Package 'Rita'

November 16, 2026

Type Package

Title Automated Transformations, Normality Testing, and Reporting

Version 2.0.1

Description Automatically evaluates common normalizing transformations of numeric variables and selects the best-fitting transformation for each based on squared correlation with theoretical normal quantiles. Available transformations include Rankit, arcsine, logit/log-odds, logarithmic, square-root, and inverse/reciprocal. Performs six normality tests (Thode, 2002, <doi:10.1201/9780203910894>), incorporating simulation-based p-value calibration for small-sample Jarque–Bera and optionally Lilliefors, as well as subsampling for Shapiro–Wilk when sample sizes exceed its supported limit. Available tests include Anderson–Darling, Pearson chi-square, D'Agostino–Pearson omnibus, Jarque–Bera, Kolmogorov–Smirnov with Lilliefors correction, and Shapiro–Wilk. A descriptive statistical report of the retained input data in their original units, including the five-number summary and mathematical moments, is also presented. The Rankit empirical normal quantile transformation (ENQT) (Soloman and Sawilowsky, 2009, <doi:10.22237/jmasm/1257034080>) is provided for those requiring rank-based normalization and to facilitate its adoption. Structured analytical results and paired visualizations of original and transformed distributions are also available.

License MIT + file LICENSE

Encoding UTF-8

Imports stats, lattice, grid

RoxygenNote 7.1.2

NeedsCompilation no

Author Daniel Mattei [aut, cre],
John Ruscio [aut]

Maintainer Daniel Mattei <DMattei@live.com>

Repository CRAN

Date/Publication 2026-11-16 09:30:10 UTC

Contents

ADTest . . . 2
arcsineXform . . . 3
chisqTest . . . 4
DPTest . . . 6
inverseXform . . . 7
JBTest . . . 8
KSLTest . . . 9
kurtCoeff . . . 11
logitXform . . . 12
logXform . . . 14
MasterTest . . . 15
MasterXform . . . 15
popSD . . . 16
rankitXform . . . 17
Rita . . . 18
RitaPlots . . . 22
rita_analysis . . . 23
rita_plotting . . . 24
skewCoeff . . . 26
squareXform . . . 27
SWTest . . . 28

Index **30**

`ADTest` *Anderson-Darling Test*

Description

This function computes the one-sample Anderson-Darling test statistic and p-value for fit to a normal distribution.

Usage

```
ADTest(data, alpha = 0.05, j = 1)
```

Arguments

`data`	Data of a univariate distribution for which the test statistic is computed (vector)
`alpha`	The significance threshold for the upper-tail test
`j`	The # hypotheses tested; used to compute a Bonferonni correction, if applicable; should remain at its default if multiple testing is not an issue (scalar)

Details

An adjusted statistic provided by D'agostino & Stephens (1986) is used, where the mean and variance of the population are treated as unknown. D'agostino & Stephen's (1986) text provides the equations used to obtain the function's p-values.

The modified statistic follows Table 4.7, Case 3, and the p-value follows Table 4.9. Log-CDF and log-survival probabilities are evaluated directly. Standardization uses centered, scaled coordinates. Constant observations, fewer than two non-missing observations, or infinite observations terminate testing with an error. Missing values (NA and NaN) are omitted before calculating the statistic.

At the approximation breakpoints, 0.20 uses the first formula, 0.34 uses the second, and 0.60 uses the fourth. The statistic and p-value are rounded to three decimals; significance uses the unrounded p-value and alpha / j. Per limitations of D'agostino & Stephen's approximation, extrapolation to extreme statistics can yield invalid p-values, including values above one; the returned p-value and significance flag must not be interpreted as calibrated inference in that regime.

Value

An object including the test statistic, p-value, and a significance flag (list)

Precision metadata

Successful computed results also carry a rita_test attribute with the unrounded statistic and p.value, n_tested, precision, calibration and details. It records existing computations without rerunning the test. Unsupported or undefined results retain their existing sentinel/NA representation and may lack this attribute. See rita_analysis for reporting conventions.

References

D'agostino, R. B., & Stephens, M. A. (1986). Goodness-of-fit-techniques (Vol. 68). CRC press.

Examples

```
values <- rnorm(100)
x <- ADTest(data = values)
```

`arcsineXform` *Arcsine Transformation*

Description

This function transforms the scale, if needed, to values of unity. Then, the data is transformed by taking the arcsine of the square root. Per the recommendations of Osborne(2002), data points are left-anchored at 0 to maximize the efficacy of the square-root transformation used enroute to the arcsine.

