Package: tsexplorer (via r-universe)

October 22, 2026

Type Package

Title Exploratory and Diagnostic Tools for Time Series and Forecasts

Version 0.1.0

Description Exploratory, descriptive and diagnostic tools for univariate and multivariate time series and tools to evaluate and compare forecasts: summary statistics and gaussianity tests; sample autocorrelation functions with the Ljung and Box (1978) <doi:10.1093/biomet/65.2.297> and Monti (1994) <doi:10.2307/2337080> tests; information criteria and Granger (1969) <doi:10.2307/1912791> causality; unit root tests of Dickey and Fuller (1979) <doi:10.1080/01621459.1979.10482531> and Phillips and Perron (1988) <doi:10.1093/biomet/75.2.335>; the cointegration test of Johansen (1988) <doi:10.1016/0165-1889(88)90041-3>; heteroskedasticity tests; non-linearity tests of McLeod and Li (1983) <doi:10.1111/j.1467-9892.1983.tb00373.x>, Tsay (1986) <doi:10.1093/biomet/73.2.461> and Brock, Dechert, Scheinkman and LeBaron (1996) <doi:10.1080/07474939608800353>; spectral analysis; and forecast accuracy measures with the Diebold and Mariano (1995) <doi:10.1080/07350015.1995.10524599>, sign and Wilcoxon tests. Every tool is a pure function returning an object with print and plot methods; the graphical interfaces tooltest() and toolforecast() ('Tcl/Tk' windows or 'shiny' applications) give interactive access to all of them.

License GPL-3

URL https://github.com/djpedregal/tsexplorer

BugReports https://github.com/djpedregal/tsexplorer/issues

Encoding UTF-8

LazyData true

Depends R (>= 4.1.0)

Imports graphics, grDevices, stats, utils

Suggests shiny, tcltk, testthat (>= 3.0.0), jsonlite, tseries, urca, xts, zoo

Config/testthat/edition 3

Roxygen list(markdown = TRUE)

Config/roxygen2/version 8.1.0

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

Date/Publication 2026-10-13 10:07:57 UTC

RemoteUrl https://github.com/djpedregal/tsexplorer

RemoteRef HEAD

RemoteSha 542370c1996040ea5772321275bda25c529e3cda

RemoteSubdir r

Contents

adfcritical . . . 3
adftest . . . 4
airpas . . . 5
as_tsx_matrix . . . 5
electricity . . . 6
faccuracy . . . 7
fcompare . . . 9
fplot . . . 10
furnace . . . 11
gaselec . . . 12
gausstest . . . 12
johansentest . . . 13
lydia_pinkham . . . 14
meda . . . 15
nile . . . 15
nltests . . . 16
plotband . . . 17
scc . . . 18
stdmean . . . 18
steel . . . 19
sumstats . . . 19
testhet . . . 20
toolforecast . . . 21
tooltest . . . 23
tooltest_defaults . . . 24
unemployment . . . 25
usgdp . . . 25
varsbc . . . 25
varsbc2 . . . 26
varspectrum . . . 27
vbds . . . 28
vboxcox . . . 29
vboxplot . . . 30

vcumperiod 31
vcusum 31
vdif 32
vgranger 33
vhist 34
victims 34
vident 35
vident2 36
vphillips 37
vqqplot 38
vscatter 39
vspectrum 39
vtboxplot 40
vtsay 41

Index **42**

`adfcritical` *Approximate p-values of Dickey-Fuller statistics*

Description

Interpolates the p-value of Dickey-Fuller t and F statistics in the tables of Fuller (1976) and Dickey and Fuller (1981). The critical values are first interpolated linearly in the sample size (clamped to 25; sizes above 500 use the asymptotic row), then the p-value is interpolated linearly in the statistic. Outside the tables the p-value is set to 0 or 1.

Usage

```
adfcritical(
  x,
  n,
  table = c("t_none", "t_const", "t_trend", "f_const", "f_trend")
)
```

Arguments

`x`	Vector of statistics.
`n`	Sample size.
`table`	One of `"t_none"`, `"t_const"`, `"t_trend"` (t statistics without deterministic terms, with constant, with constant and trend), `"f_const"` (Φ_1) or `"f_trend"` (Φ_3).

Value

Vector of p-values.

Examples

```
adfcritical(c(-3.5, -2, 0), 100, "t_const")
```

---

adftest *Augmented Dickey-Fuller unit root tests*

Description

ADF regressions of z_t on z_{t-1}, p lagged differences and the deterministic terms of three models: no constant and no trend, constant, and constant plus linear trend. The tests are the t statistics of $H_0 : \rho = 1$ and the F statistics Φ_1 ($\rho = 1$ and no constant) and Φ_3 ($\rho = 1$ and no trend).

Usage

```
adftest(y, lags = NULL, names = NULL)
```

Arguments

`y`	Time series data (see `as_tsx_matrix()`).
`lags`	Number(s) of lagged differences included in the regressions. Default: `1:min(floor(sqrt(T)), 20)` for one series and `min(floor(sqrt(T)), 20)` for several series. If the largest lag exceeds `T / 2` the test is computed only for `floor(T / 2)` lags.
`names`	Optional names of the series.

Details

P-values are approximate, interpolated in the tables of Dickey and Fuller (see `adfcritical()`).

Value

An object of class `tsx_adftest` with fields `call`, `series`, `n` (sample size used for the p-values) and three data.frames, one row per series and lag:

- `none`: `series`, `lag`, `rho`, `t`, `p_t`,
- `const`: as `none` plus `const`, `t_const`, `f` (Φ_1), `p_f`,
- `trend`: as `const` plus `trend`, `t_trend`, with `f` being Φ_3.

References

Dickey, D.A. and Fuller, W.A. (1979). Distribution of the estimators for autoregressive time series with a unit root. *Journal of the American Statistical Association*, 74, 427-431.

Dickey, D.A. and Fuller, W.A. (1981). Likelihood ratio statistics for autoregressive time series with a unit root. *Econometrica*, 49, 1057-1072.

Said, S.E. and Dickey, D.A. (1984). Testing for unit roots in autoregressive-moving average models of unknown order. *Biometrika*, 71, 599-607.

Examples

```
adftest(log(airpas), lags = 1:4)
```

---

airpas — *Airline passengers*

Description

Monthly totals of international airline passengers (thousands), January 1949 to December 1960. Series G of Box and Jenkins.

Usage

```
airpas
```

Format

A monthly `ts` object with 144 observations.

Source

Box, G.E.P., Jenkins, G.M. and Reinsel, G.C. (2008). *Time Series Analysis, Forecasting and Control*. Wiley.

---

as_tsx_matrix — *Convert input data to the matrix format used by tsexplorer*

Description

Every function of the package accepts data without dates (numeric vector, matrix or numeric `data.frame`) or with dates (`ts`, `mts`, `zoo`, `xts` or a `data.frame` with one date column) and converts it with this function into a numeric `T x m` matrix (rows are time, columns are series) with column names.

Usage

```
as_tsx_matrix(y, names = NULL, trim = TRUE)
```

Arguments

`y`	Numeric vector, matrix, `ts`, `mts`, `zoo`, `xts` or `data.frame` (numeric columns and, optionally, one column of class `Date` or `POSIXct`).
`names`	Optional character vector with the names of the series. If shorter than the number of series, the remaining ones take the default names `"#1"`, `"#2"`, ... A single string with blank-separated names is also accepted.
`trim`	If `TRUE` (default), rows at the beginning and at the end of the sample where all the series are missing are removed.

Details

The dates, when available, are kept in the attribute "dates" of the matrix (one per row) and are used in the time axes of the plots and in the tables of the forecasting tools. They are:

- Date for ts objects with frequency 1, 4 or 12 (first day of each year, quarter or month), for zoo/xts objects indexed by Date and for data.frame objects with a Date column;
- POSIXct for date-time indices or columns;
- numbers (the time of the ts object, or a numeric zoo index) in the other cases.

Dates must be increasing and without repetitions.

Value

A numeric matrix with column names and, for dated inputs, the attribute "dates".

