# Package: tstests (via r-universe)

October 25, 2024

**Type** Package

**Title** Time Series Goodness of Fit and Forecast Evaluation Tests

**Version** 1.0.1

**Maintainer** Alexios Galanos <alexios@4dscape.com>

**Depends** R (>= 3.5.0), methods, tsmethods

**Description** Goodness of Fit and Forecast Evaluation Tests for timeseries models. Includes, among others, the Generalized Method of Moments (GMM) Orthogonality Test of Hansen (1982), the Nyblom (1989) parameter constancy test, the sign-bias test of Engle and Ng (1993), and a range of tests for value at risk and expected shortfall evaluation.

**Imports** data.table, flextable, Rdpack, car, ks, xts

**License** GPL-2

**Encoding** UTF-8

**LazyData** true

**RdMacros** Rdpack

**RoxygenNote** 7.3.2

**URL** https://www.nopredict.com/packages/tstests, https://github.com/tsmodels/tstests

**Suggests** knitr, rmarkdown, sandwich, testthat (>= 3.0.0), tsdistributions, tsgarch

**VignetteBuilder** knitr

**Config/testthat/edition** 3

**NeedsCompilation** no

**Author** Alexios Galanos [aut, cre, cph] (<https://orcid.org/0009-0000-9308-0457>)

**Repository** CRAN

**Date/Publication** 2024-10-24 23:00:02 UTC

# Contents



---

`arma_forecast` *Sample ARMA Forecast Data*

---

## Description

A pre-computed backtest of the SPY log returns data using an ARMA(1,1)-JSU model (see details for replication code).

## Usage

```
arma_forecast
```

## Format

`arma_forecast`**:**

A data.table with 250 rows and 5 columns:

**date** the forecast date

**actual** the realized values

**forecast** the forecast mu

**sigma** the estimated sigma

**skew** the estimated skew of the jsu distribution

**shape** the estimated shape of the jsu distribution

## Details

The replication code for the backtest based 1-step ahead forecast distribution is as follows:

```
library(xts)
library(tsarma)
# from the tsmodels github repo
data("spy", package = "tstests")
spyr <- na.omit(diff(log(spy)))
n <- NROW(spyr)
spec <- arma_modelspec(spyr, order c(1,1), distribution = "jsu")
b <- tsbacktest(spec, start = (n - 250), end = n, h = 1, estimate_every = 30,
rolling = T, trace = T)
arma_forecast <- data.table(date = b$table$forecast_date,
actual = b$table$actual, forecast = b$table$mu, sigma = b$table$sigma,
skew = b$table$skew, shape = b$table$shape)
```

---

`as_flextable.tstest.berkowitz`

*Transform a summary object into flextable*

---

## Description

Transforms a "tstest.test" object into a flextable with options on symbolic representation and model equation.

