage

```
boot_df(y, level = 0.05, boot = "AWB", B = 1999, l = NULL,
  ar_AWB = NULL, p_min = 0, p_max = NULL, ic = "MAIC", dc = 1,
  detr = "OLS", ic_scale = TRUE, verbose = FALSE,
  show_progress = FALSE, do_parallel = FALSE, nc = NULL)
```

Arguments

`y` A T-dimensional vector to be tested for unit roots. Data may also be in a time series format (e.g. `ts`, `zoo` or `xts`), or a data frame.

`level` Desired significance level of the unit root test. Default is 0.05.

`boot` String for bootstrap method to be used. Options are

- `"MBB"` Moving blocks bootstrap (Paparoditis and Politis, 2003);
- `"BWB"` Block wild bootstrap (Shao, 2011);
- `"DWB"` Dependent wild bootstrap (Shao, 2010; Rho and Shao, 2019);
- `"AWB"` Autoregressive wild bootstrap (Smeekes and Urbain, 2014a; Friedrich, Smeekes and Urbain, 2020), this is the default;
- `"SB"` Sieve bootstrap (Chang and Park, 2003; Palm, Smeekes and Urbain, 2008; Smeekes, 2013);
- `"SWB"` Sieve wild boostrap (Cavaliere and Taylor, 2009; Smeekes and Taylor, 2012).

`B` Number of bootstrap replications. Default is 1999.

`l` Desired 'block length' in the bootstrap. For the MBB, BWB and DWB boostrap, this is a genuine block length. For the AWB boostrap, the block length is transformed into an autoregressive parameter via the formula $0.01^(1/l)$ as in Smeekes and Urbain (2014a); this can be overwritten by setting `ar_AWB` directly. Default sets the block length as a function of the time series length T, via the rule $l = 1.75T^(1/3)$ of Palm, Smeekes and Urbain (2011).

`ar_AWB` Autoregressive parameter used in the AWB bootstrap method (`boot = "AWB"`). Can be used to set the parameter directly rather than via the default link to the block length l.

`p_min` Minimum lag length in the augmented Dickey-Fuller regression. Default is 0.

`p_max` Maximum lag length in the augmented Dickey-Fuller regression. Default uses the sample size-based rule $12(T/100)^{1/4}$.

`ic` String for information criterion used to select the lag length in the augmented Dickey-Fuller regression. Options are: "AIC", "BIC", "MAIC", "MBIC". Default is "MAIC" (Ng and Perron, 2001).

`dc` Numeric vector indicating the deterministic specification. Only relevant if union = FALSE. Options are (combinations of)
0 no deterministics;
1 intercept only;
2 intercept and trend.
If union = FALSE, the default is adding an intercept (a warning is given).

`detr` String vector indicating the type of detrending to be performed. Only relevant if union = FALSE. Options are: "OLS" and/or "QD" (typically also called GLS, see Elliott, Rothenberg and Stock, 1996). The default is "OLS".

`ic_scale` Logical indicator whether or not to use the rescaled information criteria of Cavaliere et al. (2015) (TRUE) or not (FALSE). Default is TRUE.

`verbose` Logical indicator whether or not information on the outcome of the unit root test needs to be printed to the console. Default is FALSE.

`show_progress` Logical indicator whether a bootstrap progress update should be printed to the console. Default is FALSE.

`do_parallel` Logical indicator whether bootstrap loop should be executed in parallel. Parallel computing is only available if OpenMP can be used, if not this option is ignored. Default is FALSE.

`nc` The number of cores to be used in the parallel loops. Default is to use all but one.

Details

The options encompass many test proposed in the literature. dc = "OLS" gives the standard augmented Dickey-Fuller test, while dc = "QD" provides the DF-GLS test of Elliott, Rothenberg and Stock (1996). The bootstrap algorithm is always based on a residual bootstrap (under the alternative) to obtain residuals rather than a difference-based bootstrap (under the null), see e.g. Palm, Smeekes and Urbain (2008).

Lag length selection is done automatically in the ADF regression with the specified information criterion. If one of the modified criteria of Ng and Perron (2001) is used, the correction of Perron and Qu (2008) is applied. For very short time series (fewer than 50 time points) the maximum lag length is adjusted downward to avoid potential multicollinearity issues in the bootstrap. To overwrite data-driven lag length selection with a pre-specified lag length, simply set both the minimum ‘p_min‘ and maximum lag length ‘p_max‘ for the selection algorithm equal to the desired lag length.

Value

Values of the Dickey-Fuller test statistics and corresponding bootstrap p-values.

Errors and warnings

`Error: Multiple time series not allowed. Switch to a multivariate method such as iADFtest, or change argume`
The function is a simple wrapper around iADFtest to facilitate use for single time series. It does not support multiple time series, as iADFtest is specifically suited for that.

References

Chang, Y. and Park, J. (2003). A sieve bootstrap for the test of a unit root. *Journal of Time Series Analysis*, 24(4), 379-400.

Cavaliere, G. and Taylor, A.M.R (2009). Heteroskedastic time series with a unit root. *Econometric Theory*, 25, 1228–1276.

Cavaliere, G., Phillips, P.C.B., Smeekes, S., and Taylor, A.M.R. (2015). Lag length selection for unit root tests in the presence of nonstationary volatility. *Econometric Reviews*, 34(4), 512-536.

Elliott, G., Rothenberg, T.J., and Stock, J.H. (1996). Efficient tests for an autoregressive unit root. *Econometrica*, 64(4), 813-836.

Friedrich, M., Smeekes, S. and Urbain, J.-P. (2020). Autoregressive wild bootstrap inference for nonparametric trends. *Journal of Econometrics*, 214(1), 81-109.

Ng, S. and Perron, P. (2001). Lag Length Selection and the Construction of Unit Root Tests with Good Size and Power. *Econometrica*, 69(6), 1519-1554,

Palm, F.C., Smeekes, S. and Urbain, J.-P. (2008). Bootstrap unit root tests: Comparison and extensions. *Journal of Time Series Analysis*, 29(1), 371-401.

Paparoditis, E. and Politis, D.N. (2003). Residual-based block bootstrap for unit root testing. *Econometrica*, 71(3), 813-855.

Perron, P. and Qu, Z. (2008). A simple modification to improve the finite sample properties of Ng and Perron's unit root tests. *Economic Letters*, 94(1), 12-19.

Rho, Y. and Shao, X. (2019). Bootstrap-assisted unit root testing with piecewise locally stationary errors. *Econometric Theory*, 35(1), 142-166.