Examples

```
as_tsx_matrix(airpas)[1:3, , drop = FALSE]
as_tsx_matrix(cbind(a = 1:5, b = c(NA, 2:5)))
d <- data.frame(date = as.Date("2024-01-01") + 0:4, x = c(3, 1, 4, 1, 5))
attr(as_tsx_matrix(d), "dates")
```

---

electricity *Electricity data*

Description

Two high frequency series (without dates or names).

Usage

```
electricity
```

Format

A numeric matrix with 2472 rows and columns y1 and y2.

faccuracy *Forecast accuracy measures*

Description

Error measures of one or several forecasts of one series over the forecasting period (observations after `origin`), and measures of the uncertainty of their prediction intervals when `stdpy` is given.

Usage

```
faccuracy(
  y,
  py,
  stdpy = NULL,
  origin = 0,
  series = 1,
  level = 0.95,
  names = NULL
)
```

Arguments

`y`	Actual values (see `as_tsx_matrix()`); it may include the in-sample period.
`py`	Forecasts: a matrix with `n * m` columns, being `m` the number of series in `y` and `n` the number of forecasts; forecast `f` of series `s` is in column `(f - 1) * m + s`. `y` and `py` must be of the same type (an error is raised otherwise): both without dates (vectors, matrices or numeric `data.frames`; aligned at the end), both `ts` with the same frequency (aligned on their common time grid), or both `zoo`, both `xts` or both `data.frame` with a date column (aligned by their dates, which must be of the same class). See `as_tsx_matrix()` for the dated inputs.
`stdpy`	Optional standard deviations of the forecasts, with the layout of `py` (the standard deviation of forecast `f` of series `s` is in column `(f - 1) * m + s`; missing columns at the end are taken as not available). The prediction intervals are `py -/+ z * stdpy`, with `z = qnorm((1 + level) / 2)`. With dated inputs it may be of the same type as `py` or have no dates and the rows of `py`.
`origin`	Forecast origin: number of in-sample observations. If `y` is longer than `py`, it is set to `nrow(y) - nrow(py)`. With dated inputs and `origin = 0`, it is the number of periods before the first date of `py` (counted from the earliest date of `y` and `py`).
`series`	Series to evaluate (index or name).
`level`	Confidence level of the prediction intervals computed from `stdpy` (default 0.95); the significance level is $\alpha = 1-$ `level`.
`names`	Optional names of the series.

Details

Point forecast measures, with $e_t = F_t - y_t$:

- Bias and MAE: mean error and mean absolute error.
- Bias+MAE: $|Bias| + MAE$.
- RMSE: root mean squared error.
- MASE: MAE relative to the in-sample MAE of the one step naive forecast (only when `origin > 0`).

Interval measures (only with `stdpy`), with $\alpha = 1-$ `level` and the bounds of the Gaussian prediction intervals $l_t = F_t - z\sigma_t$ and $u_t = F_t + z\sigma_t$, being σ_t the standard deviation of the forecast (`stdpy`) and z the $1 - \alpha/2$ quantile of the standard normal distribution (1.96 for `level = 0.95`):

- Coverage: proportion of actual values inside $[l_t, u_t]$ (to be compared with `level`).
- Range: mean width $u_t - l_t$.
- MIS (mean interval score): mean of $(u_t - l_t) + \frac{2}{\alpha}(l_t - y_t)1(y_t < l_t) + \frac{2}{\alpha}(y_t - u_t)1(y_t > u_t)$.
- Pinball loss of the lower bound (quantile $\tau = \alpha/2$) and of the upper bound ($\tau = 1 - \alpha/2$): mean of $\tau(y_t - q_t)$ if $y_t \geq q_t$ and $(1 - \tau)(q_t - y_t)$ otherwise.

If no period after `origin` has both an actual value and a forecast (the forecasts do not overlap the actual values), an error of class `tsx_no_overlap` is raised. The same applies to `fcompare()`; `fplot()` plots these data without problems.

Value

An object of class `tsx_faccuracy` with fields `call`, `series`, `table` (data.frame, one row per forecast, with columns `forecast`, `n`, `bias`, `mae`, `bias_mae`, `rmse`, `mase`, `coverage`, `range`, `mis`, `pinball_lower` and `pinball_upper`; the interval measures are `NA` without `stdpy`) and `by_horizon` (data.frame with the same measures accumulated up to each period of the forecasting sample, with columns `forecast`, `t`, `date` (only for dated inputs), `actual`, `fcst`, `e` and the measures) and `dates` (dates of the forecasting sample for dated inputs, otherwise `NULL`).

References

Hyndman, R.J. and Koehler, A.B. (2006). Another look at measures of forecast accuracy. *International Journal of Forecasting*, 22, 679-688.

Gneiting, T. and Raftery, A.E. (2007). Strictly proper scoring rules, prediction, and estimation. *Journal of the American Statistical Association*, 102, 359-378.

Svetunkov, I. (2023). *Forecasting and Analytics with the Augmented Dynamic Adaptive Model (ADAM)*. Chapman and Hall/CRC. https://openforecast.org/adam/

Examples

```
# data without dates (forecasts aligned with the end of y)
y <- as.vector(log(airpas))
f1 <- y[121:132]          # naive seasonal forecast
f2 <- rep(y[132], 12)     # naive forecast
faccuracy(y, cbind(f1, f2))
```

```
# with 95% intervals of the first forecast (standard deviation 0.05)
faccuracy(y, f1, rep(0.05, 12))
# data with dates (both of the same type), aligned by their dates
d <- seq(as.Date("1949-01-01"), by = "month", length.out = 144)
yd <- data.frame(date = d, y = y)
fd <- data.frame(date = d[133:144], f1 = f1, f2 = f2)
faccuracy(yd, fd)
```

`fcompare` *Forecast comparison tests*

Description

For every pair of forecasts of one series, tests of the null hypothesis of equal accuracy based on squared errors (MSE) and on absolute errors (MAE): Diebold-Mariano (with `trunc` autocovariances in the long run variance of the loss differential), sign test and Wilcoxon signed rank test (exact for up to 15 non-zero differences, normal approximation without continuity correction otherwise). An error of class `tsx_no_overlap` is raised when the actual values and the forecasts do not overlap (see `faccuracy()`).

Usage

```
fcompare(y, py, origin = 0, trunc = 1, series = 1, names = NULL)
```

Arguments

`y`	Actual values (see `as_tsx_matrix()`); it may include the in-sample period.
`py`	Forecasts: a matrix with `n * m` columns, being `m` the number of series in `y` and `n` the number of forecasts; forecast `f` of series `s` is in column `(f - 1) * m + s`. `y` and `py` must be of the same type (an error is raised otherwise): both without dates (vectors, matrices or numeric `data.frames`; aligned at the end), both `ts` with the same frequency (aligned on their common time grid), or both `zoo`, both `xts` or both `data.frame` with a date column (aligned by their dates, which must be of the same class). See `as_tsx_matrix()` for the dated inputs.
`origin`	Forecast origin: number of in-sample observations. If `y` is longer than `py`, it is set to `nrow(y) - nrow(py)`. With dated inputs and `origin = 0`, it is the number of periods before the first date of `py` (counted from the earliest date of `y` and `py`).
`trunc`	Truncation lag of the long run variance of the Diebold-Mariano test (number of autocovariances). Default 1.
`series`	Series to evaluate (index or name).
`names`	Optional names of the series.

Value

An object of class `tsx_fcompare` with fields `call`, `series`, `dates` (dates of the forecasting sample for dated inputs, otherwise `NULL`) and `table`, a data.frame with one row per pair of forecasts and loss function: `f1`, `f2`, `loss` (`"mse"` or `"mae"`), `dm`, `p_dm`, `sign`, `p_sign`, `wilcoxon` (sum of the ranks of the positive differences) and `p_wilcoxon`.

References

Diebold, F.X. and Mariano, R.S. (1995). Comparing predictive accuracy. *Journal of Business and Economic Statistics*, 13, 253-263.