## Usage

```
## S3 method for class 'tstest.berkowitz'
as_flextable(
  x,
  digits = max(3L, getOption("digits") - 3L),
  signif.stars = getOption("show.signif.stars"),
  include.decision = FALSE,
  table.caption = x$test_name,
  footnote.reference = FALSE,
  ...
)

## S3 method for class 'tstest.dac'
as_flextable(
  x,
  digits = max(3L, getOption("digits") - 3L),
  signif.stars = getOption("show.signif.stars"),
  include.decision = FALSE,
  table.caption = x$test_name,
  footnote.reference = FALSE,
```

```
  ...
)

## S3 method for class 'tstest.gmm'
as_flextable(
  x,
  digits = max(3L, getOption("digits") - 3L),
  signif.stars = getOption("show.signif.stars"),
  include.decision = FALSE,
  collapse = TRUE,
  use.symbols = TRUE,
  table.caption = x$test_name,
  footnote.reference = FALSE,
  ...
)

## S3 method for class 'tstest.hongli'
as_flextable(
  x,
  digits = max(3L, getOption("digits") - 3L),
  signif.stars = getOption("show.signif.stars"),
  include.decision = FALSE,
  table.caption = x$test_name,
  footnote.reference = FALSE,
  ...
)

## S3 method for class 'tstest.minzar'
as_flextable(
  x,
  digits = max(3L, getOption("digits") - 3L),
  signif.stars = getOption("show.signif.stars"),
  include.decision = FALSE,
  table.caption = x$test_name,
  footnote.reference = FALSE,
  ...
)

## S3 method for class 'tstest.nyblom'
as_flextable(
  x,
  digits = max(3L, getOption("digits") - 3L),
  signif.stars = getOption("show.signif.stars"),
  include.decision = FALSE,
  use.symbols = TRUE,
  table.caption = x$test_name,
  footnote.reference = FALSE,
  ...
```

```
)

## S3 method for class 'tstest.shortfall_de'
as_flextable(
  x,
  digits = max(3L, getOption("digits") - 3L),
  signif.stars = getOption("show.signif.stars"),
  include.decision = FALSE,
  table.caption = x$test_name,
  footnote.reference = FALSE,
  ...
)

## S3 method for class 'tstest.signbias'
as_flextable(
  x,
  digits = max(3L, getOption("digits") - 3L),
  signif.stars = getOption("show.signif.stars"),
  include.decision = FALSE,
  use.symbols = TRUE,
  table.caption = x$test_name,
  footnote.reference = FALSE,
  ...
)

## S3 method for class 'tstest.vares'
as_flextable(
  x,
  digits = max(3L, getOption("digits") - 3L),
  signif.stars = getOption("show.signif.stars"),
  include.decision = FALSE,
  table.caption = x$test_name,
  footnote.reference = FALSE,
  ...
)

## S3 method for class 'tstest.var_cp'
as_flextable(
  x,
  digits = max(3L, getOption("digits") - 3L),
  signif.stars = getOption("show.signif.stars"),
  include.decision = FALSE,
  table.caption = x$test_name,
  footnote.reference = FALSE,
  ...
)
```

### Arguments

| | |
|---|---|
| x | an object of which inherits a "tstest" class. |
| digits | integer, used for number formatting. Optionally, to avoid scientific notation, set 'options(scipen=999)'. |
| signif.stars | logical. If TRUE, 'significance stars' are printed. |
| include.decision | prints out whether to reject the NULL at the 5% level of significance. |
| table.caption | an optional string for the table caption. |
| footnote.reference | whether to include the reference paper of the test in the footnotes. |
| ... | not currently used. The returned object can be manipulated further using flextable. |
| collapse | collapses the results for multiple lags to just report the joint test. |
| use.symbols | for tests which either have parameters for which the latex symbols were included in the calling function or for which the tests generate values which can be represented as latex symbols, then these will be generated. |

### Value

A flextable object.

---

## berkowitz_test *Berkowitz Forecast Density Test*

---

### Description

The forecast density test of Berkowitz (2001).

### Usage

```
berkowitz_test(x, lags = 1, ...)
```

### Arguments

| | |
|---|---|
| x | a series representing the PIT transformed actuals given the forecast values. |
| lags | the number of autoregressive lags (positive and greater than 0). |
| ... | additional arguments passed to the arima function which estimates the unrestricted model. |

### Value

An object of class "tstest.berkowitz" which has a print and as_flextable method.

## References

Berkowitz J (2001). "Testing density forecasts, with applications to risk management." *Journal of Business & Economic Statistics*, **19**(4), 465–474.

Jarque CM, Bera AK (1987). "A test for normality of observations and regression residuals." *International Statistical Review/Revue Internationale de Statistique*, 163–172.

## Examples

```
library(tsdistributions)
data(garch_forecast)
x <- pdist('jsu', q = garch_forecast$actual, mu = garch_forecast$forecast,
sigma = garch_forecast$sigma, skew = garch_forecast$skew,
shape = garch_forecast$shape)
print(berkowitz_test(x))
```

---

# dac_test *Directional Accuracy Tests*

---

## Description

The directional accuracy test of Pesaran and Timmermann (1992), and excess profitability test of Anatolyev and Gerko (2005).