Usage

```
arcsineXform(sample)
```

Arguments

sample	Nonempty finite numeric vector, matrix, array, or data frame without missing values.

Details

Negative minima are first shifted to zero. If the resulting maximum exceeds 1, values are divided by that maximum.

Input must be nonempty and finite, with no NA or NaN. Numeric vectors, matrices, arrays and data frames are accepted; dimensions and names are retained. Bounds are computed over the whole input. Constants return zero.

Value

The arcsine-transformed data, retaining input dimensions and names

References

Osborne, J. W. (2002). Notes on the use of data transformations. Practical Assessment, Research and Evaluation, 9(1), 42-50.

Osborne, J. W. (2002). The Effects of Minimum Values on Data Transformations. Retrieved from https://files.eric.ed.gov/fulltext/ED463313.pdf

Examples

```
values <- rnorm(100)
x <- arcsineXform(values)
```

---

chisqTest *Chi-Square Test*

Description

This function computes the chi-square test for normality.

Usage

```
chisqTest(data, alpha = 0.05, j = 1, df = NULL)
```

Arguments

data	Data of a univariate distribution for which the test statistic is computed (numeric vector, or numeric matrix/array flattened to a vector)
alpha	The upper-tail significance threshold used for hypothesis-testing
j	The # hypotheses tested; used to compute a Bonferonni correction, if applicable; should remain at its default if multiple testing is not an issue (scalar)

`df` NULL (default) uses k - 3 degrees of freedom, where k is the number of bins after input cleaning. A positive, finite numeric scalar overrides the degrees of freedom used for the chi-square reference distribution.

Details

The number of cells is k = ceiling(2*n^(2/5)) (Moore, 1986). Numerically computed quantiles of the fitted normal distribution define equiprobable cells with infinite outer endpoints and expected counts of (n / k), including for empty cells. The mean and sample standard deviation are estimated from all usable observations (missing values NA and NaN are omitted). Calculations use centered and scaled coordinates to improve numerical stability. Cells are right-closed, so observations exactly on an interior boundary are assigned to the lower cell.

At least three finite observations with positive sample variance are required. Ties are permitted, although the continuous-normal reference distribution does not account for rounding or discretization. Small expected cell counts may further compromise the chi-square approximation; cells are not merged.

By default, df = NULL uses the conventional (k - 3) degrees-of-freedom approximation, where (k) is determined after missing-value removal. An explicitly supplied positive, finite df overrides the reference distribution's degrees of freedom. Because the normal parameters are estimated from ungrouped observations, the resulting chi-square reference is approximate rather than an exact null distribution (Moore, 1986, Sections 3.2.2 and 3.2.7). Significance decisions compare the unrounded upper-tail p-value with alpha / j. Displayed values are rounded.

Value

An object including the test statistic, p-value, and a significance flag (list)

Precision metadata

Successful computed results also carry a rita_test attribute with the unrounded statistic and p.value, n_tested, precision, calibration and details. It records existing computations without rerunning the test. Unsupported or undefined results retain their existing sentinel/NA representation and may lack this attribute. See rita_analysis for reporting conventions.

References

Moore, D.S., (1986) Tests of the chi-squared type. In: D'agostino, R.B. and Stephens, M.A., eds.: Goodness-of-Fit Techniques. Marcel Dekker, New York.

Examples

```
values <- rnorm(100)
x <- chisqTest(data = values)
k <- ceiling(2 * length(values)^(2/5))
approximate <- chisqTest(values, df = k - 3) # Explicit approximation, not exact.
```

DPTest *D'agostino Pearson Omnibus Test*

Description

This function computes the D'agostino Pearson omnibus test using adjusted Fisher- Pearson skewness and kurtosis estimators.

Usage

```
DPTest(data, alpha = 0.05, j = 1, warn = T)
```

Arguments

data	Data of a univariate distribution for which the test statistic is computed (vector)
alpha	The significance threshold for the upper-tail omnibus test
j	The # hypotheses tested; used to compute a Bonferonni correction, if applicable; should remain at its default if multiple testing is not an issue (scalar)
warn	Logical; emit warnings for termination at N < 8, the kurtosis approximation at N < 20, and undefined numerical results.

Details

The adjusted skewness and kurtosis coefficients are computed using the sample standard deviation in centered, scaled coordinates to improve numerical stability. These coefficients are converted to moment ratios using equations 24–25 of D'Agostino, Belanger, and D'Agostino (1990) before constructing the standardized skewness and kurtosis statistics. The omnibus p-value is calculated from the upper tail of a chi-square distribution with two degrees of freedom.