Wilcoxon, F. (1945). Individual comparisons by ranking methods. *Biometrics Bulletin*, 1, 80-83.

Examples

```
y <- as.vector(log(airpas))
fcompare(y, cbind(y[121:132], rep(y[132], 12)))
```

`fplot` *Plot of actual values and forecasts*

Description

Actual values of one series with one or all its forecasts, their confidence bands and the forecast errors, optionally for a subperiod. The returned object holds the data to plot; `plot()` draws it.

Usage

```
fplot(
  y,
  py,
  stdpy = NULL,
  origin = 0,
  series = 1,
  forecast = 1,
  s = 12,
  subsample = 0,
  level = 0.95,
  names = NULL
)
```

Arguments

`y`	Actual values (see `as_tsx_matrix()`); it may include the in-sample period.
`py`	Forecasts: a matrix with `n * m` columns, being `m` the number of series in `y` and `n` the number of forecasts; forecast `f` of series `s` is in column `(f - 1) * m + s`. `y` and `py` must be of the same type (an error is raised otherwise): both without dates (vectors, matrices or numeric `data.frames`; aligned at the end), both `ts` with the same frequency (aligned on their common time grid), or both `zoo`, both `xts` or both `data.frame` with a date column (aligned by their dates, which must be of the same class). See `as_tsx_matrix()` for the dated inputs.

stdpy	Optional standard deviations of the forecasts, with the layout of py (the standard deviation of forecast f of series s is in column (f - 1) * m + s; missing columns at the end are taken as not available). The prediction intervals are py -/+ z * stdpy, with z = qnorm((1 + level) / 2). With dated inputs it may be of the same type as py or have no dates and the rows of py.
origin	Forecast origin: number of in-sample observations. If y is longer than py, it is set to nrow(y) - nrow(py). With dated inputs and origin = 0, it is the number of periods before the first date of py (counted from the earliest date of y and py).
series	Series to evaluate (index or name).
forecast	Forecast to plot (1..n) or "all".
s	Length of the subperiods for the navigation (default 12).
subsample	0 for the full sample or k for the k-th subperiod of length s.
level	Confidence level of the prediction intervals computed from stdpy (default 0.95); the significance level is $\alpha = 1-$ level.
names	Optional names of the series.

Value

An object of class tsx_fplot with fields call, series, time, actual, forecasts (matrix), lower, upper (matrices of bands py -/+ z * stdpy, or NULL without stdpy), errors (forecast minus actual for one forecast, absolute errors for all), origin and dates (date of each row for dated inputs, otherwise NULL).

Examples

```
y <- as.vector(log(airpas))
p <- fplot(y, cbind(y[121:132], rep(y[132], 12)), forecast = "all")
plot(p)
# dated forecasts after the end of the data (aligned by their dates)
y <- log(airpas)
fut <- ts(rep(y[144], 12), start = c(1961, 1), frequency = 12)
plot(fplot(y, fut, s = 24, subsample = 6))
# with 90% intervals (standard deviations growing with the horizon)
plot(fplot(y, fut, 0.03 * sqrt(1:12), level = 0.9, s = 24, subsample = 6))
```

---

furnace *Gas furnace data*

Description

Box-Jenkins series J: output CO2 concentration and input gas rate of a gas furnace, sampled every 9 seconds.

Usage

```
furnace
```

Format

A numeric matrix with 296 rows and columns `co2` and `gas_rate`.

Source

Box, G.E.P., Jenkins, G.M. and Reinsel, G.C. (2008). *Time Series Analysis, Forecasting and Control*. Wiley.

---

gaselec — *Gas and electricity data*

Description

Two monthly series from January 1992 (unnamed).

Usage

```
gaselec
```

Format

A monthly `mts` object with 213 observations of `y1` and `y2`.

---

gausstest — *Gaussianity tests*

Description

Bera-Jarque and Kolmogorov-Smirnov-Lilliefors tests of normality for every series and, for more than one series, the multivariate Bera-Jarque test.

Usage

```
gausstest(y, digits = 4, names = NULL)
```

Arguments

`y`	Time series data (see `as_tsx_matrix()`).
`digits`	Number of decimals for printing.
`names`	Optional names of the series.

Details

The Lilliefors p-values are approximate: they are interpolated (and linearly extrapolated, clamped to `[0, 1]`) in the table of critical values of Lilliefors (1967) for p = 0.20, 0.15, 0.10, 0.05 and 0.01.

Value

An object of class `tsx_gausstest` with fields `call`, `series`, `table` (data.frame with columns `series`, `n`, `skewness`, `kurtosis` (excess), `bj`, `p_bj`, `ksl`, `p_ksl`) and, for more than one series, `bj_multi` and `p_bj_multi`.

References

Jarque, C.M. and Bera, A.K. (1987). A test for normality of observations and regression residuals. *International Statistical Review*, 55, 163-172.

Lilliefors, H.W. (1967). On the Kolmogorov-Smirnov test for normality with mean and variance unknown. *Journal of the American Statistical Association*, 62, 399-402.

Examples

```
gausstest(airpas)
gausstest(log(airpas))
```

---

johansentest *Johansen cointegration tests*

Description

Maximum eigenvalue and trace statistics of Johansen (1988) for the number of cointegrating relations of a VAR model in levels with `lag` lags, under three specifications of the deterministic terms (none, constant, constant and linear trend). For one series the test is a unit root test.

Usage

```
johansentest(y, lags = NULL, names = NULL)
```

Arguments

`y`	Time series data (see `as_tsx_matrix()`).
`lags`	Lag order(s) of the VAR in levels (the VECM includes `lag - 1` lagged differences). Default: `2:min(floor(sqrt(T)), 20)` for one series (lag 1 is removed) and `min(floor(sqrt(T)), 20)` for several.
`names`	Optional names of the series.

Details

P-values are approximate, interpolated (and linearly extrapolated, clamped to `[0, 1]`) in tables of critical values for up to 5 non-stationary directions; beyond that they are `NA`. The model without deterministic terms uses the tables of Osterwald-Lenum (1992, Table 0) and the model with an unrestricted constant those of Johansen and Juselius (1990, Table A1). There are no tables for the model with an unrestricted linear trend, whose p-values are `NA`.

Value

An object of class `tsx_johansentest` with fields `call`, `series` and

- `table`: data.frame with columns `lag`, `type` (`"max"` or `"trace"`), `h0`, `h1` (number of cointegrating relations under the null and the alternative), `stat_none`, `p_none`, `stat_const`, `p_const`, `stat_trend`, `p_trend`,
- `eigen`: data.frame with columns `lag`, `i`, `eigen_none`, `lr_none`, `eigen_const`, `lr_const`, `eigen_trend`, `lr_trend` (eigenvalues and $-T\log(1-\lambda_i)$).

References

Johansen, S. (1988). Statistical analysis of cointegration vectors. *Journal of Economic Dynamics and Control*, 12, 231-254.

Johansen, S. and Juselius, K. (1990). Maximum likelihood estimation and inference on cointegration. *Oxford Bulletin of Economics and Statistics*, 52, 169-210.

Osterwald-Lenum, M. (1992). A note with quantiles of the asymptotic distribution of the maximum likelihood cointegration rank test statistics. *Oxford Bulletin of Economics and Statistics*, 54, 461-472.

Examples

```
johansentest(log(lydia_pinkham))
```

---

lydia_pinkham *Lydia Pinkham advertising and sales*

Description

Annual advertising expenditure and sales of the Lydia E. Pinkham Medicine Company, 1907-1960.

Usage

```
lydia_pinkham
```

Format

An annual `mts` object with 54 observations of `advertising` and `sales`.

Source

Palda, K.S. (1964). *The Measurement of Cumulative Advertising Effects*. Prentice-Hall.

meda — *Median absolute deviation with respect to the median*

Description

Median absolute deviation with respect to the median

Usage

```
meda(y)
```

Arguments

y	Time series data (see as_tsx_matrix()).

Value

Numeric vector with the MEDA of each series (missing values are ignored).

Examples

```
meda(airpas)
```

nile — *Nile river flow*

Description

Annual flow of the river Nile at Aswan, 1871-1970.