## Usage

```
dac_test(actual, forecast, ...)
```

## Arguments

| | |
|---|---|
| actual | a series representing the actual value of the series in the out of sample period. |
| forecast | the forecast values of the series in the out of sample period. |
| ... | not currently used. |

## Details

The null hypothesis for the test of Pesaran and Timmermann (1992) is that the actual and predicted are independent (no sign predictability), whereas the test of Anatolyev and Gerko (2005) measures the significance of the excess profitability under the null hypothesis of no excess excess profitability. Both are Hausman type tests asymptotically distributed as standard Normal.

## Value

An object of class "tstest.dac" which has a print and as_flextable method.

## Note

The test will not work with constant forecasts.

## References

Pesaran,M.H., Timmermann,A. (1992). "A simple nonparametric test of predictive performance." *Journal of Business & Economic Statistics*, **10**(4), 461–465.

Anatolyev,S., Gerko,A. (2005). "A trading approach to testing for predictability." *Journal of Business & Economic Statistics*, **23**(4), 455–461.

## Examples

```
data(arma_forecast)
print(dac_test(arma_forecast$actual, arma_forecast$forecast))
```

---

`garch_forecast` *Sample GARCH Forecast Data*

---

## Description

A pre-computed backtest of the SPY log returns data using a GARCH(1,1)-JSU model (see details for replication code).

## Usage

```
garch_forecast
```

## Format

`garch_forecast`**:**

A data.table with 250 rows and 5 columns:

**date** the forecast date

**actual** the realized values

**forecast** the forecast mu

**sigma** the forecast sigma

**skew** the estimated skew of the jsu distribution

**shape** the estimated shape of the jsu distribution

## Details

The replication code for the backtest based 1-step ahead forecast distribution is as follows:

```
library(xts)
library(tsgarch)
data("spy", package = "tstests")
spyr <- na.omit(diff(log(spy)))
n <- NROW(spyr)
spec <- garch_modelspec(spyr, model = "garch", constant = T,
distribution = "jsu")
```

```
b <- tsbacktest(spec, start = (n - 250), end = n, h = 1, estimate_every = 30,
rolling = T, trace = T)
garch_forecast <- data.table(date = b$table$forecast_date,
actual = b$table$actual, forecast = b$table$mu, sigma = b$table$sigma,
skew = b$table$skew, shape = b$table$shape)
```

---

`gmm_test` *GMM Orthogonality Test*

---

## Description

The GMM orthogonality test of Hansen (1982).

## Usage

```
gmm_test(x, lags = 1, skewness = 0, kurtosis = 3, ...)
```

## Arguments

| | |
|---|---|
| `x` | a series representing the standardized residuals of some estimated model. |
| `lags` | the lags for the co-moment test. |
| `skewness` | the skewness of the estimated model residuals. |
| `kurtosis` | the kurtosis of the estimated model residuals. |
| `...` | not currently used. |

## Details

For parametric models estimated with a particular distribution, the skewness and kurtosis should flow from the distributional model. See for example `dskewness` and `dkurtosis`.

## Value

An object of class "tstest.gmm" which has a print and as_flextable method.

## References

Hansen,L.P. (1982). "Large sample properties of generalized method of moments estimators." *Econometrica*, **50**(4), 1029–1054.