Smeekes, S. (2013). Detrending bootstrap unit root tests. *Econometric Reviews*, 32(8), 869-891.

Shao, X. (2010). The dependent wild bootstrap. *Journal of the American Statistical Association*, 105(489), 218-235.

Shao, X. (2011). A bootstrap-assisted spectral test of white noise under unknown dependence. *Journal of Econometrics*, 162, 213-224.

Smeekes, S. and Taylor, A.M.R. (2012). Bootstrap union tests for unit roots in the presence of nonstationary volatility. *Econometric Theory*, 28(2), 422-456.

Smeekes, S. and Urbain, J.-P. (2014a). A multivariate invariance principle for modified wild bootstrap methods with an application to unit root testing. GSBE Research Memorandum No. RM/14/008, Maastricht University

See Also

iADFtest

Examples

```
# boot_df on GDP_BE
GDP_BE_df <- boot_df(MacroTS[, 1], B = 399, dc = 2, detr = "OLS", verbose = TRUE)
```

boot_union *Bootstrap Union Test for Unit Roots*

Description

Performs bootstrap unit root test based on the union of rejections of 4 tests with different number of deterministic components and different type of detrending (Harvey, Leybourne and Taylor, 2012; Smeekes and Taylor, 2012).

Usage

```
boot_union(y, level = 0.05, boot = "AWB", B = 1999, l = NULL,
  ar_AWB = NULL, p_min = 0, p_max = NULL, ic = "MAIC",
  ic_scale = TRUE, verbose = FALSE, show_progress = FALSE,
  do_parallel = FALSE, nc = NULL)
```

Arguments

`y` A T-dimensional vector to be tested for unit roots. Data may also be in a time series format (e.g. `ts`, `zoo` or `xts`), or a data frame.

`level` Desired significance level of the unit root test. Default is 0.05.

`boot` String for bootstrap method to be used. Options are

- `"MBB"` Moving blocks bootstrap (Paparoditis and Politis, 2003);
- `"BWB"` Block wild bootstrap (Shao, 2011);
- `"DWB"` Dependent wild bootstrap (Shao, 2010; Rho and Shao, 2019);
- `"AWB"` Autoregressive wild bootstrap (Smeekes and Urbain, 2014a; Friedrich, Smeekes and Urbain, 2020), this is the default;
- `"SB"` Sieve bootstrap (Palm, Smeekes and Urbain, 2008);
- `"SWB"` Sieve wild boostrap (Cavaliere and Taylor, 2009; Smeekes and Taylor, 2012).

`B` Number of bootstrap replications. Default is 1999.

`l` Desired 'block length' in the bootstrap. For the MBB, BWB and DWB boostrap, this is a genuine block length. For the AWB boostrap, the block length is transformed into an autoregressive parameter via the formula $0.01^(1/l)$ as in Smeekes and Urbain (2014a); this can be overwritten by setting `ar_AWB` directly. Default sets the block length as a function of the time series length T, via the rule $l = 1.75T^(1/3)$ of Palm, Smeekes and Urbain (2011).

`ar_AWB` Autoregressive parameter used in the AWB bootstrap method (`boot = "AWB"`). Can be used to set the parameter directly rather than via the default link to the block length l.

`p_min` Minimum lag length in the augmented Dickey-Fuller regression. Default is 0.

`p_max` Maximum lag length in the augmented Dickey-Fuller regression. Default uses the sample size-based rule $12(T/100)^{1/4}$.

`ic`	String for information criterion used to select the lag length in the augmented Dickey-Fuller regression. Options are: `"AIC"`, `"BIC"`, `"MAIC"`, `"MBIC"`. Default is `"MAIC"` (Ng and Perron, 2001).
`ic_scale`	Logical indicator whether or not to use the rescaled information criteria of Cavaliere et al. (2015) (`TRUE`) or not (`FALSE`). Default is `TRUE`.
`verbose`	Logical indicator whether or not information on the outcome of the unit root test needs to be printed to the console. Default is `FALSE`.
`show_progress`	Logical indicator whether a bootstrap progress update should be printed to the console. Default is FALSE.
`do_parallel`	Logical indicator whether bootstrap loop should be executed in parallel. Parallel computing is only available if OpenMP can be used, if not this option is ignored. Default is FALSE.
`nc`	The number of cores to be used in the parallel loops. Default is to use all but one.

Details

The union is taken over the combination of tests with intercept only and intercept plus trend, coupled with OLS detrending and QD detrending, as in Harvey, Leybourne and Taylor (2012) and Smeekes an Taylor (2012). The bootstrap algorithm is always based on a residual bootstrap (under the alternative) to obtain residuals rather than a difference-based bootstrap (under the null), see e.g. Palm, Smeekes and Urbain (2008).

Lag length selection is done automatically in the ADF regressions with the specified information criterion. If one of the modified criteria of Ng and Perron (2001) is used, the correction of Perron and Qu (2008) is applied. To overwrite data-driven lag length selection with a pre-specified lag length, simply set both the minimum ‘p_min‘ and maximum lag length ‘p_max‘ for the selection algorithm equal to the desired lag length.

Value

Value of the union test statistic and the bootstrap p-values.

Errors and warnings

`Error: Multiple time series not allowed. Switch to a multivariate method such as iADFtest, or change argume`
: The function is a simple wrapper around `iADFtest` to facilitate use for single time series. It does not support multiple time series, as `iADFtest` is specifically suited for that.

References

Chang, Y. and Park, J. (2003). A sieve bootstrap for the test of a unit root. *Journal of Time Series Analysis*, 24(4), 379-400.

Cavaliere, G. and Taylor, A.M.R (2009). Heteroskedastic time series with a unit root. *Econometric Theory*, 25, 1228–1276.

Cavaliere, G., Phillips, P.C.B., Smeekes, S., and Taylor, A.M.R. (2015). Lag length selection for unit root tests in the presence of nonstationary volatility. *Econometric Reviews*, 34(4), 512-536.

Friedrich, M., Smeekes, S. and Urbain, J.-P. (2020). Autoregressive wild bootstrap inference for nonparametric trends. *Journal of Econometrics*, 214(1), 81-109.

Harvey, D.I., Leybourne, S.J., and Taylor, A.M.R. (2012). Testing for unit roots in the presence of uncertainty over both the trend and initial condition. *Journal of Econometrics*, 169(2), 188-195.

Ng, S. and Perron, P. (2001). Lag Length Selection and the Construction of Unit Root Tests with Good Size and Power. *Econometrica*, 69(6), 1519-1554,

Palm, F.C., Smeekes, S. and Urbain, J.-P. (2008). Bootstrap unit root tests: Comparison and extensions. *Journal of Time Series Analysis*, 29(1), 371-401.