Usage

```
nile
```

Format

An annual ts object with 100 observations.

Source

Cobb, G.W. (1978). The problem of the Nile: conditional solution to a changepoint problem. *Biometrika*, 65, 243-251.

`nltests` *Non-linearity tests based on squares*

Description

McLeod-Li (portmanteau on the ACF of the squared series), Monti (on the PACF of the squares) and Pena-Rodriguez (determinant of the autocorrelation matrix of the squares) tests.

Usage

```
nltests(y, ncoef = NULL, series = 1, names = NULL)
```

Arguments

`y`	Time series data (see `as_tsx_matrix()`).
`ncoef`	Number of lags. Default `min(38, ceiling(T / 4 / m * 2))`.
`series`	Series to analyse (index or name).
`names`	Optional names of the series.

Value

An object of class `tsx_nltests` with fields `call`, `series` and `table`, a data.frame with columns `lag`, `acf`, `se_acf`, `q_mll`, `p_mll`, `pacf`, `se_pacf`, `q_mt`, `p_mt`, `d_pr`, `p_pr` (statistics and p-values of the three tests for each lag).

References

McLeod, A.I. and Li, W.K. (1983). Diagnostic checking ARMA time series models using squared-residual autocorrelations. *Journal of Time Series Analysis*, 4, 269-273.

Monti, A.C. (1994). A proposal for a residual autocorrelation test in linear models. *Biometrika*, 81, 776-780.

Pena, D. and Rodriguez, J. (2002). A powerful portmanteau test of lack of fit for time series. *Journal of the American Statistical Association*, 97, 601-610.

Examples

```
nltests(vdif(log(airpas), c(1, 1), c(1, 12)), ncoef = 12)
```

plotband *Time series plot with confidence bands*

Description

Plot of one series with its mean and mean +/- 2 standard deviations, of a subsample of length s, of all the subsamples of length s overlaid, or of all the series together (optionally standardised).

Usage

```
plotband(
  y,
  s = 12,
  standard = FALSE,
  fixed = TRUE,
  series = 1,
  subsample = 0,
  overlay = FALSE,
  names = NULL
)
```

Arguments

y	Time series data (see as_tsx_matrix()).
s	Length of the subsamples (seasonal period). Default 12.
standard	For series = "all", plot the standardised series.
fixed	Keep the vertical scale of the full sample when plotting a subsample.
series	One series (index or name) or "all".
subsample	0 for the full sample, k for the k-th subsample of length s (observations (k - 1) * s + 1 to k * s).
overlay	For one series, plot all the subsamples of length s overlaid (one line per subsample).
names	Optional names of the series.

Value

An object of class tsx_plotband with fields call, series, data (the data to plot: a matrix whose rows are the time indices time), time, dates (dates of time for dated inputs, NULL otherwise or with overlay), mean, sd and ylim.

Examples

```
plot(plotband(airpas))
plot(plotband(airpas, overlay = TRUE))
plot(plotband(scc, series = "all", standard = TRUE))
```

scc — *SCC data*

Description

Three series (without dates or names).

Usage

```
scc
```

Format

A numeric matrix with 62 rows and columns `y1`, `y2` and `y3`.

stdmean — *Standard deviation - mean plots*

Description

Splits the series into groups of `s` consecutive observations (the first `T mod s` observations are discarded) and plots a dispersion measure against a location measure of each group, with the least squares line. A positive slope suggests a transformation to stabilise the variance.

Usage

```
stdmean(y, s = 12, type = 0, numbers = TRUE, series = 1, names = NULL)
```

Arguments

`y`	Time series data (see `as_tsx_matrix()`).
`s`	Length of the groups (seasonal period). Default 12.
`type`	`0`: standard deviation vs mean; `1`: MEDA vs median; `2`: range vs mean.
`numbers`	Label each point with the number of its group in the plot.
`series`	Series to analyse (index or name).
`names`	Optional names of the series.

Value

An object of class `tsx_stdmean` with fields `call`, `series`, `table` (data.frame with columns `group`, `location`, `dispersion`) and `coef` (intercept and slope of the least squares line).

Examples

```
plot(stdmean(airpas))
plot(stdmean(log(airpas), type = 1))
```

steel — *Steel data*

Description

Example series (without dates).

Usage

```
steel
```

Format

A numeric vector with 157 observations.

sumstats — *Summary descriptive statistics*

Description

Computes for every series the number of data points, missing values, percentiles, means (arithmetic, geometric and harmonic), dispersion measures, skewness and kurtosis, a test of zero mean and, for more than one series, the correlation matrix.

Usage

```
sumstats(
  y,
  gaussian = TRUE,
  nsimul = 3000,
  digits = 4,
  seed = NULL,
  names = NULL
)
```

Arguments

`y`	Time series data (see `as_tsx_matrix()`).
`gaussian`	If `TRUE` (default) the probability of zero mean is computed from the t distribution; if `FALSE` it is estimated by bootstrap (`nsimul` replications) smoothed with a kernel density.
`nsimul`	Number of bootstrap replications when `gaussian = FALSE`.
`digits`	Number of decimals for printing.
`seed`	Optional seed for the bootstrap (the global random number generator state is restored on exit).
`names`	Optional names of the series.

Value

An object of class `tsx_sumstats` with fields:

- `call`: options used,
- `series`: names of the series,
- `stats`: data.frame with one row per series and columns `n`, `missing`, `min`, `p10`, `p25`, `mean`, `p_mean_zero`, `geometric_mean`, `harmonic_mean`, `median`, `mode`, `p75`, `p90`, `max`, `iqr`, `range`, `sd`, `var`, `mean_abs_dev`, `median_abs_dev`, `mean_sd_ratio`, `skewness`, `kurtosis` (excess kurtosis). `p_mean_zero` is the two sided p-value of the test of zero mean and `range` is `max - min`. `mode` is the most frequent value (the smallest one in case of ties; with continuous data with no repeated values it equals the minimum),
- `cor`: correlation matrix (`NULL` for one series) and `se_cor` its standard error $1/\sqrt{T}$.

Examples

```
sumstats(airpas)
sumstats(scc, names = c("A", "B", "C"))
```

---

testhet *Heteroskedasticity tests*

Description

Variance ratio F tests between non-overlapping subsamples of each series.

Usage

```
testhet(y, timecuts = NULL, names = NULL)
```

Arguments

`y`	Time series data (see `as_tsx_matrix()`).
`timecuts`	Definition of the subsamples: • a vector of `n` cut points divides the sample into `n + 1` consecutive subsamples (cut points closer than 5 observations to the ends are discarded), • a `n x 2` matrix gives the first and last observation of each subsample (overlapping or invalid rows are discarded). If no valid subsamples remain, the sample is split into three parts. Default `c(floor(T / 3), floor(2 * T / 3) + 2)`.
`names`	Optional names of the series.

Value

An object of class `tsx_testhet` with fields `call`, `series`, `samples` (matrix with first and last observation of each subsample) and `table`, a data.frame with columns `series`, `start1`, `end1`, `start2`, `end2`, `f` (ratio of variances, always <= 1) and `p_value` (two sided).

Examples

```
testhet(airpas)
testhet(log(airpas), c(48, 96))
testhet(airpas, rbind(c(1, 30), c(50, 90), c(100, 144)))
```

`toolforecast` *Interactive forecasting tool*

Description

Graphical interface for the evaluation of forecasts: plot of actual values and forecasts with confidence bands (computed from the standard deviations of the forecasts) and forecast errors, navigation by subperiods, and tables of accuracy measures (`faccuracy()`) and comparison tests (`fcompare()`). When the forecasts do not overlap the actual values (e.g. true out-of-sample forecasts) the table shows only the forecasts (and the bounds of their intervals), without actual values nor accuracy measures.