At least eight usable observations are required (N >= 8). However, the kurtosis approximation is intended for sample sizes of at least 20. Accordingly, for 8 <= N < 20, a warning is issued when warn = TRUE. The test statistic and p-value are displayed to three decimal places, while significance is determined using the unrounded p-value.

The kurtosis transformation uses a signed real cube root. If the transformation denominator is zero, the observations are constant, or intermediate calculations produce nonfinite values, the named statistic, p-value, and significance flag are returned as NA, with a warning when warn = TRUE. Missing observations (NA and NaN) are omitted before determining sample size. Infinite observations also produce NA results.

Value

An object including the test statistic, p-value, and a significance flag (list)

Precision metadata

Successful computed results also carry a rita_test attribute with the unrounded statistic and p.value, n_tested, precision, calibration and details. It records existing computations without rerunning the test. Unsupported or undefined results retain their existing sentinel/NA representation and may lack this attribute. See rita_analysis for reporting conventions.

References

D'agostino, R. B., & Stephens, M. A. (1986). Goodness-of-fit-techniques (Vol. 68). CRC press.

D'Agostino, R. B., Belanger, A., & D'Agostino, R. B., Jr. (1990). A Suggestion for Using Powerful and Informative Tests of Normality. *The American Statistician, 44*(4), 316-321. https://doi.org/10.2307/2684359

Shreve, Joni N. and Donna Dea Holland . 2018. SAS® Certification Prep Guide: Statistical Business Analysis Using SAS®9. Cary, NC: SAS Institute Inc.

Examples

```
values <- rnorm(100)
x <- DPTest(data = values)
```

---

inverseXform *Inverse/Reciprocal Transformation*

Description

This function imputes minimum values per the recommendations of Osborne (2002) and reflects the input (multiplies by -1), anchors its minimum at 1, and takes the reciprocal.

Usage

```
inverseXform(sample)
```

Arguments

sample	Nonempty finite numeric vector, matrix, array, or data frame without missing values.

Details

Computes the reflected inverse transformation 1 / ((max(sample) - sample) + 1), with the minimum of the reflected data anchored at 1 before taking reciprocals. The transformation preserves the direction of the original scale.

Input must be nonempty and finite, with no NA or NaN. The maximum is taken over the whole input. Constants return one.

Value

The reciprocal-transformed data, retaining input dimensions and names

References

Osborne, J. W. (2002). Notes on the use of data transformations. Practical Assessment, Research and Evaluation, 9(1), 42-50.

Osborne, J. W. (2002). The Effects of Minimum Values on Data Transformations. Retrieved from https://files.eric.ed.gov/fulltext/ED463313.pdf

Examples

```
values <- rnorm(100)
x <- inverseXform(values)
```

JBTest *Jarque-Bera Test*

Description

Tests normality using the adjusted Fisher-Pearson skewness and excess kurtosis.

Usage

```
JBTest(data, alpha = 0.05, j = 1, N_Sample = 10000, warn = T)
```

Arguments

data	Numeric observations for one univariate distribution. NA and NaN are omitted and numeric matrices and arrays are flattened.
alpha	Upper-tail significance level.
j	Number of hypotheses in the Bonferroni adjustment. Default 1.
N_Sample	Positive integer number of normal-null replicates for 4 <= n < 2000. Default 10000.
warn	Logical controlling warnings for simulation and termination below four usable observations.

Details

With n usable observations and adjusted skewness and excess kurtosis coefficients (G1) and (G2), the test statistic is JB = (n / 6) (G1^2 + G2^2 / 4) (Jarque & Bera, 1980). Both coefficients are calculated using the sample standard deviation. Fewer than four usable observations produce a failure result, with a diagnostic message printed when warn = T. Infinite observations and undefined variance at supported sample sizes produce errors.

For 4 <= n < 2000, the p-value is estimated using N_Sample independent samples from the standard normal distribution. Each simulated sample is standardized using its own mean and sample standard

deviation before calculating the statistic. The Monte Carlo upper-tail p-value is (1 + exceedances) / (N_Sample + 1), with ties counted as exceedances. Use set.seed() before calling the function to reproduce results under the same R random-number generation configuration. Simulation advances the caller's RNG state. For n >= 2000, the p-value uses an asymptotic chi-square approximation with two degrees of freedom. Both approaches assume independent observations from a continuous normal distribution under the null hypothesis.

Significance is determined by comparing the unrounded p-value with the strict threshold alpha / j. With the default 10,000 simulation replicates, the minimum attainable Monte Carlo p-value is 1 / 10001, and the approximate simulation standard error at p = 0.05 is 0.00218. More stringent significance thresholds or greater precision may require additional replicates.