## Examples

```
library(tsgarch)
library(tsdistributions)
library(data.table)
library(xts)
data("spy")
spyr <- na.omit(diff(log(spy)))
spec <- garch_modelspec(spyr, model = "egarch", order = c(2,1), constant = TRUE,
distribution = "jsu")
mod <- estimate(spec)
skewness <- dskewness("jsu", skew = coef(mod)["skew"], shape = coef(mod)["shape"])
# kurtosis is dkurtosis is the excess over the Normal (3) so we add back 3
# since the test takes the actual not excess kurtosis.
kurtosis <- dkurtosis("jsu", skew = coef(mod)["skew"], shape = coef(mod)["shape"]) + 3
test <- gmm_test(residuals(mod, standardize = TRUE), lags = 2, skewness = skewness,
kurtosis = kurtosis)
print(test, collapse = TRUE, include.decision = TRUE)
```

---

# hongli_test *The Non-Parametric Density Test of Hong and Li*

---

## Description

Implements the Non-Parametric Density Test of Hong and Li (2005).

## Usage

```
hongli_test(x, lags = 4, conf_level = 0.95, ...)
```

## Arguments

| | |
|---|---|
| x | a series representing the PIT transformed actuals given the forecast values. |
| lags | the number lags to use for testing the joint hypothesis. |
| conf_level | the confidence level for generating the critical values which serve as thresholds for deciding on the null hypothesis. |
| ... | none. |

## Details

A novel method to analyze how well a conditional density fits the underlying data is through the probability integral transformation (PIT) discussed in Rosenblatt (1952) and used in the berkowitz_test. Hong and Li (2005) introduced a nonparametric portmanteau test, building on the work of Ait-Sahalia (1996), which tests the joint hypothesis of i.i.d and uniformity for a series of PIT transformed data. To achieve this, it tests for misspecification in the conditional moments of the model transformed standardized residuals, and is distributed as N(0, 1) under the null of a correctly specified model. These moment tests are reported as 'M(1,1)' to 'M(4,4)' in the output, with 'M(1,2)'

related to ARCH-in-mean effects, and 'M(2,1)' to leverage, while 'W' is the Portmanteu type test statistic for general misspecification (using p lags) and also distributed as N(0, 1) under the null of a correctly specified model. Only upper tail critical values are used in this test. The interested reader is referred to the paper for more details.

## Value

An object of class "tstest.hongli" which has a print and "as_flextable" method.

## References

Hong,Y., Li,H. (2005). "Nonparametric specification testing for continuous-time models with applications to term structure of interest rates." *Review of Financial Studies*, **18**(1), 37–84.

## Examples

```
library(tsdistributions)
data(garch_forecast)
x <- pdist('jsu', q = garch_forecast$actual, mu = garch_forecast$forecast,
sigma = garch_forecast$sigma, skew = garch_forecast$skew,
shape = garch_forecast$shape)
print(hongli_test(x), include.decision = TRUE)
```

---

# minzar_test *Mincer-Zarnowitz Test*

---

## Description

The forecast unbiasedness test of Mincer and Zarnowitz (1969).

## Usage

```
minzar_test(actual, forecast, ...)
```

## Arguments

| | |
|---|---|
| actual | a vector representing the actual values of a series. |
| forecast | a vector representing the forecasted values of the series. |
| ... | additional arguments passed to linearHypothesis, except the "test" argument which is fixed to Chisq. |

## Value

An object of class "tstest.minzar" which has a print and as_flextable method.

## References

Mincer JA, Zarnowitz V (1969). "The evaluation of economic forecasts." In *Economic forecasts and expectations: Analysis of forecasting behavior and performance*, 3–46. NBER.

## Examples

```
data(arma_forecast)
test <- minzar_test(arma_forecast$actual, arma_forecast$forecast)
test
```

---

`nyblom_test` *Nyblom-Hansen Parameter Constancy Test*

---

## Description

The parameter constancy test of Nyblom (1989).