Paparoditis, E. and Politis, D.N. (2003). Residual-based block bootstrap for unit root testing. *Econometrica*, 71(3), 813-855.

Perron, P. and Qu, Z. (2008). A simple modification to improve the finite sample properties of Ng and Perron's unit root tests. *Economic Letters*, 94(1), 12-19.

Rho, Y. and Shao, X. (2019). Bootstrap-assisted unit root testing with piecewise locally stationary errors. *Econometric Theory*, 35(1), 142-166.

Shao, X. (2010). The dependent wild bootstrap. *Journal of the American Statistical Association*, 105(489), 218-235.

Shao, X. (2011). A bootstrap-assisted spectral test of white noise under unknown dependence. *Journal of Econometrics*, 162, 213-224.

Smeekes, S. (2013). Detrending bootstrap unit root tests. *Econometric Reviews*, 32(8), 869-891.

Smeekes, S. and Taylor, A.M.R. (2012). Bootstrap union tests for unit roots in the presence of nonstationary volatility. *Econometric Theory*, 28(2), 422-456.

Smeekes, S. and Urbain, J.-P. (2014a). A multivariate invariance principle for modified wild bootstrap methods with an application to unit root testing. GSBE Research Memorandum No. RM/14/008, Maastricht University

See Also

iADFtest

Examples

```
# boot_union on GDP_BE
GDP_BE_df <- boot_union(MacroTS[, 1], B = 399, verbose = TRUE)
```

`BSQTtest` *Bootstrap Sequential Quantile Test*

Description

Performs the Bootstrap Sequential Quantile Test (BSQT) proposed by Smeekes (2015).

Usage

```
BSQTtest(y, q = 0:NCOL(y), level = 0.05, boot = "AWB", B = 1999,
  l = NULL, ar_AWB = NULL, union = TRUE, p_min = 0, p_max = NULL,
  ic = "MAIC", dc = NULL, detr = NULL, ic_scale = TRUE,
  verbose = FALSE, show_progress = FALSE, do_parallel = FALSE,
  nc = NULL)
```

Arguments

`y` A T-dimensional vector or a (T x N)-matrix of N time series with T observations to be tested for unit roots. Data may also be in a time series format (e.g. `ts`, `zoo` or `xts`), or a data frame, as long as each column represents a single time series.

`q` Numeric vector of quantiles or units to be tested. Default is to test each unit sequentially.

`level` Desired significance level of the unit root test. Default is 0.05.

`boot` String for bootstrap method to be used. Options are

- `"MBB"` Moving blocks bootstrap (Paparoditis and Politis, 2003; Palm, Smeekes and Urbain, 2011);
- `"BWB"` Block wild bootstrap (Shao, 2011; Smeekes and Urbain, 2014a);
- `"DWB"` Dependent wild bootstrap (Shao, 2010; Smeekes and Urbain, 2014a; Rho and Shao, 2019);
- `"AWB"` Autoregressive wild bootstrap (Smeekes and Urbain, 2014a; Friedrich, Smeekes and Urbain, 2020), this is the default;
- `"SB"` Sieve bootstrap (Chang and Park, 2003; Palm, Smeekes and Urbain, 2008; Smeekes, 2013);
- `"SWB"` Sieve wild boostrap (Cavaliere and Taylor, 2009; Smeekes and Taylor, 2012).

`B` Number of bootstrap replications. Default is 1999.

`l` Desired 'block length' in the bootstrap. For the MBB, BWB and DWB boostrap, this is a genuine block length. For the AWB boostrap, the block length is transformed into an autoregressive parameter via the formula $0.01^(1/l)$ as in Smeekes and Urbain (2014a); this can be overwritten by setting `ar_AWB` directly. Default sets the block length as a function of the time series length T, via the rule $l = 1.75T^(1/3)$ of Palm, Smeekes and Urbain (2011).

`ar_AWB` Autoregressive parameter used in the AWB bootstrap method (`boot = "AWB"`). Can be used to set the parameter directly rather than via the default link to the block length l.

`union` Logical indicator whether or not to use bootstrap union tests (`TRUE`) or not (`FALSE`), see Smeekes and Taylor (2012). Default is `TRUE`.

`p_min` Minimum lag length in the augmented Dickey-Fuller regression. Default is 0.

`p_max` Maximum lag length in the augmented Dickey-Fuller regression. Default uses the sample size-based rule $12(T/100)^{1/4}$.

ic
: String for information criterion used to select the lag length in the augmented Dickey-Fuller regression. Options are: "AIC", "BIC", "MAIC", "MBIC". Default is "MAIC" (Ng and Perron, 2001).

dc
: Numeric vector indicating the deterministic specification. Only relevant if union = FALSE. Options are (combinations of)
  0 no deterministics;
  1 intercept only;
  2 intercept and trend.
  If union = FALSE, the default is adding an intercept (a warning is given).

detr
: String vector indicating the type of detrending to be performed. Only relevant if union = FALSE. Options are: "OLS" and/or "QD" (typically also called GLS, see Elliott, Rothenberg and Stock, 1996). The default is "OLS".

ic_scale
: Logical indicator whether or not to use the rescaled information criteria of Cavaliere et al. (2015) (TRUE) or not (FALSE). Default is TRUE.

verbose
: Logical indicator whether or not information on the outcome of the unit root test needs to be printed to the console. Default is FALSE.

show_progress
: Logical indicator whether a bootstrap progress update should be printed to the console. Default is FALSE.

do_parallel
: Logical indicator whether bootstrap loop should be executed in parallel. Parallel computing is only available if OpenMP can be used, if not this option is ignored. Default is FALSE.

nc
: The number of cores to be used in the parallel loops. Default is to use all but one.

Details

The parameter q can either be set as an increasing sequence of integers smaller or equal to the number of series N, or fractions of the total number of series (quantiles). For N time series, setting q = 0:N means each unit should be tested sequentially. In this case the method is equivalent to the StepM method of Romano and Wolf (2005), and therefore controls the familywise error rate. To split the series in K equally sized groups, use q = 0:K / K.

By convention and in accordance with notation in Smeekes (2015), the first entry of the vector should be equal to zero, while the second entry indicates the end of the first group, and so on. If the initial 0 or final value (1 or N) are omitted, they are automatically added by the function.

See iADFtest for details on the bootstrap algorithm and lag selection.