Value

A three-element list of named scalars for the statistic, p-value (both rounded to three decimals), and significance flag. For a simulated result the monte_carlo attribute records n, n_sim, exceedances, min_p and the unrounded p.value. A displayed zero can represent a positive probability below 0.0005. For n < 4 the return is list(rep("F", 3)).

Precision metadata

Successful computed results also carry a rita_test attribute with the unrounded statistic and p.value, n_tested, precision, calibration and details. It records existing computations without rerunning the test. Unsupported or undefined results retain their existing sentinel/NA representation and may lack this attribute. See rita_analysis for reporting conventions.

References

Jarque, C. M. and Bera, A. K. (1980). Efficient tests for normality, homoscedasticity and serial independence of regression residuals. Economics Letters, 6(3), pp. 255-259.

Shreve, Joni N. and Donna Dea Holland. 2018. SAS® Certification Prep Guide: Statistical Business Analysis Using SAS®9. Cary, NC: SAS Institute Inc.

Examples

```
set.seed(42)
x <- JBTest(qnorm(ppoints(40)), warn = FALSE)
attr(x, "monte_carlo")$p.value
```

KSLTest *Kolmogorov-Smirnov-Lilliefors Test*

Description

This function computes the Lilliefors variant of the one-sample Kolmogorov-Smirnov test.

Usage

```
KSLTest(data, alpha = 0.05, j = 1, warn = T,
  method = "molin-abdi", n_sim = 100000)
```

Arguments

data	The data of a univariate distribution for which the test statistic is computed (vector)
alpha	The two-sided decision threshold used for hypothesis-testing (scalar)
j	The # hypotheses tested; used to compute a Bonferonni correction, if applicable; should remain at its default if multiple testing is not an issue (scalar)
warn	Logical; warn by default that clipped polynomial output is an approximation-boundary result, not a literal tail probability of zero or one.
method	Exact calibration name: "molin-abdi" (default) or "monte-carlo".
n_sim	Positive integer simulation count, used only for "monte-carlo"; default 100000.

Details

The default method = "molin-abdi" uses the polynomial approximation of Molin and Abdi (1998).

The alternative method = "monte-carlo" replaces the polynomial approximation with simulation-based calibration. Each of n_sim independent standard-normal samples has the same usable sample size as the observed data. The mean and sample standard deviation are re-estimated for each simulated sample before calculating the Lilliefors statistic (D). The upper-tail p-value is calculated as (1 + count(D_sim >= D_obs)) / (n_sim + 1), counting simulated statistics equal to or greater than the observed statistic as exceedances. This procedure simulates the normal null distribution rather than resampling the observed data. Use set.seed() before calling the function for reproducibility; simulation advances R's random-number generator state.

At least four finite, nonconstant observations are required after omitting NA and NaN. Input must be a numeric vector. Centered and scaled calculations improve numerical stability, provided the relevant differences remain representable. Ties are accepted when calculating the statistic, but continuous-normal calibration is not established for rounded or discretized observations.

Monte Carlo p-values are returned without rounding to prevent small positive estimates from being displayed as zero. The statistic (D) retains three-decimal rounding, and significance decisions use the unrounded p-value with the strict threshold alpha/j. The minimum attainable Monte Carlo p-value is 1 / (n_sim + 1), reflecting simulation resolution rather than a lower bound on the true probability.

The monte_carlo attribute records the usable sample size, simulation count, exceedances, unrounded statistic, minimum attainable p-value, presence of ties, and a 95% exact Clopper-Pearson binomial confidence interval for the underlying null-tail probability conditional on the observed statistic. This interval quantifies simulation uncertainty, not robustness to model misspecification. Stringent significance thresholds may require substantially more simulations. The default Molin–Abdi method uses three-decimal rounding.

Value

An object including the test statistic, p-value, and a significance flag (list)

Precision metadata

Successful computed results also carry a rita_test attribute with the unrounded statistic and p.value, n_tested, precision, calibration and details. It records existing computations without rerunning the test. Unsupported or undefined results retain their existing sentinel/NA representation and may lack this attribute. See rita_analysis for reporting conventions.

References

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.

Molin, P., & Abdi, H. (1998). New Tables and numerical approximation for the KolmogorovSmirnov/Lillierfors/Van Soest test of normality.