## Usage

```
nyblom_test(
  x,
  scores = NULL,
  parameter_names = colnames(scores),
  parameter_symbols = NULL,
  ...
)
```

## Arguments

| | |
|---|---|
| `x` | a series representing the standardized residuals of some estimated model. |
| `scores` | the log likelihood score matrix. The `estfun` method if exported by a package for a model will return this matrix. |
| `parameter_names` | optional character vector of the parameter names. Usually read off the column names of the score matrix. |
| `parameter_symbols` | an optional character vector of the latex names of the parameters which can be used when printing using the flextable format. |
| `...` | not currently used. |

## Details

The p-values for the test statistic are based on a pre-computed density, by simulation using equation 3.3 of Nyblom (1989), with up to 40 parameters and saved as an internal data object within the package. A kernel density is used to fit the 10,000 samples of the distribution before extracting the p-values. The original simulation generated more than 100,000 data points but these were compressed to quantiles at intervals of 0.001 in order to keep the package size under 5MB.

## Value

An object of class "tstest.nyblom" which has a print and as_flextable method.

## References

Nyblom,J. (1989). "Testing for the constancy of parameters over time." *Journal of the American Statistical Association*, **84**(405), 223–230.

## Examples

```
library(tsgarch)
library(xts)
data("spy")
spyr <- na.omit(diff(log(spy)))
spec <- garch_modelspec(spyr[1:1200], model = "garch", order = c(1,1),
constant = TRUE, distribution = "norm")
mod <- estimate(spec)
test <- nyblom_test(residuals(mod, standardize = TRUE), scores = estfun(mod),
parameter_names = names(coef(mod)),
parameter_symbols = mod$parmatrix[estimate == 1]$symbol)
print(test)
```

---

# print.tstest.berkowitz

*Test Print method*

---

## Description

Print method for objects inheriting class "tstest"

## Usage

```
## S3 method for class 'tstest.berkowitz'
print(
  x,
  digits = max(3L, getOption("digits") - 3L),
  signif.stars = getOption("show.signif.stars"),
  include.decision = FALSE,
  ...
)

## S3 method for class 'tstest.dac'
print(
  x,
  digits = max(3L, getOption("digits") - 3L),
  signif.stars = getOption("show.signif.stars"),
  include.decision = FALSE,
```

```
  ...
)

## S3 method for class 'tstest.gmm'
print(
  x,
  digits = max(3L, getOption("digits") - 3L),
  signif.stars = getOption("show.signif.stars"),
  include.decision = FALSE,
  collapse = TRUE,
  ...
)

## S3 method for class 'tstest.hongli'
print(
  x,
  digits = max(3L, getOption("digits") - 3L),
  signif.stars = getOption("show.signif.stars"),
  include.decision = FALSE,
  ...
)

## S3 method for class 'tstest.minzar'
print(
  x,
  digits = max(3L, getOption("digits") - 3L),
  signif.stars = getOption("show.signif.stars"),
  include.decision = FALSE,
  ...
)

## S3 method for class 'tstest.nyblom'
print(
  x,
  digits = max(3L, getOption("digits") - 3L),
  signif.stars = getOption("show.signif.stars"),
  include.decision = FALSE,
  ...
)

## S3 method for class 'tstest.shortfall_de'
print(
  x,
  digits = max(3L, getOption("digits") - 3L),
  signif.stars = getOption("show.signif.stars"),
  include.decision = FALSE,
  ...
)
```

```
## S3 method for class 'tstest.signbias'
print(
  x,
  digits = max(3L, getOption("digits") - 3L),
  signif.stars = getOption("show.signif.stars"),
  include.decision = FALSE,
  ...
)

## S3 method for class 'tstest.vares'
print(
  x,
  digits = max(3L, getOption("digits") - 3L),
  signif.stars = getOption("show.signif.stars"),
  include.decision = FALSE,
  ...
)

## S3 method for class 'tstest.var_cp'
print(
  x,
  digits = max(3L, getOption("digits") - 3L),
  signif.stars = getOption("show.signif.stars"),
  include.decision = FALSE,
  ...
)
```

## Arguments

| | |
|---|---|
| x | an object inheriting class "tstest.test". |
| digits | integer, used for number formatting. Optionally, to avoid scientific notation, set 'options(scipen=999)'. |
| signif.stars | logical. If TRUE, 'significance stars' are printed. |
| include.decision | prints out whether to reject the NULL at the 5% level of significance. |
| ... | not currently used. |
| collapse | collapses the results for multiple lags to just report the joint test. |

## Value

Invisibly returns the original object.