Usage

```
toolforecast(
  y,
  py,
  stdpy = NULL,
  origin = 0,
  names = NULL,
  level = 0.95,
  gui = c("tcltk", "shiny"),
  launch.browser = TRUE
)
```

Arguments

`y`	Actual values (see `as_tsx_matrix()`); it may include the in-sample period.
`py`	Forecasts: a matrix with `n * m` columns, being `m` the number of series in `y` and `n` the number of forecasts; forecast `f` of series `s` is in column `(f - 1) * m + s`. `y` and `py` must be of the same type (an error is raised otherwise): both without dates (vectors, matrices or numeric `data.frames`; aligned at the end), both `ts` with the same frequency (aligned on their common time grid), or both `zoo`, both `xts` or both `data.frame` with a date column (aligned by their dates, which must be of the same class). See `as_tsx_matrix()` for the dated inputs.
`stdpy`	Optional standard deviations of the forecasts, with the layout of `py` (the standard deviation of forecast `f` of series `s` is in column `(f - 1) * m + s`; missing columns at the end are taken as not available). The prediction intervals are `py -/+ z * stdpy`, with `z = qnorm((1 + level) / 2)`. With dated inputs it may be of the same type as `py` or have no dates and the rows of `py`.

origin	Forecast origin: number of in-sample observations. If y is longer than py, it is set to nrow(y) - nrow(py). With dated inputs and origin = 0, it is the number of periods before the first date of py (counted from the earliest date of y and py).
names	Optional names of the series.
level	Confidence level of the prediction intervals computed from stdpy (default 0.95), used in the plot and in the interval measures of faccuracy(). It can also be changed in the *Options* of the interface.
gui	"tcltk" (default) or "shiny".
launch.browser	Passed to shiny::runApp() (only for Shiny).

Details

The actual values and the forecasts must be of the same type, without dates or with dates (ts, zoo, xts or data.frame with a date column; see faccuracy()); otherwise an error is raised. With dates, the plots show them on the time axis and the tables include them.

Two interfaces with the same behaviour are available: gui = "tcltk" (default; a window with the menus *Series*, *Forecasts* and *Options*, the navigation buttons and the tables, with the plot in an R graphics window) and gui = "shiny" (a web application, needs the package shiny). With tcltk its text and graphics windows are independent of those of tooltest() and are shared with the results of faccuracy(), fcompare() and fplot() printed at the console (see tsexplorer-package): a new call reuses them and brings them to the front.

Value

For Shiny, called for its side effect (runs the application). For tcltk, invisibly, an environment with the state of the interface (series, forecast, s, subsample, trunc, level, last_call, result and tables).

Examples

```
if (interactive()) {
  y <- as.vector(log(airpas))
  py <- cbind(y[121:132], rep(y[132], 12))
  toolforecast(y, py)
  toolforecast(y, py, gui = "shiny")
  # with 90% intervals of the first forecast (its standard deviations)
  toolforecast(y, py, rep(0.05, 12), level = 0.9)
  # with dates (actual values and forecasts of the same type)
  d <- seq(as.Date("1949-01-01"), by = "month", length.out = 144)
  toolforecast(data.frame(date = d, y = y), data.frame(date = d[133:144], py))
}
```

tooltest — *Interactive test tool*

Description

Graphical interface with a menu with all the exploratory, descriptive and diagnostic tools of the package, a selector of series, the options of each tool and the graphical and text outputs. Every result shown can be obtained without the GUI by calling the corresponding function (e.g. `vident()`) and `print()` / `plot()`.

Usage

```
tooltest(
  y,
  names = NULL,
  ...,
  gui = c("tcltk", "shiny"),
  launch.browser = TRUE
)
```

Arguments

`y`	Time series data (see `as_tsx_matrix()`).
`names`	Optional names of the series.
`...`	Options per tool, as named lists, e.g. `vident = list(ncoef = 50, s = 4, npar = 1)` or `varspectrum = list(16)` (see `tooltest_defaults()`).
`gui`	`"tcltk"` (default) or `"shiny"`.
`launch.browser`	Passed to `shiny::runApp()` (only for Shiny).

Details

The data may have no dates or have them (`ts`, `zoo`, `xts` or `data.frame` with a date column; see `as_tsx_matrix()`); with dates, the plots with a time axis show them.

Two interfaces with the same behaviour are available:

- `gui = "tcltk"` (default): a window with the menus *Tests*, *Series* and *Options*, navigation buttons and the text output, with the plots in an R graphics window. It only needs the Tcl/Tk support included in R. Its text and graphics windows are independent of those of `toolforecast()` and are shared with the results of its tools printed at the console (see tsexplorer-package): a new call reuses them and brings them to the front.
- `gui = "shiny"`: a web application (needs the package `shiny`).

Value

For Shiny, called for its side effect (runs the application). For tcltk, invisibly, an environment with the state of the interface: `tool`, `opts`, `last_call` and `result` (the object shown).

Examples

```
if (interactive()) {
  tooltest(airpas)
  tooltest(airpas, gui = "shiny")
  tooltest(vdif(log(lydia_pinkham)), vident = list(ncoef = 12))
  d <- seq(as.Date("1949-01-01"), by = "month", length.out = 144)
  tooltest(data.frame(date = d, passengers = as.vector(airpas)))
}
```

---

tooltest_defaults *Default options of the tools of tooltest()*

Description

Single source of the default options of every tool (they depend on the sample size `T` and the number of series `m`). It is used by the `tooltest()` application; the functions of the package resolve the same defaults when their arguments are `NULL`.

Usage

```
tooltest_defaults(y, ...)
```

Arguments

`y`	Time series data (see `as_tsx_matrix()`).
`...`	Options to override, given as named lists per tool, e.g. `vident = list(ncoef = 50, s = 4, npar = 1)`. Unnamed elements are assigned in the order of the options of the tool, e.g. `varspectrum = list(16)`. When a linked option is changed in one tool, it is propagated to the other tools of its group.

Details

Some options are linked across tools: the seasonal period `s` (`plotband`, `vboxplot`, `vtboxplot`, `vident`, `vident2`, `stdmean` and the `group` of `vboxcox`), the number of lags of the correlograms (`vident`, `vident2`, `varsbc`), the lags of the unit root tests (`adftest`, `vphillips`, `johansentest`) and the number of lags of the non-linearity tests (`nltests`, `varsbc2`). The groups are returned in the attribute `"links"`.

Value

A named list (one element per tool) of named lists of options, with attribute `"links"`.

Examples

```
str(tooltest_defaults(airpas)$vident)
tooltest_defaults(airpas, vboxplot = list(s = 4))$plotband$s
```

unemployment *Unemployment data*

Description

Two monthly series from January 2000 (unnamed).

Usage

```
unemployment
```

Format

A monthly `mts` object with 124 observations of `y1` and `y2`.

usgdp *US GDP*

Description

US gross domestic product, quarterly data (without dates).

Usage

```
usgdp
```

Format

A numeric vector with 241 observations.

varsbc *Information criteria of AR and VAR models*

Description

Corrected Akaike (AICc), Schwarz (SBC) and Hannan-Quinn (HQC) information criteria of AR (one series) or VAR (several series) models of orders `0:ncoef`, and the likelihood ratio test of each order against the previous one.

Usage

```
varsbc(y, ncoef = NULL, series = 1, names = NULL)
```

Arguments

y	Time series data (see as_tsx_matrix()).
ncoef	Maximum order. Default min(38, ceiling(T / 4 / m * 2)), limited to T - 3.
series	Series to analyse: one series (index or name) for AR models, several or "all" for a VAR model.
names	Optional names of the series.

Value

An object of class tsx_varsbc with fields call, series (series analysed), table (data.frame with columns lag, aicc, sbc, hqc, chi, p_chi) and best (order minimising each criterion).

References

Lütkepohl, H. (2005). *New Introduction to Multiple Time Series Analysis*. Springer.

Examples

```
r <- varsbc(vdif(log(airpas), c(1, 1), c(1, 12)), ncoef = 15)
r
plot(r)
```

---

varsbc2 — *Non-linearity test based on the Schwarz criterion of squares*

Description

Schwarz information criterion of AR models of orders 0:ncoef fitted to the squares of the series. A minimum at order 0 suggests linearity.