Examples

```
values <- rnorm(100)
x <- KSLTest(data = values)
set.seed(2026)
mc <- KSLTest(values, method = "monte-carlo", n_sim = 999)
attr(mc, "monte_carlo") # Increase n_sim for precise tail inference.
```

---

kurtCoeff *Adjusted Fisher-Pearson Excess Sample Kurtosis*

Description

Adjusted Fisher-Pearson Excess Sample Kurtosis

Usage

```
kurtCoeff(data, sd)
```

Arguments

data	The data for which kurtosis is computed (vector)
sd	The supplied standard deviation (scalar)

Value

The kurtosis value (scalar)

References

Shreve, Joni N. and Donna Dea Holland . 2018. SAS® Certification Prep Guide: Statistical Business Analysis Using SAS®9. Cary, NC: SAS Institute Inc.

Examples

```
values <- rnorm(100)
x <- kurtCoeff(data = values, sd = sd(values))
```

`logitXform` *Logit/Log-Odds Transformation*

Description

This function transforms data via the logit/log-odds transformation.

Usage

```
logitXform(
  sample, divisor = 2,
  input = c("auto", "counts", "proportions", "percentages", "continuous")
)
```

Arguments

`sample`	The input data (vector, matrix, or dataframe)
`divisor`	Legacy argument, retained for compatibility but soon to be deprecated.
`input`	Input representation: auto (legacy scaling), counts, proportions, percentages, or continuous.

Details

logitXform() first uses input to put the data on a scale from 0 to 1. It then adjusts any values at 0 or 1 and applies the log-odds transformation with qlogis().

Choosing an input mode

"proportions" keeps values as supplied. Each value must be between 0 and 1, and the values can have any total.

"percentages" divides each value by 100. Each value must be between 0 and 100, and the values can have any total.

"counts" divides each nonnegative value by the sum of all values. The sum must be positive. Matrices and data frames use one total across all entries.

"continuous" shifts the values so that a negative minimum becomes zero. It then divides by the resulting maximum if that maximum exceeds 1.

"auto" is the default. If any value is negative, it uses the "continuous" rule. For nonnegative data, it divides by 100 when the values sum to 100 and leaves them as supplied when they sum to 1. Otherwise, it divides by the maximum if the maximum is at least 1. Choose an explicit mode when you know what the values represent, since the same automatic rules apply to counts and continuous scores.

Adjusting zeros and ones

After preprocessing, if any value equals 0 or 1, the function sets epsilon to half the smallest strictly positive value. It replaces each 0 with epsilon and each 1 with 1 - epsilon. Values already strictly between 0 and 1 are passed to qlogis() as they are. The divisor argument is accepted for compatibility and has no effect.

The endpoint adjustment can change the ordering of values or make distinct values equal. For example, c(0, 1) becomes c(0.5, 0.5) after the adjustment and returns c(0, 0) on the log-odds scale.

Input requirements

Supply at least one finite numeric value. NA, NaN, and infinite values are rejected. Numeric matrices, arrays, and data frames are flattened in column-major order, and the result is a vector. All-zero proportions raise an error because there is no positive value from which to calculate epsilon. The function also raises an error if preprocessing causes a positive proportion to round down to zero or if an adjusted endpoint cannot be represented strictly between 0 and 1. A constant proportion strictly between 0 and 1 is valid.

Rita() and MasterXform() use "auto". Convert known counts to proportions before passing them to either function. Rita() applies complete-case cleaning and variable-eligibility rules before transformation.

Value

The logit-transformed data (vector)

References

Stevens, S., Valderas, J. M., Doran, T., Perera, R., & Kontopantelis, E. (2016). Analysing indicators of performance, satisfaction, or safety using empirical logit transformation. bmj, 352.

Osborne, J. W. (2002). Notes on the use of data transformations. Practical Assessment, Research and Evaluation, 9(1), 42-50.

Osborne, J. W. (2002). The Effects of Minimum Values on Data Transformations. Retrieved from https://files.eric.ed.gov/fulltext/ED463313.pdf

Warton, D. I., & Hui, F. K. (2011). The arcsine is asinine: the analysis of proportions in ecology. Ecology, 92(1), 3-10.

emudrak (2014). Empirical logit transformation on percentage data. Cross Validated, answer 110037 to question 109702, July 30.

Examples

```
values <- rnorm(100)
x <- logitXform(values)
logitXform(c(10,20,30), input = "counts")
logitXform(c(0,0.1,0.99,1), input = "proportions")
```

`logXform` *Logarithmic Transformation*

Description

This function imputes minimum values per the recommendations of Osborne (2002) and subsequently transforms the data to a base-10 logarithmic scale.

Usage

```
logXform(sample)
```

Arguments

`sample`	Nonempty finite numeric vector, matrix, array, or data frame without missing values.