Value

A list with the following components

rej_H0
: Logical indicator whether the null hypothesis of a unit root is rejected (TRUE) or not (FALSE);

BSQT_sequence
: Details on the unit root tests: outcome of the sequential steps, value of the test statistics and p-values.

For the union test (union = TRUE), the output is arranged per time series. If union = FALSE, the output is arranged per time series, type of deterministic component (dc) and detrending method (detr).

Errors and warnings

`Error: Resampling-based bootstraps MBB and SB cannot handle missing values.` If the time series in `y` have different starting and end points (and thus some series contain `NA` values at the beginning and/or end of the sample, the resampling-based moving block bootstrap (MBB) and sieve bootstrap (SB) cannot be used, as they create holes (internal missings) in the bootstrap samples. Switch to another bootstrap method or truncate your sample to eliminate `NA` values.

`Error: Invalid input values for q: must be quantiles or positive integers.` Construction of `q` does not satisfy the criteria listed under 'Details'.

`Warning: SB and SWB bootstrap only recommended for iADFtest; see help for details.` Although the sieve bootstrap methods `"SB"` and `"SWB"` can be used, Smeekes and Urbain (2014b) show that these are not suited to capture general forms of dependence across units, and using them for joint or multiple testing is not valid. This warning thereofre serves to recommend the user to consider a different bootstrap method.

`Warning: Deterministic specification in argument dc is ignored, as union test is applied.` The union test calculates the union of all four combinations of deterministic components (intercept or intercept and trend) and detrending methods (OLS or QD). Setting deterministic components manually therefore has no effect.

`Warning: Detrending method in argument detr is ignored, as union test is applied.` The union test calculates the union of all four combinations of deterministic components (intercept or intercept and trend) and detrending methods (OLS or QD). Setting detrending methods manually therefore has no effect.

References

Chang, Y. and Park, J. (2003). A sieve bootstrap for the test of a unit root. *Journal of Time Series Analysis*, 24(4), 379-400.

Cavaliere, G. and Taylor, A.M.R (2009). Heteroskedastic time series with a unit root. *Econometric Theory*, 25, 1228–1276.

Cavaliere, G., Phillips, P.C.B., Smeekes, S., and Taylor, A.M.R. (2015). Lag length selection for unit root tests in the presence of nonstationary volatility. *Econometric Reviews*, 34(4), 512-536.

Elliott, G., Rothenberg, T.J., and Stock, J.H. (1996). Efficient tests for an autoregressive unit root. *Econometrica*, 64(4), 813-836.

Friedrich, M., Smeekes, S. and Urbain, J.-P. (2020). Autoregressive wild bootstrap inference for nonparametric trends. *Journal of Econometrics*, 214(1), 81-109.

Ng, S. and Perron, P. (2001). Lag Length Selection and the Construction of Unit Root Tests with Good Size and Power. *Econometrica*, 69(6), 1519-1554,

Palm, F.C., Smeekes, S. and Urbain, J.-P. (2008). Bootstrap unit root tests: Comparison and extensions. *Journal of Time Series Analysis*, 29(1), 371-401.

Palm, F. C., Smeekes, S., and Urbain, J.-.P. (2011). Cross-sectional dependence robust block bootstrap panel unit root tests. *Journal of Econometrics*, 163(1), 85-104.

Paparoditis, E. and Politis, D.N. (2003). Residual-based block bootstrap for unit root testing. *Econometrica*, 71(3), 813-855.

Perron, P. and Qu, Z. (2008). A simple modification to improve the finite sample properties of Ng and Perron's unit root tests. *Economic Letters*, 94(1), 12-19.

Rho, Y. and Shao, X. (2019). Bootstrap-assisted unit root testing with piecewise locally stationary errors. *Econometric Theory*, 35(1), 142-166.

Romano, J. P. and Wolf, M. (2005). Stepwise multiple testing as formalized data snooping. *Econometrica*, 73(4), 1237-1282. #' @references Shao, X. (2010). The dependent wild bootstrap. *Journal of the American Statistical Association*, 105(489), 218-235.

Shao, X. (2011). A bootstrap-assisted spectral test of white noise under unknown dependence. *Journal of Econometrics*, 162, 213-224.

Smeekes, S. (2013). Detrending bootstrap unit root tests. *Econometric Reviews*, 32(8), 869-891.

Smeekes, S. (2015). Bootstrap sequential tests to determine the order of integration of individual units in a time series panel. *Journal of Time Series Analysis*, 36(3), 398-415.

Smeekes, S. and Taylor, A.M.R. (2012). Bootstrap union tests for unit roots in the presence of nonstationary volatility. *Econometric Theory*, 28(2), 422-456.

Smeekes, S. and Urbain, J.-P. (2014a). A multivariate invariance principle for modified wild bootstrap methods with an application to unit root testing. GSBE Research Memorandum No. RM/14/008, Maastricht University

Smeekes, S. and Urbain, J.-P. (2014b). On the applicability of the sieve bootstrap in time series panels. *Oxford Bulletin of Economics and Statistics*, 76(1), 139-151.

See Also

iADFtest

Examples

```
# BSQTtest on GDP_BE and GDP_DE
two_series_BSQTtest <- BSQTtest(MacroTS[, 1:2], boot = "AWB", B = 399,  verbose = TRUE)
```

---

check_missing_insample_values
Check Missing Values in Sample

Description

Check Missing Values in Sample

Usage

```
check_missing_insample_values(X)
```

Arguments

X	A $(T \mathrm{x} N)$-matrix of N time series with T observations. Data may also be in a time series format (e.g. ts, zoo or xts) or data frame.

Value

N-dimensional vector, for each series whether missing values are present (TRUE) or not (FALSE)

diff_mult — *Differences of Multiple Time Series*

Description

Performs differencing of multiple time series, with possibly different orders for each time series.

Usage

```
diff_mult(y, d, keep_NAs = TRUE)
```

Arguments

y	A $(T\text{x}N)$-matrix of N time series with T observations. Data may also be in a time series format (e.g. ts, zoo or xts) or data frame.
d	An N-dimensional vector containing the orders
keep_NAs	Logical indicator whether or not to keep the NA values resulting from differencing at the beginning of the sample. Default is TRUE. If FALSE, the entire row containing the NA values is removed.

Value

The appropriately differenced data in the same format as the original data.

find_nonmissing_subsample — *Find Non-Missing Subsamples*

Description

Find Non-Missing Subsamples

Usage

```
find_nonmissing_subsample(X)
```

Arguments

X	A $(T\text{x}N)$-matrix of N time series with T observations. Data may also be in a time series format (e.g. ts, zoo or xts) or data frame. Assumes a prior check on missing values in-sample has been done.