Usage

```
varsbc2(y, ncoef = NULL, series = 1, names = NULL)
```

Arguments

y	Time series data (see as_tsx_matrix()).
ncoef	Maximum order. Default min(38, ceiling(sqrt(T))), limited to T - 3.
series	Series to analyse (index or name).
names	Optional names of the series.

Value

An object of class tsx_varsbc2 with fields call, series, table (data.frame with columns lag and sbc), best (order with minimum SBC) and linear (TRUE when best is 0).

Examples

```
plot(varsbc2(vdif(log(airpas), c(1, 1), c(1, 12))))
```

`varspectrum` *Autoregressive spectrum*

Description

Spectrum of an AR model of order `order` (with constant) fitted by least squares to one series, $f(\omega) = \sigma^2/(2\pi)|1/\phi(e^{-i\omega})|^2$, in the same scale as the periodogram of `vspectrum()`.

Usage

```
varspectrum(
  y,
  order = 16,
  nfreq = NULL,
  xlog = c(FALSE, TRUE),
  series = 1,
  names = NULL
)
```

Arguments

`y`	Time series data (see `as_tsx_matrix()`).
`order`	Order of the AR model. Default 16 (limited to `T / 2`).
`nfreq`	Number of frequencies in `[0, pi)`. `0` or `NULL` uses the Fourier frequencies `2 pi j / T`; values below 10 are replaced by `floor(T / 2)`. Default `floor(T / 2)`.
`xlog`	Logical vector of length 2: logarithmic scale for the frequency (period) axis and for the power axis in the plot. Default `c(FALSE, TRUE)`.
`series`	Series to analyse (index or name).
`names`	Optional names of the series.

Value

An object of class `tsx_varspectrum` with fields `call`, `series`, `ar` (estimated AR coefficients), `sigma2` (innovation variance) and `table` (data.frame with columns `freq`, radians, and `spectrum`).

Examples

```
plot(varspectrum(vdif(log(airpas)), order = 14))
```

vbds *BDS test of independence*

Description

Brock, Dechert and Scheinkman (BDS) test of the null hypothesis that a series is independent and identically distributed, for embedding dimensions `2:maxdim`. P-values are asymptotic (standard normal).

Usage

```
vbds(y, maxdim = 2, distance = 1.5, names = NULL)
```

Arguments

`y`	Time series data (see `as_tsx_matrix()`).
`maxdim`	Maximum embedding dimension (at least 2). Default 2.
`distance`	Dimensional distance as a fraction of the standard deviation of each series. Default 1.5. It is reduced to `min(0.9 * (max(y) - mean(y)) / sd(y))` (minimum over the series) if larger than that.
`names`	Optional names of the series.

Details

The computations follow the implementation of Ludwig Kanzler (1998), rewritten without the bitwise storage that program uses to save memory.

Value

An object of class `tsx_vbds` with fields `call` (effective `maxdim` and `distance`), `series` and `table`, a data.frame with columns `series`, `dim`, `w` (BDS statistic) and `p_value`.

References

Brock, W.A., Dechert, W.D., Scheinkman, J.A. and LeBaron, B. (1996). A test for independence based on the correlation dimension. *Econometric Reviews*, 15, 197-235.

Kanzler, L. (1999). Very fast and correctly sized estimation of the BDS statistic. Working paper, University of Oxford.

Examples

```
vbds(vdif(log(airpas), c(1, 1), c(1, 12)), maxdim = 4)
```

vboxcox *Box-Cox transformation*

Description

Estimates the Box-Cox parameter λ of each series with the method of Guerrero (1993): the series is divided into groups of `group` consecutive observations and λ minimises the coefficient of variation of $s_j/\bar{z}_j^{1-\lambda}$ (standard deviation over mean of each group). It also evaluates, over a grid of values of λ (`axislambdas`), the standard deviation, the mean over standard deviation and the criterion of the transformed series, and returns the transformed data.

Usage

```
vboxcox(
  y,
  lambdas = NULL,
  group = 12,
  axislambdas = seq(-1, 1, by = 0.1),
  series = 1,
  names = NULL
)
```

Arguments

`y`	Time series data (see `as_tsx_matrix()`).
`lambdas`	Optional values of λ for the transformation of each series (the last one is repeated for the remaining series). If `NULL` they are estimated.
`group`	Number of observations per group (usually the seasonal period). Default 12.
`axislambdas`	Grid of λ values for the plot.
`series`	Series used for the grid evaluation and the plot.
`names`	Optional names of the series.

Details

Series with non-positive values are shifted by `abs(min) + 1e-4` before the transformation.

Value

An object of class `tsx_vboxcox` with fields `call`, `series`, `lambdas` (estimated or supplied, one per series), `estimated` (logical), `transformed` (matrix of transformed series) and `grid`, a data.frame for the selected series with columns `lambda`, `sd`, `mean_sd`, `criterion` (minus the coefficient of variation), and their standardised versions `sd_std`, `mean_sd_std`, `criterion_std` (as plotted).

References

Box, G.E.P. and Cox, D.R. (1964). An analysis of transformations. *Journal of the Royal Statistical Society B*, 26, 211-252.

Guerrero, V.M. (1993). Time-series analysis supported by power transformations. *Journal of Forecasting*, 12, 37-48.

Examples

```
r <- vboxcox(airpas)
r$lambdas
plot(r)
```

`vboxplot` *Box plot of a time series by groups of* `s` *consecutive observations*

Description

For each group of `s` consecutive observations (e.g. each year of monthly data) shows the mean, median, 25% and 75% percentiles, minimum and maximum. Optionally highlights the observations of one position within the groups (e.g. one month across years).

Usage

```
vboxplot(y, s = 12, series = 1, highlight = 0, names = NULL)
```

Arguments

`y`	Time series data (see `as_tsx_matrix()`).
`s`	Length of the groups (seasonal period). Default 12, limited to `T / 2`.
`series`	Series to analyse (index or name).
`highlight`	`0` for none, or the position `1..s` within the groups whose values are highlighted.
`names`	Optional names of the series.

Value

An object of class `tsx_vboxplot` with fields `call`, `series`, `stats` (data.frame with columns `group`, `mean`, `median`, `p25`, `p75`, `min`, `max`), `highlight` (values highlighted, one per group) and `group_dates` (date of the first observation of each group for dated inputs, otherwise `NULL`).

Examples

```
plot(vboxplot(airpas, highlight = 7))
```

vcumperiod — *Cumulative periodogram*

Description

Normalised cumulative periodogram of one series with the 1% and 5% Kolmogorov-Smirnov bands of the white noise hypothesis.

Usage

```
vcumperiod(y, series = 1, names = NULL)
```

Arguments

y	Time series data (see as_tsx_matrix()).
series	Series to analyse (index or name).
names	Optional names of the series.

Value

An object of class tsx_vcumperiod with fields call, series, table (data.frame with columns freq, frequency in cycles per period from 0 to 0.5, and cum_periodogram) and bands (half width of the 1% and 5% bands).

References

Box, G.E.P., Jenkins, G.M., Reinsel, G.C. and Ljung, G.M. (2015). *Time Series Analysis: Forecasting and Control*, 5th ed. Wiley.

Examples

```
plot(vcumperiod(vdif(log(airpas), c(1, 1), c(1, 12))))
```

vcusum — *CUSUM and CUSUM of squares tests*

Description

Cumulative sums of the standardised series (CUSUM) and of its squared deviations from the mean (CUSUMSQ), with 5% confidence bands.

Usage

```
vcusum(y, series = 1, names = NULL)
```

Arguments

y	Time series data (see as_tsx_matrix()).
series	Series to analyse (index or name).
names	Optional names of the series.

Value

An object of class tsx_vcusum with fields call, series and table, a data.frame with columns t, cusum, cusum_lower, cusum_upper, cusumsq, cusumsq_lower, cusumsq_upper, and dates (dates of the rows for dated inputs, otherwise NULL).

References

Brown, R.L., Durbin, J. and Evans, J.M. (1975). Techniques for testing the constancy of regression relationships over time. *Journal of the Royal Statistical Society B*, 37, 149-192.