`shortfall_de_test` *Expected Shortfall DE Test*

## Description

The expected shortfall test of Du and Escanciano (2017).

## Usage

```
shortfall_de_test(x, alpha = 0.05, lags = 1, boot = FALSE, n_boot = 2000, ...)
```

## Arguments

| | |
|---|---|
| `x` | the probability integral transformed series (pit). |
| `alpha` | the quantile level for calculating the forecast value at risk and expected shortfall. |
| `lags` | the numbers of lags to use for the conditional test. |
| `boot` | whether to use bootstrap simulation for estimating the p-values. |
| `n_boot` | the bootstrap replications used to calculate the p-value. |
| `...` | not currently used. |

## Details

The test of Du and Escanciano (2017) combines ideas from Berkowitz (2001) and Christoffersen (1998) to create an unconditional and conditional shortfall test based on the probability integral transformed actuals conditioned on the forecast distribution to evaluate the severity and independence of the residuals shortfall (based on violations of VaR). The unconditional test (severity) checks for the mean of cumulative violations using a t-test, whilst the conditional test (independence) is a Portmanteau test applied to estimated cumulative violations. A bootstrap approach to calculating the distribution of the test statistics is available for finite samples, similar to the suggestions of McNeil (2000).

## Value

An object of class "tstest.shortfall_de" which has a print and as_flextable method.

## References

Du Z, Escanciano JC (2017). "Backtesting expected shortfall: accounting for tail risk." *Management Science*, **63**(4), 940–958.

Berkowitz J (2001). "Testing density forecasts, with applications to risk management." *Journal of Business & Economic Statistics*, **19**(4), 465–474.

Christoffersen PF (1998). "Evaluating interval forecasts." *International Economic Review*, 841–862.

McNeil,A.J., Frey,R. (2000). "Estimation of tail-related risk measures for heteroscedastic financial time series: An extreme value approach." *Journal of Empirical Finance*, **7**(3-4), 271–300.

## Examples

```
library(tsdistributions)
data("garch_forecast")
x <- pdist("jsu", q = garch_forecast$actual, mu = garch_forecast$forecast,
sigma = garch_forecast$sigma, skew = garch_forecast$skew,
shape = garch_forecast$shape)
print(shortfall_de_test(x, alpha = 0.05, lags = 4))
```

---

`signbias_test` *Sign Bias Test*

---

## Description

The sign bias test of Engle and Ng (1993).

## Usage

```
signbias_test(x, sigma = 1, ...)
```

## Arguments

| | |
|---|---|
| `x` | a series representing the residuals of some estimated model. |
| `sigma` | either a scalar representing the residuals standard deviation else a vector of the same length as x representing the conditional standard deviation of the residuals. |
| `...` | additional arguments passed to `linearHypothesis`, except the "test" which is fixed to use the Chisq test. |

## Value

An object of class "tstest.signbias" which has a print and as_flextable method.

## References

Engle RF, Ng VK (1993). "Measuring and testing the impact of news on volatility." *The Journal of Finance*, **48**(5), 1749–1778.

## Examples

```
library(tsgarch)
library(tsdistributions)
library(xts)
data("spy")
spyr <- na.omit(diff(log(spy)))
spec <- garch_modelspec(spyr, model = "garch", order = c(1,1),
constant = TRUE, distribution = "jsu")
mod <- estimate(spec)
print(signbias_test(residuals(mod), sigma(mod)))
```

## spy *SPY ETF Adjusted Close*

### Description

The adjusted closing price of the SPY ETF.

### Usage

```
spy
```

### Format

spy:

An xts vector with 7597 observations spanning the period 1993-01-29 / 2023-03-30 from Yahoo Finance.