Value

A list with the following components

`range` (2xN)-dimensional matrix containing the first and last non-missing observation in each column of X.

`all_equal` Logical value indicating whether all series have the same non-missing indices.

`iADFtest` *Individual Unit Root Tests without multiple testing control*

Description

This function performs bootstrap unit root tests on each time series individually.

Usage

```
iADFtest(y, level = 0.05, boot = "AWB", B = 1999, l = NULL,
  ar_AWB = NULL, union = TRUE, p_min = 0, p_max = NULL, ic = "MAIC",
  dc = NULL, detr = NULL, ic_scale = TRUE, verbose = FALSE,
  show_progress = FALSE, do_parallel = FALSE, nc = NULL)
```

Arguments

`y` A T-dimensional vector or a (T x N)-matrix of N time series with T observations to be tested for unit roots. Data may also be in a time series format (e.g. `ts`, `zoo` or `xts`), or a data frame, as long as each column represents a single time series.

`level` Desired significance level of the unit root test. Default is 0.05.

`boot` String for bootstrap method to be used. Options are

- `"MBB"` Moving blocks bootstrap (Paparoditis and Politis, 2003; Palm, Smeekes and Urbain, 2011);
- `"BWB"` Block wild bootstrap (Shao, 2011; Smeekes and Urbain, 2014a);
- `"DWB"` Dependent wild bootstrap (Shao, 2010; Smeekes and Urbain, 2014a; Rho and Shao, 2019);
- `"AWB"` Autoregressive wild bootstrap (Smeekes and Urbain, 2014a; Friedrich, Smeekes and Urbain, 2020), this is the default;
- `"SB"` Sieve bootstrap (Chang and Park, 2003; Palm, Smeekes and Urbain, 2008; Smeekes, 2013);
- `"SWB"` Sieve wild boostrap (Cavaliere and Taylor, 2009; Smeekes and Taylor, 2012).

`B` Number of bootstrap replications. Default is 1999.

l	Desired 'block length' in the bootstrap. For the MBB, BWB and DWB boostrap, this is a genuine block length. For the AWB boostrap, the block length is transformed into an autoregressive parameter via the formula $0.01^(1/l)$ as in Smeekes and Urbain (2014a); this can be overwritten by setting `ar_AWB` directly. Default sets the block length as a function of the time series length T, via the rule $l = 1.75T^(1/3)$ of Palm, Smeekes and Urbain (2011).
ar_AWB	Autoregressive parameter used in the AWB bootstrap method (`boot = "AWB"`). Can be used to set the parameter directly rather than via the default link to the block length l.
union	Logical indicator whether or not to use bootstrap union tests (`TRUE`) or not (`FALSE`), see Smeekes and Taylor (2012). Default is `TRUE`.
p_min	Minimum lag length in the augmented Dickey-Fuller regression. Default is 0.
p_max	Maximum lag length in the augmented Dickey-Fuller regression. Default uses the sample size-based rule $12(T/100)^{1/4}$.
ic	String for information criterion used to select the lag length in the augmented Dickey-Fuller regression. Options are: `"AIC"`, `"BIC"`, `"MAIC"`, `"MBIC"`. Default is `"MAIC"` (Ng and Perron, 2001).
dc	Numeric vector indicating the deterministic specification. Only relevant if `union = FALSE`. Options are (combinations of) 0 no deterministics; 1 intercept only; 2 intercept and trend. If `union = FALSE`, the default is adding an intercept (a warning is given).
detr	String vector indicating the type of detrending to be performed. Only relevant if `union = FALSE`. Options are: `"OLS"` and/or `"QD"` (typically also called GLS, see Elliott, Rothenberg and Stock, 1996). The default is `"OLS"`.
ic_scale	Logical indicator whether or not to use the rescaled information criteria of Cavaliere et al. (2015) (`TRUE`) or not (`FALSE`). Default is `TRUE`.
verbose	Logical indicator whether or not information on the outcome of the unit root test needs to be printed to the console. Default is `FALSE`.
show_progress	Logical indicator whether a bootstrap progress update should be printed to the console. Default is FALSE.
do_parallel	Logical indicator whether bootstrap loop should be executed in parallel. Parallel computing is only available if OpenMP can be used, if not this option is ignored. Default is FALSE.
nc	The number of cores to be used in the parallel loops. Default is to use all but one.

Details

The options encompass many test proposed in the literature. `dc = "OLS"` gives the standard augmented Dickey-Fuller test, while `dc = "QD"` provides the DF-GLS test of Elliott, Rothenberg and Stock (1996). The bootstrap algorithm is always based on a residual bootstrap (under the alternative) to obtain residuals rather than a difference-based bootstrap (under the null), see e.g. Palm, Smeekes and Urbain (2008).

Lag length selection is done automatically in the ADF regression with the specified information criterion. If one of the modified criteria of Ng and Perron (2001) is used, the correction of Perron and Qu (2008) is applied. For very short time series (fewer than 50 time points) the maximum lag length is adjusted downward to avoid potential multicollinearity issues in the bootstrap. To overwrite data-driven lag length selection with a pre-specified lag length, simply set both the minimum 'p_min' and maximum lag length 'p_max' for the selection algorithm equal to the desired lag length.

Value

A list with the following components

- `rej_H0` Logical indicator whether the null hypothesis of a unit root is rejected (`TRUE`) or not (`FALSE`);
- `ADF_tests` Details on the unit root tests: value of the test statistics and p-values.

For the union test (`union = TRUE`), the output is arranged per time series. If `union = FALSE`, the output is arranged per time series, type of deterministic component (`dc`) and detrending method (`detr`).

Warnings

The function may give the following warnings.

`Warning: Missing values cause resampling bootstrap to be executed for each time series individually.` If the time series in `y` have different starting and end points (and thus some series contain `NA` values at the beginning and/or end of the sample, the resampling-based moving block bootstrap (MBB) and sieve bootstrap (SB) cannot be used directly, as they create holes (internal missings) in the bootstrap samples. These bootstrap methods are therefore not applied jointly as usual, but individually to each series.

`Warning: Deterministic specification in argument dc is ignored, as union test is applied.` The union test calculates the union of all four combinations of deterministic components (intercept or intercept and trend) and detrending methods (OLS or QD). Setting deterministic components manually therefore has no effect.

`Warning: Detrending method in argument detr is ignored, as union test is applied.` The union test calculates the union of all four combinations of deterministic components (intercept or intercept and trend) and detrending methods (OLS or QD). Setting detrending methods manually therefore has no effect.