Examples

```
plot(vcusum(vdif(log(airpas), c(1, 1), c(1, 12))))
```

---

vdif — *Differencing of a vector of time series*

Description

Applies multiplicative difference operators $\prod_j (1 - B^{s_j})^{d_j}$ to each series.

Usage

```
vdif(y, order = 1, s = 1)
```

Arguments

y	Time series data (see as_tsx_matrix()).
order	Difference orders. A vector applies the same multiplicative polynomials to every series; a matrix with one row per series allows different orders for each series (the last row is recycled).
s	Seasonal period of each polynomial (same layout as order).

Value

Matrix of differenced data. The first r observations are lost, being r the largest total order of differencing among the series.

Examples

```
y <- log(airpas)
head(vdif(y, c(1, 1), c(1, 12)))
head(vdif(cbind(y, y), rbind(c(1, 0), c(1, 1)), c(1, 12)))
```

`vgranger` *Granger causality tests*

Description

Wald tests (F and chi-squared versions) of Granger non-causality on a VAR model without constant estimated on the demeaned series.

Usage

```
vgranger(y, lags = NULL, tests = NULL, names = NULL)
```

Arguments

`y`	Time series data (see `as_tsx_matrix()`), at least two series.
`lags`	Lags of the VAR model (block constraints are possible, e.g. `c(1, 2, 12)`). Default `1:min(floor(sqrt(T)), 20)`. If the largest lag exceeds `T / 2`, `1:2` is used.
`tests`	Character vector of tests (or a single blank-separated string). Default: every pair of series in both directions. Invalid tests are ignored.
`names`	Optional names of the series.

Details

Each test is written with letters for the series (`a` is the first one, `b` the second, ...): the letters before `-` are the equations and the letters after `-` the possibly causing variables. For instance `"a-b"` tests whether `b` causes `a`, and `"ab-c"` whether `c` causes `a` and `b`.

Value

An object of class `tsx_vgranger` with fields `call`, `series` and `table`, a data.frame with columns `test`, `h0`, `constraints`, `f`, `p_f`, `chi2`, `p_chi2`.

References

Granger, C.W.J. (1969). Investigating causal relations by econometric models and cross-spectral methods. *Econometrica*, 37, 424-438.

Examples

```
vgranger(vdif(log(lydia_pinkham)), lags = 1:3)
```

vhist — *Histogram with gaussian and non-parametric density estimates*

Description

Histogram with gaussian and non-parametric density estimates

Usage

```
vhist(y, bins = NULL, series = 1, names = NULL)
```

Arguments

y	Time series data (see as_tsx_matrix()).
bins	Number of bins. Default ceiling(sqrt(T)).
series	Series to analyse (index or name).
names	Optional names of the series.

Value

An object of class tsx_vhist with fields call, series, hist (data.frame with bin mid, lower, upper and count) and density (data.frame with x, normal and kernel, both scaled to counts).

Examples

```
plot(vhist(vdif(log(airpas), c(1, 1), c(1, 12))))
```

victims — *Road casualties*

Description

Monthly number of car drivers killed or seriously injured in Great Britain, January 1969 to December 1984.

Usage

```
victims
```

Format

A monthly ts object with 192 observations.

Source

Harvey, A.C. and Durbin, J. (1986). The effects of seat belt legislation on British road casualties: a case study in structural time series modelling. *Journal of the Royal Statistical Society A*, 149, 187-227.

vident *Sample autocorrelation and partial autocorrelation functions*

Description

Univariate sample ACF and PACF of one series, with their standard errors (Bartlett for the ACF, $1/\sqrt{T}$ for the PACF), the Ljung-Box portmanteau test on the ACF and the Monti (1994) test on the PACF.

Usage

```
vident(y, ncoef = NULL, s = 12, npar = 0, series = 1, names = NULL)
```

Arguments

y	Time series data (see as_tsx_matrix()).
ncoef	Number of lags. Default min(38, ceiling(T / 4 / m * 2)), limited to T - 5.
s	Seasonal period, only used to highlight the seasonal lags in the plot.
npar	Number of estimated parameters when y is a set of residuals; the degrees of freedom of the portmanteau tests are lag - npar.
series	Series to analyse (index or name).
names	Optional names of the series.

Value

An object of class tsx_vident with fields call, series (name of the series analysed) and table, a data.frame with columns lag, acf, se_acf, q_lb, p_lb, pacf, se_pacf, q_mt, p_mt.

References

Ljung, G.M. and Box, G.E.P. (1978). On a measure of lack of fit in time series models. *Biometrika*, 65, 297-303.

Monti, A.C. (1994). A proposal for a residual autocorrelation test in linear models. *Biometrika*, 81, 776-780.

Examples

```
r <- vident(vdif(log(airpas), c(1, 1), c(1, 12)))
r
plot(r)
```

vident2 *Multivariate sample autocorrelation functions*

Description

Sample autocorrelation matrices and partial lag autocorrelation matrices (Heyse and Wei, 1985) or partial autoregression matrices (Tiao and Box, 1981) of a vector of time series, together with the information criteria, the likelihood ratio statistic of adding one lag to a VAR model and a multivariate portmanteau test.

Usage

```
vident2(y, ncoef = NULL, s = 12, npar = 0, type = 0, names = NULL)

## S3 method for class 'tsx_vident2'
plot(x, series = 1, ...)
```

Arguments

y	Time series data (see as_tsx_matrix()), at least two series.
ncoef	Number of lags. Default min(38, ceiling(T / 4 / m * 2)).
s	Seasonal period (only for plots).
npar	Number of estimated parameters (if y are residuals); the degrees of freedom of the portmanteau test are m^2 (lag - npar).
type	0 for partial lag autocorrelation matrices, 1 for partial autoregression matrices.
names	Optional names of the series.
x	An object of class tsx_vident2.
series	For the plot, one series (index or name) to plot its ACF and partial function, or two series to plot their cross correlations.
...	Not used.

Value

An object of class tsx_vident2 with fields

- call, series,
- acf, se_acf, pacf, se_pacf: arrays m x m x (ncoef + 1) for lags 0:ncoef. Element [a, b, k + 1] is the correlation between series a at time t and series b at time t - k. At lag 0 the partial matrices repeat the correlation matrix,
- stats: data.frame with columns lag, aic, sbc, chi, p_chi (likelihood ratio test of a VAR(lag) against a VAR(lag - 1)), p_q (multivariate portmanteau test of Hosking (1980) with the correction of Li and McLeod (1981), $T\sum_k tr(R_k' R_0^{-1} R_k R_0^{-1}) + m^2 k(k+1)/(2T)$) and det_sigma.

References

Tiao, G.C. and Box, G.E.P. (1981). Modeling multiple time series with applications. *Journal of the American Statistical Association*, 76, 802-816.

Heyse, J.F. and Wei, W.W.S. (1985). The partial lag autocorrelation function. Technical Report 32, Temple University.

Hosking, J.R.M. (1980). The multivariate portmanteau statistic. *Journal of the American Statistical Association*, 75, 602-608.

Li, W.K. and McLeod, A.I. (1981). Distribution of the residual autocorrelations in multivariate ARMA time series models. *Journal of the Royal Statistical Society B*, 43, 231-239.

Examples

```
r <- vident2(vdif(log(lydia_pinkham)), ncoef = 8)
r
plot(r, series = c(1, 2))
```

`vphillips` *Phillips-Perron unit root tests*

Description

Phillips-Perron Z_t statistics computed on the same three regressions as `adftest()` (no deterministic terms, constant, constant and trend, each one including `lag` lagged differences). The long run variance of the residuals is estimated with a Bartlett (Newey-West) window with `ncovs` autocovariances.

Usage

```
vphillips(y, lags = NULL, ncovs = NULL, names = NULL)
```

Arguments

`y`	Time series data (see `as_tsx_matrix()`).
`lags`	Number(s) of lagged differences included in the regressions. Default: `1:min(floor(sqrt(T)), 20)` for one series and `min(floor(sqrt(T)), 20)` for several series. If the largest lag exceeds `T / 2` the test is computed only for `floor(T / 2)` lags.
`ncovs`	Number of autocovariances of the residuals in the long run variance. Default `floor(4 * (T / 100)^0.25)`.
`names`	Optional names of the series.