Details

Computes log10((sample - min(sample)) + 1), including when all values are already positive. Centering precedes the unit offset to retain the minimum-1 anchor at large locations. Opposite-sign extremes use equivalent scaled arithmetic to avoid intermediate overflow.

Input must be nonempty and finite, with no NA or NaN. Numeric vectors, matrices, arrays and data frames are accepted; dimensions and names are retained. Anchoring uses the minimum over the whole input. Constants return zero. Rita() performs its own complete-case cleaning before calling transformations.

Value

The log-transformed data, retaining input dimensions and names

References

Osborne, J. W. (2002). Notes on the use of data transformations. Practical Assessment, Research and Evaluation, 9(1), 42-50.

Osborne, J. W. (2002). The Effects of Minimum Values on Data Transformations. Retrieved from https://files.eric.ed.gov/fulltext/ED463313.pdf

Examples

```
values <- rnorm(100)
x <- logXform(values)
```

MasterTest *Master Normality Testing Function*

Description

This is a master function to call the appropriate test(s) to be used in the 'Rita' function.

Usage

```
MasterTest(c, data, alpha = 0.05, j = 1, ...)
```

Arguments

c	Numeric method code or descriptive name as documented for Rita(). MasterTest supports individual tests (1 through 6); numeric 7 retains its legacy NULL return. MasterXform also supports best (1).
data	The data of a univariate distribution for which the test statistic is computed (vector)
alpha	Significance level for the selected test (scalar). A usable p-value below alpha / j gives a rejection.
j	Number of hypotheses for the Bonferroni correction (scalar, default 1).
...	Additional arguments passed to the selected individual test, such as large_n/subsample_n for SWTest or method/n_sim for KSLTest.

Value

An results object specific to the test designated with the 'c' argument (list)

Examples

```
values <- rnorm(100)
x <- MasterTest(c = "shapiro-wilk", data = values)
set.seed(2026)
mc <- MasterTest("ksl", values, method = "monte-carlo", n_sim = 999)
```

MasterXform *Master Transformation Function*

Description

This is a master function used to perform the appropriate transformation(s) within the 'Rita' function.

Usage

```
MasterXform(c, data)
```

Arguments

c	Numeric method code or descriptive name as documented for Rita(). MasterTest supports individual tests (1 through 6); numeric 7 retains its legacy NULL return. MasterXform also supports best (1).
data	The data of a univariate distribution for which the test statistic is computed (vector)

Value

Output from the appropriate subfunction (list)

Examples

```
values <- rnorm(100)
x <- MasterXform(c = "log", data = values)
```

---

popSD *Converts Sample Standard Deviations into Population Equivalents*

Description

This function converts a sample standard deviation (SD) input into the population equivalent. This code is vectorized to convert several sample standard deviations for univariate distributions of identical sample-sizes, if desired.

Usage

```
popSD(s, n)
```

Arguments

s	The sample SD(s) (vector)
n	The sample-size for each SD to be converted (vector)

Value

The population SD(s) (vector)

References

Ruscio, J. (2021). Fundamentals of research design and statistical analysis. Ewing, NJ: The College of New Jersey, Psychology Department.

Examples

```
values <- rnorm(100)
x <- popSD(s = sd(values),n = 100)
```

`rankitXform` *Rankit Transformation*

Description

This function transforms data via the Rankit, a member of the families of 'rank-based normalization methods' and 'empirical normal quantile transformations' employed in both the social sciences and quantitative genetics.

Usage

```
rankitXform(sample)
```

Arguments

`sample`	The input data (vector)

Details

Uses qnorm((rank(sample) - 0.5) / length(sample)) with R's default average ranks for ties. The original order is retained. The existing rank() defaults also rank NA and NaN last, in their order of occurrence, rather than omitting them. Infinite values are ranked as extreme observations. Rita() applies its own complete-case cleaning before this function is called.

Value

The Rankit-transformed data (vector)

References

Soloman, S. R., & Sawilowsky, S. S. (2009). Impact of rank-based normalizing transformations on the accuracy of test scores. Journal of Modern Applied Statistical Methods, 8(2), 9.

Peng, B., Robert, K. Y., DeHoff, K. L., & Amos, C. I. (2007, December). Normalizing a large number of quantitative traits using empirical normal quantile transformation. In BMC proceedings (Vol. 1, No. 1, p. S156). BioMed Central. doi: 10.1186/1753-6561-1-s1-s156

Bliss, C. I., Greenwood, M. L., & White, E. S. (1956). A rankit analysis of paired comparisons for measuring the effect of sprays on flavor. Biometrics, 12(4), 381-403.