## var_cp_test *Value at Risk CP Test*

### Description

The value at risk coverage and duration tests of Kupiec (1995) and Christoffersen and Pelletier (1998,2004).

### Usage

```
var_cp_test(actual, forecast, alpha, ...)
```

### Arguments

| | |
|---|---|
| actual | a series representing the actual value of the series in the out of sample period. |
| forecast | the forecast values of the series at the quantile given by alpha (the forecast value at risk). |
| alpha | the quantile level used to calculate the forecast value at risk. |
| ... | not currently used. |

## Details

The unconditional (Kupiec 1995) and conditional (Christoffersen and Pelletier 1998) coverage tests evaluate the correctness and independence of value at risk violations (failures), individually and jointly. Correctness is measured in terms of the expected and actual violations of value at risk for a given quantile and data size, whilst independence checks the clustering of violations with past violations, which is key in determining whether a model can accurately capture the higher order dynamics of a series. The duration of time between value ar risk violations (no-hits) should ideally be independent and not cluster. Under the null hypothesis of a correctly specified risk model, the no-hit duration should have no memory. Since the only continuous distribution which is memory free is the exponential, the test can conducted on any distribution which embeds the exponential as a restricted case, and a likelihood ratio test then conducted to see whether the restriction holds. Following Christoffersen and Pelletier (2004), the Weibull distribution is used with parameter 'b=1' representing the case of the exponential.

## Value

An object of class "tstest.var_cp" which has a print and as_flextable method.

## References

Kupiec,P.H. (1995). "Techniques for verifying the accuracy of risk measurement models." *The Journal of Derivatives*, **3**(2), 73–84.

Christoffersen PF (1998). "Evaluating interval forecasts." *International Economic Review*, 841–862.

Christoffersen PF, Pelletier,D. (2004). "Backtesting value-at-risk: A duration-based approach." *Journal of Financial Econometrics*, **2**(1), 84–108.

## Examples

```
library(tsdistributions)
data("garch_forecast")
q <- qdist("jsu", p = 0.05, mu = garch_forecast$forecast, sigma = garch_forecast$sigma,
skew = garch_forecast$skew, shape = garch_forecast$shape)
var_cp_test(actual = garch_forecast$actual, forecast = q, alpha = 0.05)
```

---

# var_test *Value at Risk and Expected Shortfall Tests*

---

## Description

The value at risk coverage and duration tests of Kupiec (1995) and Christoffersen and Pelletier (1998,2004), and expected shortfall test of Du and Escanciano (2017).

## Usage

```
var_test(
  actual,
  forecast,
  x,
  alpha,
  lags = 1,
  boot = FALSE,
  n_boot = 2000,
  ...
)
```

## Arguments

| | |
|---|---|
| actual | a series representing the actual value of the series in the out of sample period. |
| forecast | the forecast values of the series at the quantile given by alpha (the forecast value at risk). |
| x | the probability integral transformed series (pit). |
| alpha | the quantile level used to calculate the forecast value at risk. |
| lags | the numbers of lags to use for the conditional shortfall test. |
| boot | whether to use bootstrap simulation for estimating the p-values of the conditional shortfall test. |
| n_boot | the bootstrap replications used to calculate the p-value. |
| ... | not currently used. |

## Details

This is a condensed table of both the var_cp_test and shortfall_de_test.

## Value

An object of class "tstest.vares" which has a print and as_flextable method.

## References

Kupiec,P.H. (1995). "Techniques for verifying the accuracy of risk measurement models." *The Journal of Derivatives*, **3**(2), 73–84.

Christoffersen PF (1998). "Evaluating interval forecasts." *International Economic Review*, 841–862.

Christoffersen PF, Pelletier,D. (2004). "Backtesting value-at-risk: A duration-based approach." *Journal of Financial Econometrics*, **2**(1), 84–108.

Du Z, Escanciano JC (2017). "Backtesting expected shortfall: accounting for tail risk." *Management Science*, **63**(4), 940–958.

# Index