References

Chang, Y. and Park, J. (2003). A sieve bootstrap for the test of a unit root. *Journal of Time Series Analysis*, 24(4), 379-400.

Cavaliere, G. and Taylor, A.M.R (2009). Heteroskedastic time series with a unit root. *Econometric Theory*, 25, 1228–1276.

Cavaliere, G., Phillips, P.C.B., Smeekes, S., and Taylor, A.M.R. (2015). Lag length selection for unit root tests in the presence of nonstationary volatility. *Econometric Reviews*, 34(4), 512-536.

Elliott, G., Rothenberg, T.J., and Stock, J.H. (1996). Efficient tests for an autoregressive unit root. *Econometrica*, 64(4), 813-836.

Friedrich, M., Smeekes, S. and Urbain, J.-P. (2020). Autoregressive wild bootstrap inference for nonparametric trends. *Journal of Econometrics*, 214(1), 81-109.

Ng, S. and Perron, P. (2001). Lag Length Selection and the Construction of Unit Root Tests with Good Size and Power. *Econometrica*, 69(6), 1519-1554,

Palm, F.C., Smeekes, S. and Urbain, J.-P. (2008). Bootstrap unit root tests: Comparison and extensions. *Journal of Time Series Analysis*, 29(1), 371-401.

Palm, F. C., Smeekes, S., and Urbain, J.-.P. (2011). Cross-sectional dependence robust block bootstrap panel unit root tests. *Journal of Econometrics*, 163(1), 85-104.

Paparoditis, E. and Politis, D.N. (2003). Residual-based block bootstrap for unit root testing. *Econometrica*, 71(3), 813-855.

Perron, P. and Qu, Z. (2008). A simple modification to improve the finite sample properties of Ng and Perron's unit root tests. *Economic Letters*, 94(1), 12-19.

Rho, Y. and Shao, X. (2019). Bootstrap-assisted unit root testing with piecewise locally stationary errors. *Econometric Theory*, 35(1), 142-166.

Shao, X. (2010). The dependent wild bootstrap. *Journal of the American Statistical Association*, 105(489), 218-235.

Shao, X. (2011). A bootstrap-assisted spectral test of white noise under unknown dependence. *Journal of Econometrics*, 162, 213-224.

Smeekes, S. (2013). Detrending bootstrap unit root tests. *Econometric Reviews*, 32(8), 869-891.

Smeekes, S. and Taylor, A.M.R. (2012). Bootstrap union tests for unit roots in the presence of nonstationary volatility. *Econometric Theory*, 28(2), 422-456.

Smeekes, S. and Urbain, J.-P. (2014a). A multivariate invariance principle for modified wild bootstrap methods with an application to unit root testing. GSBE Research Memorandum No. RM/14/008, Maastricht University

Examples

```
# iADFtest on GDP_BE and GDP_DE
two_series_iADFtest <- iADFtest(MacroTS[, 1:2], boot = "MBB", B = 399,
verbose = TRUE)
```

`MacroTS` *Macroeconomic Time Series*

Description

Macroeconomic data from Eurostat on GDP, consumption, inflation and unemployment for Belgium, Germany, France, the Netherlands and the United Kingdom.

Usage

```
MacroTS
```

Format

A time series object containing 20 macroeconomic seasonally adjusted time series, quarterly observed from 1992-2019 for Belgium (BE), Germany (DE), France (FR), the Netherlands (NL) and the United Kingdom (UK).

GDP_BE Gross domestic product at market prices (index, 2015=100) for Belgium.

GDP_DE Gross domestic product at market prices (index, 2015=100) for Germany.

GDP_FR Gross domestic product at market prices (index, 2015=100) for France.

GDP_NL Gross domestic product at market prices (index, 2015=100) for the Netherlands.

GDP_UK Gross domestic product at market prices (index, 2015=100) for the United Kingdom.

CONS_BE Final consumption expenditure (index, 2015=100) for Belgium.

CONS_DE Final consumption expenditure (index, 2015=100) for Germany.

CONS_FR Final consumption expenditure (index, 2015=100) for France.

CONS_NL Final consumption expenditure (index, 2015=100) for the Netherlands.

CONS_UK Final consumption expenditure (index, 2015=100) for the United Kingdom.

HICP_BE Harmonised Indices of Consumer Prices (annual rate of change, 2015=100) for Belgium.

HICP_DE Harmonised Indices of Consumer Prices (annual rate of change, 2015=100) for Germany.

HICP_FR Harmonised Indices of Consumer Prices (annual rate of change, 2015=100) for France.

HICP_N Harmonised Indices of Consumer Prices (annual rate of change, 2015=100) for the Netherlands.

HICP_UK Harmonised Indices of Consumer Prices (annual rate of change, 2015=100) for the United Kingdom.

UR_BE Unemployment rate (percentage of the active population) for Belgium.

UR_DE Unemployment rate (percentage of the active population) for Germany.

UR_FR Unemployment rate (percentage of the active population) for France.

UR_NL Unemployment rate (percentage of the active population) for the Netherlands.

UR_UK Unemployment rate (percentage of the active population) for the United Kingdom.

Note

- Unemployment rates are seasonally but not calendar adjusted, all other series are both seasonally and calendar adjusted.
- Quarterly inflation rates are sampled from Eurostat's monthly series with annual rates of change as the final month of the respective quarter.
- The unemployment rate for France excludes overseas territories ('France continental' in the Eurostat database).

Source

https://ec.europa.eu/eurostat/data/database `order_integration` *Determine Order of Integration*

Description

Determines the order of integration for each time series in a dataset via a sequence of unit root tests, and differences the data accordingly to eliminate stochastic trends.

Usage

```
order_integration(y, max_order = 2, test = NULL, plot_orders = FALSE, ...)
```

Arguments

`y`	A $(T\text{x}N)$-matrix of N time series with T observations. Data may also be in a time series format (e.g. `ts`, `zoo` or `xts`) or data frame.
`max_order`	The maximum order of integration of the time series. Default is 2.
`test`	The unit root tests to be used in the procedure. For multiple time series the options are "iADFtest", "BSQTtest" and "bFDRtest", with "iADFtest" the default. For single time series the options are "boot_df" and "boot_union", with the latter the default.
`plot_orders`	Logical indicator whether the resulting orders of integration should be plotted. Default is `FALSE`.
`...`	Optional arguments passed to the chosen unit root test function.