Examples

```
values <- rnorm(100)
x <- rankitXform(values)
```

Rita *Rita*

Description

Rita is the phonetic interpretation of how the acronym 'REDA' (R Exploratory Data Analysis) may sound. This namesake function summarizes an input dataset by the arithmetic mean, sample standard deviation, '5 number summary' (minimum, maximum, median, and quartiles) and adjusted Fisher-Pearson skewness and excess kurtosis. Rita then normalizes the dataset with several possible transformations and provides diagnostic plots of the normalizing transformation with the best-fitting squared correlation with theoretical normal quantiles. One of many possible normality tests are then performed to provide context to the best-fitting method. Additional visualizations are output according to an algorithm or as specified by the user.

Usage

```
Rita(data, test = 1, xform = 1, alpha = 0.05, j = 1, autoPlot = T,
  histPlot = F, densPlot = F, stripPlot = F, violinPlot = F,
  xformPlot = F, return = T, seed = NULL, sw_large_n = "subsample",
  sw_subsample_n = 5000, ksl_method = "molin-abdi", ksl_n_sim = 100000,
  plot_profile = "compact", plots_per_page = 3, display_plots = TRUE)
```

Arguments

`data` Input dataset (matrix, dataframe, or vector). For a univariate distribution, submit a vector or a subsetted matrix or dataframe. If results for many univariate distributions are desired, submit a matrix or dataframe with each column representing a given variable if all distributions are of the same sample-size. If not, it is recommended to call Rita repeatedly for each variable.

`test` Desired normality test (numeric code or descriptive name). Names: shapiro-wilk (shapiro, sw), kolmogorov-smirnov-lilliefors (ksl, lilliefors), anderson-darling (ad), jarque-bera (jb), dagostino-pearson (dp), chi-square (chisq), or all. Matching is case-insensitive and exact; surrounding whitespace is ignored. By default (test = 'sw' / test = 1), Rita will present the results of the Shapiro-wilk test to the user.

- **Shapiro-Wilk (default)**
  `test` = 1, `"shapiro-wilk"`, `"shapiro"`, or `"sw"`
- **Kolmogorov-Smirnov with Lilliefors correction**
  `test` = 2, `"kolmogorov-smirnov-lilliefors"`, `"ksl"`, `"lilliefors"`, or `"kolmogorov-smirnov"`
- **Anderson-Darling**
  `test` = 3, `"anderson-darling"`, or `"ad"`
- **Jarque-Bera**
  `test` = 4, `"jarque-bera"`, or `"jb"`
- **D'Agostino-Pearson omnibus**
  `test` = 5, `"dagostino-pearson"`, `"d'agostino-pearson"`, `"dagostino"`, or `"dp"`

- **Pearson chi-square**
  `test` = 6, `"chi-square"`, `"chisq"`, `"chisqtest"`, or `"chi-squared"`
- **All six tests for the best-performing transformation**
  `test` = 7, `"all"`, or `"all-tests"`

The order of the tests printed corresponds to the order of the variables stored within the input dataset.

`xform` Desired normalization method (numeric code or descriptive name). Names: best, log (log10, logarithmic), inverse (reciprocal), square-root (sqrt), arcsine (arc-sine), logit (log-odds), or rankit (enqt). Matching is case-insensitive and exact; surrounding whitespace is ignored. By default (xform = 1), Rita will assess which method performs best and (a.) return the transformed data to the user, and (b.) visualize the data according to the settings of the plotting arguments.

Please note that, per the recommendations of Osborne (2002), a constant is added prior to the logarithmic transformation and after reflection for the inverse transformation to ensure that the minimum value is anchored at 1, and prior to the square-root transformation to ensure a left anchor of 0.

Similarly, the arc-sine and logit transformations are applied after converting the units, if needed, to ensure that variables are bounded between 0 and 1.

The "best performing" method is identified by comparing goodness-of-fit to the straight line of the QQ plot, measured as squared Pearson correlation, for the quantiles of the data normalized by a given method and the standard normal distribution. If a tie is present between transformations for a variable, the first candidate in the existing transformation order is selected.