Value

An object of class `tsx_vphillips` with fields `call`, `series`, `n` and the data.frames `none`, `const` and `trend` with columns `series`, `lag`, `rho`, `z_t` (Phillips-Perron statistic) and `p_value`.

References

Phillips, P.C.B. and Perron, P. (1988). Testing for a unit root in time series regression. *Biometrika*, 75, 335-346.

Newey, W.K. and West, K.D. (1987). A simple, positive semi-definite, heteroskedasticity and autocorrelation consistent covariance matrix. *Econometrica*, 55, 703-708.

Examples

```
vphillips(log(airpas), lags = 1:3)
```

`vqqplot` *Normal QQ plot*

Description

Empirical quantiles of one series against the quantiles of a normal distribution with the same mean and standard deviation, with the line through the 25% and 75% quartiles.

Usage

```
vqqplot(y, series = 1, names = NULL)
```

Arguments

`y`	Time series data (see `as_tsx_matrix()`).
`series`	Series to analyse (index or name).
`names`	Optional names of the series.

Value

An object of class `tsx_vqqplot` with fields `call`, `series`, `table` (data.frame with `theoretical` and `empirical` quantiles), `quartiles` (x and y coordinates of the quartiles) and `line` (end points of the reference line).

Examples

```
plot(vqqplot(vdif(log(airpas), c(1, 1), c(1, 12))))
```

vscatter — *Scatter plots*

Description

Scatter plot of two series with the least squares line or, for `series = "all"`, the matrix of scatter plots with histograms in the diagonal.

Usage

```
vscatter(y, bins = NULL, series = c(1, 2), names = NULL)
```

Arguments

`y`	Time series data (see `as_tsx_matrix()`), at least two series.
`bins`	Number of bins of the histograms. Default `ceiling(sqrt(T))`.
`series`	Two series (indices or names; the first one on the horizontal axis) or `"all"`.
`names`	Optional names of the series.

Value

An object of class `tsx_vscatter` with fields `call`, `series`, `data` (matrix of the series) and `coef`, a data.frame with the least squares lines of every ordered pair: `x`, `y`, `intercept`, `slope`.

Examples

```
plot(vscatter(scc))
plot(vscatter(scc, series = "all"))
```

vspectrum — *Spectrum, cross spectrum, gains and coherence*

Description

Periodogram of one series, smoothed with a triangular window of length `smooth`, or, for two series, the gains and squared coherence computed from the smoothed cross periodogram. The periodogram is $|\sum_t z_t e^{-i\omega t}|^2/(2\pi T)$ at the Fourier frequencies; near the frequencies 0 and π the smoothing window uses the symmetry of the periodogram.

Usage

```
vspectrum(y, smooth = 1, xlog = c(FALSE, TRUE), series = 1, names = NULL)
```

Arguments

y	Time series data (see as_tsx_matrix()).
smooth	Length of the triangular smoothing window (an even value is increased by one; 1 means no smoothing). Default 1.
xlog	Logical vector of length 2: logarithmic scale for the frequency (period) axis and for the power axis in the plot. Default c(FALSE, TRUE).
series	One series (index or name) or two series for the cross spectrum.
names	Optional names of the series.

Value

An object of class tsx_vspectrum with fields call, series and table, a data.frame with column freq (radians, from 0 to pi) and

- one series: periodogram and spectrum (smoothed),
- two series: cross_amplitude (modulus of the smoothed cross spectrum), spectrum1, spectrum2 (smoothed), gain12 (cross amplitude over the spectrum of the first series), gain21 and coherence (squared coherence).

Examples

```
plot(vspectrum(vdif(log(airpas)), smooth = 5))
plot(vspectrum(vdif(log(lydia_pinkham)), smooth = 7, series = c(1, 2)))
```

---

vtboxplot *Transpose box plot of a time series*

Description

Standardises each group of s consecutive observations and shows, for each position 1..s within the groups (e.g. each month), the mean, median, 25% and 75% percentiles, minimum and maximum across groups. Optionally highlights one group (e.g. one year).

Usage

```
vtboxplot(y, s = 12, series = 1, highlight = 0, names = NULL)
```

Arguments

y	Time series data (see as_tsx_matrix()).
s	Length of the groups (seasonal period). Default 12, limited to T / 2.
series	Series to analyse (index or name).
highlight	0 for none, or the group 1..ceiling(T / s) to highlight.
names	Optional names of the series.

Value

An object of class `tsx_vtboxplot` with fields `call`, `series`, `stats` (data.frame with columns `position`, `mean`, `median`, `p25`, `p75`, `min`, `max`), `data` (matrix `s x groups` of standardised values) and `highlight`.

Examples

```
plot(vtboxplot(airpas, highlight = 12))
```

---

vtsay — *Tsay (1986) non-linearity test*

Description

Tsay (1986) F test: the residuals e_t of an AR(`lags`) model with constant are regressed on the cross products $z_{t-i}z_{t-j}$ ($i \leq j$), previously orthogonalised with respect to the constant and the lags. With $k = p(p+1)/2$ products and n observations,

$$F = \frac{(\sum e_t^2 - \sum \varepsilon_t^2)/k}{\sum \varepsilon_t^2/(n-p-k-1)}$$

follows an $F(k, n-p-k-1)$ distribution under linearity. Missing values remove the corresponding rows.

Usage

```
vtsay(y, lags = 4, names = NULL)
```

Arguments

`y`	Time series data (see `as_tsx_matrix()`).
`lags`	Order of the AR model (at least 2). Default 4.
`names`	Optional names of the series.

Value

An object of class `tsx_vtsay` with fields `call`, `series`, `table` (data.frame with columns `series`, `f`, `df1`, `df2`, `p_value`) and `regressions` (data.frame with the auxiliary regression of the AR residuals on the orthogonalised products: `series`, `variable`, `beta`, `t`, `p_value`, two sided).

References

Tsay, R.S. (1986). Nonlinearity tests for time series. *Biometrika*, 73, 461-466.

Examples

```
vtsay(vdif(log(airpas), c(1, 1), c(1, 12)))
```

Index

∗ **datasets**
    airpas, 5
    electricity, 6
    furnace, 11
    gaselec, 12
    lydia_pinkham, 14
    nile, 15
    scc, 18
    steel, 19
    unemployment, 25
    usgdp, 25
    victims, 34

adfcritical, 3
adfcritical(), *4*
adftest, 4
adftest(), *37*
airpas, 5
as_tsx_matrix, 5
as_tsx_matrix(), *4*, *7*, *9*, *10*, *12*, *13*, *15–21*, *23*, *24*, *26–41*

electricity, 6

faccuracy, 7
faccuracy(), *9*, *21*, *22*
fcompare, 9
fcompare(), *8*, *21*, *22*
fplot, 10
fplot(), *8*, *22*
furnace, 11

gaselec, 12
gausstest, 12

johansentest, 13

lydia_pinkham, 14

meda, 15
nile, 15
nltests, 16

plot.tsx_vident2 *(*vident2*)*, 36
plotband, 17

scc, 18
shiny::runApp(), *22*, *23*
stdmean, 18
steel, 19
sumstats, 19

testhet, 20
toolforecast, 21
toolforecast(), *23*
tooltest, 23
tooltest(), *22*, *24*
tooltest_defaults, 24
tooltest_defaults(), *23*
tsexplorer-package, *22*, *23*

unemployment, 25
usgdp, 25

varsbc, 25
varsbc2, 26
varspectrum, 27
vbds, 28
vboxcox, 29
vboxplot, 30
vcumperiod, 31
vcusum, 31
vdif, 32
vgranger, 33
vhist, 34
victims, 34
vident, 35
vident(), *23*
vident2, 36
vphillips, 37
vqqplot, 38

vscatter, 39
vspectrum, 39
vspectrum(), *27*
vtboxplot, 40
vtsay, 41