Details

The function follows the approach laid out in Smeekes and Wijler (2020), where all series is differenced $d-1$ times, where d is the specified maximum order, and these differenced series are tested for unit roots. The series foe which the unit root null is not rejected, are classified as $I(d)$ and removed from consideration. The remaining series are integrated, and tested for unit roots again, leading to a classification of $I(d-1)$ series if the null is not rejected. This is continued until a non-rejection is observed for all time series, or the series are integrated back to their original level. The series for which the null hypothesis is rejected in the final stage are classified as $I(0)$.

Care must be taken when using `BSQTtest` when the argument `q` is given as a sequence of integers. As at each step series are removed, one may end up with fewer series to test than indicated in `q`. While integers larger than the number of series will automatically be removed - along with a warning - by the test, it is recommend to set `q` in the form of quantiles.

Plotting the orders of integration requires the `ggplot2` package to be installed; plot will be skipped and a warning is given if not. For plots the function `plot_order_integration` is called. The user may prefer to set `plot_orders = FALSE` and call this function directly using the returned value of `order_int` in order to have more control over plot settings and save the plot object.

Value

A list with the following components

- `order_int` A vector with the found orders of integration of each time series.
- `diff_data` The appropriately differenced data according to `order_int` in the same format as the original data.

References

Smeekes, S. and Wijler, E. (2020). Unit roots and cointegration. In P. Fuleky (Ed.) *Macroeconomic Forecasting in the Era of Big Data*, Chapter 17, pp. 541-584. *Advanced Studies in Theoretical and Applied Econometrics*, vol. 52. Springer.

Examples

```
# Use iADFtest to determine the order of GDP_BE and GDP_DE
orders_iADFtest <- order_integration(MacroTS[, 1:2], test = "iADFtest", B = 199)
```

---

paneltest *Panel Unit Root Test*

Description

Performs a test on a multivariate (panel) time series by testing the null hypothesis that all series have a unit root. The test is based on averaging the individual test statistics, also called the Group-Mean (GM) test in Palm, Smeekes and Urbain (2011).

Usage

```
paneltest(y, level = 0.05, boot = "AWB", B = 1999, l = NULL,
  ar_AWB = NULL, union = TRUE, p_min = 0, p_max = NULL, ic = "MAIC",
  dc = NULL, detr = NULL, ic_scale = TRUE, verbose = FALSE,
  show_progress = FALSE, do_parallel = FALSE, nc = NULL)
```

Arguments

- `y` A T-dimensional vector or a (T x N)-matrix of N time series with T observations to be tested for unit roots. Data may also be in a time series format (e.g. `ts`, `zoo` or `xts`), or a data frame, as long as each column represents a single time series.
- `level` Desired significance level of the unit root test. Default is 0.05.
- `boot` String for bootstrap method to be used. Options are
  - `"MBB"` Moving blocks bootstrap (Paparoditis and Politis, 2003; Palm, Smeekes and Urbain, 2011);
  - `"BWB"` Block wild bootstrap (Shao, 2011; Smeekes and Urbain, 2014a);

"DWB" Dependent wild bootstrap (Shao, 2010; Smeekes and Urbain, 2014a; Rho and Shao, 2019);

"AWB" Autoregressive wild bootstrap (Smeekes and Urbain, 2014a; Friedrich, Smeekes and Urbain, 2020), this is the default;

"SB" Sieve bootstrap (Chang and Park, 2003; Palm, Smeekes and Urbain, 2008; Smeekes, 2013);

"SWB" Sieve wild boostrap (Cavaliere and Taylor, 2009; Smeekes and Taylor, 2012).

B — Number of bootstrap replications. Default is 1999.

l — Desired 'block length' in the bootstrap. For the MBB, BWB and DWB boostrap, this is a genuine block length. For the AWB boostrap, the block length is transformed into an autoregressive parameter via the formula $0.01^(1/l)$ as in Smeekes and Urbain (2014a); this can be overwritten by setting ar_AWB directly. Default sets the block length as a function of the time series length T, via the rule $l = 1.75T^(1/3)$ of Palm, Smeekes and Urbain (2011).

ar_AWB — Autoregressive parameter used in the AWB bootstrap method (boot = "AWB"). Can be used to set the parameter directly rather than via the default link to the block length l.

union — Logical indicator whether or not to use bootstrap union tests (TRUE) or not (FALSE), see Smeekes and Taylor (2012). Default is TRUE.

p_min — Minimum lag length in the augmented Dickey-Fuller regression. Default is 0.

p_max — Maximum lag length in the augmented Dickey-Fuller regression. Default uses the sample size-based rule $12(T/100)^{1/4}$.

ic — String for information criterion used to select the lag length in the augmented Dickey-Fuller regression. Options are: "AIC", "BIC", "MAIC", "MBIC". Default is "MAIC" (Ng and Perron, 2001).

dc — Numeric vector indicating the deterministic specification. Only relevant if union = FALSE. Options are (combinations of)

0 no deterministics;

1 intercept only;

2 intercept and trend.

If union = FALSE, the default is adding an intercept (a warning is given).

detr — String vector indicating the type of detrending to be performed. Only relevant if union = FALSE. Options are: "OLS" and/or "QD" (typically also called GLS, see Elliott, Rothenberg and Stock, 1996). The default is "OLS".

ic_scale — Logical indicator whether or not to use the rescaled information criteria of Cavaliere et al. (2015) (TRUE) or not (FALSE). Default is TRUE.

verbose — Logical indicator whether or not information on the outcome of the unit root test needs to be printed to the console. Default is FALSE.

show_progress — Logical indicator whether a bootstrap progress update should be printed to the console. Default is FALSE.

do_parallel — Logical indicator whether bootstrap loop should be executed in parallel. Parallel computing is only available if OpenMP can be used, if not this option is ignored. Default is FALSE.

nc — The number of cores to be used in the parallel loops. Default is to use all but one.

Details

See `iADFtest` for details on the bootstrap algorithm and lag selection.

Value

For the union test (`union = TRUE`), the test statistic and p-value are returned. If `union = FALSE`, the test statistics and p-values are reported per type of deterministic component (`dc`) and detrending method (`detr`).

Errors and warnings

`Error: Resampling-based bootstraps MBB and SB cannot handle missing values.` If the time series in `y` have different starting and end points (and thus some series contain `NA` values at the beginning and/or end of the sample, the resampling-based moving block bootstrap (MBB) and sieve bootstrap (SB) cannot be used, as they create holes (internal missings) in the bootstrap samples. Switch to another bootstrap method or truncate your sample to eliminate `NA` values.

`Warning: SB and SWB bootstrap only recommended for iADFtest; see help for details.` Although the sieve bootstrap methods `"SB"` and `"SWB"` can be used, Smeekes and Urbain (2014b) show that these are not suited to capture general forms of dependence across units, and using them for joint or multiple testing is not valid. This warning thereofre serves to recommend the user to consider a different bootstrap method.

`Warning: Deterministic specification in argument dc is ignored, as union test is applied.` The union test calculates the union of all four combinations of deterministic components (intercept or intercept and trend) and detrending methods (OLS or QD). Setting deterministic components manually therefore has no effect.

`Warning: Detrending method in argument detr is ignored, as union test is applied.` The union test calculates the union of all four combinations of deterministic components (intercept or intercept and trend) and detrending methods (OLS or QD). Setting detrending methods manually therefore has no effect.

References

Chang, Y. and Park, J. (2003). A sieve bootstrap for the test of a unit root. *Journal of Time Series Analysis*, 24(4), 379-400.

Cavaliere, G. and Taylor, A.M.R (2009). Heteroskedastic time series with a unit root. *Econometric Theory*, 25, 1228–1276.

Cavaliere, G., Phillips, P.C.B., Smeekes, S., and Taylor, A.M.R. (2015). Lag length selection for unit root tests in the presence of nonstationary volatility. *Econometric Reviews*, 34(4), 512-536.

Elliott, G., Rothenberg, T.J., and Stock, J.H. (1996). Efficient tests for an autoregressive unit root. *Econometrica*, 64(4), 813-836.

Friedrich, M., Smeekes, S. and Urbain, J.-P. (2020). Autoregressive wild bootstrap inference for nonparametric trends. *Journal of Econometrics*, 214(1), 81-109.

Ng, S. and Perron, P. (2001). Lag Length Selection and the Construction of Unit Root Tests with Good Size and Power. *Econometrica*, 69(6), 1519-1554,

Palm, F.C., Smeekes, S. and Urbain, J.-P. (2008). Bootstrap unit root tests: Comparison and extensions. *Journal of Time Series Analysis*, 29(1), 371-401.

Palm, F. C., Smeekes, S., and Urbain, J.-.P. (2011). Cross-sectional dependence robust block bootstrap panel unit root tests. *Journal of Econometrics*, 163(1), 85-104.

Paparoditis, E. and Politis, D.N. (2003). Residual-based block bootstrap for unit root testing. *Econometrica*, 71(3), 813-855.

Perron, P. and Qu, Z. (2008). A simple modification to improve the finite sample properties of Ng and Perron's unit root tests. *Economic Letters*, 94(1), 12-19.

Rho, Y. and Shao, X. (2019). Bootstrap-assisted unit root testing with piecewise locally stationary errors. *Econometric Theory*, 35(1), 142-166.

Shao, X. (2010). The dependent wild bootstrap. *Journal of the American Statistical Association*, 105(489), 218-235.

Shao, X. (2011). A bootstrap-assisted spectral test of white noise under unknown dependence. *Journal of Econometrics*, 162, 213-224.

Smeekes, S. (2013). Detrending bootstrap unit root tests. *Econometric Reviews*, 32(8), 869-891.

Smeekes, S. and Taylor, A.M.R. (2012). Bootstrap union tests for unit roots in the presence of nonstationary volatility. *Econometric Theory*, 28(2), 422-456.

Smeekes, S. and Urbain, J.-P. (2014a). A multivariate invariance principle for modified wild bootstrap methods with an application to unit root testing. GSBE Research Memorandum No. RM/14/008, Maastricht University

Smeekes, S. and Urbain, J.-P. (2014b). On the applicability of the sieve bootstrap in time series panels. *Oxford Bulletin of Economics and Statistics*, 76(1), 139-151.

See Also

`iADFtest`

Examples

```
# paneltest on GDP_BE and GDP_DE
two_series_paneltest <- paneltest(MacroTS[, 1:2], boot = "AWB", B = 399,  verbose = TRUE)
```

`plot_missing_values` *Plot Missing Values*

Description

Plots missing values of different types for a time series dataset.

Usage

```
plot_missing_values(y, show_names = FALSE, show_legend = TRUE,
  axis_text_size = NULL, legend_size = NULL, cols = NULL)
```

Arguments

- `y` — A $(T\text{x}N)$-matrix of N time series with T observations. Data may also be in a time series format (e.g. `ts`, `zoo` or `xts`) or data frame.
- `show_names` — Show the time series' names on the plot (`TRUE`) or not (`FALSE`). Default is `TRUE`.
- `show_legend` — Logical indicator whether a legend should be displayed. Default is `TRUE`.
- `axis_text_size` — Size of the text on the axis. Default takes `ggplot2` defaults.
- `legend_size` — Size of the text in the legend if `show_legend = TRUE`. Default takes `ggplot2` defaults.
- `cols` — Vector with colours for displaying the different types of data. If the default is overwriten, four colours must be supplied.

Details

The function distinguish four types of data: observed data (non-missing) and three missing types. Type `"Balanced NA"` indicates where entire rows are missing (`NA`). These do not cause unbalancedness as the missing rows can simply be deleted. Type `"Unalanced NA"` are missing values on the beginning or end of the sample, which cause unbalancedness. These affect some (but not all) bootstrap methods, see e.g.~`bFDRtest`. Type `"Internal NA"` are missing values inside the sample, which need to be removed before the bootstrap unit root tests can be used.

This function requires the package `ggplot2` to be installed. If the package is not found, plotting is aborted.

Value

A `ggplot2` object containing the missing values plot.

`plot_order_integration` *Plot Orders of Integration*

Description

Plots a vector with orders of integration of time series.

Usage

```
plot_order_integration(d, show_names = TRUE, show_legend = TRUE,
  names_size = NULL, legend_size = NULL, cols = NULL)
```

Arguments

d	TN-dimensional vector with time series' orders of integration. Elements should be named after the respective time series to ensure easy interpretation of the plot.
show_names	Show the time series' names on the plot (TRUE) or not (FALSE). Default is TRUE.
show_legend	Logical indicator whether a legend should be displayed. Default is TRUE.
names_size	Size of the time series' names if show_names = TRUE. Default takes ggplot2 defaults.
legend_size	Size of the text in the legend if show_legend = TRUE. Default takes ggplot2 defaults.
cols	Vector with colours for displaying the orders of integration. At least as many colours as orders of integration need to be supplied. Default supplies 4 colours for displaying up to $I(3)$ series.

Details

This function requires the package ggplot2 to be installed. If the package is not found, plotting is aborted.

Value

A ggplot2 object containing the plot of the orders of integration.

Index