- **Automatic selection (default)**
  `xform` = 1, `"best"`, or `"best-performing"` For each variable, Rita compares methods 2 through 6 and selects the one with the highest squared correlation with theoretical normal quantiles. Rankit is excluded from automatic selection.
- **Base-10 logarithmic transformation**
  `xform` = 2, `"log"`, `"logarithmic"`, or `"log10"`
- **Inverse (reciprocal) transformation**
  `xform` = 3, `"inverse"`, or `"reciprocal"`
- **Square-root transformation**
  `xform` = 4, `"square-root"`, or `"sqrt"`
- **Arcsine square-root transformation**
  `xform` = 5, `"arcsine"`, `"arc-sine"`, or `"arcsin"`
- **Logit (log-odds) transformation**
  `xform` = 6, `"logit"`, or `"log-odds"`
- **Rankit (empirical normal quantile transformation)**
  `xform` = 7, `"rankit"`, or `"enqt"`

`alpha` Significance level for the normality test (scalar). A usable p-value below alpha / j gives a rejection.

`j` Number of hypotheses for the Bonferroni correction of the normality test(scalar, default 1).

`autoPlot` Logical. If `TRUE` (the default), `plot_profile` determines which plots Rita generates, and the five manual plotting flags are ignored. Choose one of the following profiles.

`"compact"` **(default)** Pairs a histogram of the original data for integer-valued variables, or a density plot otherwise, with a density plot of the transformed data.

`"full"` Uses the same types of distribution plots in the historical grouping, with integer-valued variables first. It also adds one plot of the original data per variable: a strip plot if there are at most 20 distinct values and at most 150 retained observations, or a combined violin and box plot otherwise.

`"qq"` Pairs normal Q-Q plots of the original and transformed data.

`plots_per_page` sets the number of variable pairs per page for `"compact"` and `"qq"` (default 3).

If `FALSE`, `histPlot`, `densPlot`, `stripPlot`, `violinPlot`, and `xformPlot` determine which plots Rita generates. The `plot_profile` and `plots_per_page` arguments are ignored. With all five flags at their default of `FALSE`, Rita generates no plots.

In either mode, `display_plots = FALSE` suppresses automatic drawing while still returning any plots that were generated.

`histPlot` Whether to generate histograms for each variable (boolean).

`densPlot` Whether to generate density plots for each variable (boolean).

`stripPlot` Whether to draw strip plots for each variable (boolean).

`violinPlot` Whether to draw violin plots for each variable (boolean).

`xformPlot` Whether to draw density plots for each transformed variable (boolean).

`return` Whether to return the transformed variables of the best performing method (return = T; default), or the cleaned, untransformed variables eligible for transformation (return = F) (boolean).

`seed` Optional explicit seed (scalar). Default NULL uses the caller's RNG stream. With an explicit seed, Rita restores the prior RNG kind and `.Random.seed` state on exit. Cached state outside `.Random.seed`, including the cached draw of the Box-Muller normal generator, can change.

`sw_large_n` Shapiro-Wilk policy above 5000 usable rows: "subsample" (default) or "error". All-tests mode reports an unavailable SW result under "error" and continues the other tests. SW options are ignored unless SW or all tests are requested.

`sw_subsample_n` Integer from 3 to 5000 for SW subsampling (default 5000). One row-index set is reused across selected transformations.

`ksl_method` KSL calibration: "molin-abdi" (default) or explicit "monte-carlo". No automatic fallback is performed.

`ksl_n_sim` Positive integer Monte Carlo simulation count; default 100000.

`plot_profile` Automatic profile: "compact" (default), "full", or "qq". Exact, case-sensitive names. See `rita_plotting` for layouts and compatibility.

`plots_per_page` Positive integer giving variables per compact/QQ page (default 3). Ignored for full or manual plotting. Each variable has two panels with separate scales.

`display_plots` Logical (default TRUE). FALSE suppresses automatic drawing and retains the plot objects and console statistics. In batch scripts, assign the result to avoid R's automatic printing of the returned plots.

Details

Any rows with missing values (NAs) are removed to form complete cases for calculation purposes; if desired, incomplete records should be imputed or removed with subsetting prior to calling Rita. In addition, note that any columns not numeric type or coercible to numeric are excluded from analysis, as are any numeric columns with 2 distinct values or less. Columns are coerced independently. Numeric values are preserved, and factors are interpreted by their labels. Complex-valued columns are rejected.

Descriptive skewness and excess kurtosis use adjusted Fisher-Pearson coefficients with sample SD. Cleaning reports original missingness, values made missing by numeric coercion, excluded variables and reasons, omitted rows, and retained dimensions. Users should select and prepare their analytical variables before calling Rita; no imputation or variable-specific complete-case samples are introduced.

Normality tests evaluate the selected best-fitting transformation, not every candidate. SW subsampling uses the caller's RNG by default and the same retained row indices for all tested variables; identical starting RNG states/usable row order reproduce the subset across calls. Returned sw_sampling indices refer to original rows, with complete_case_indices identifying positions after rows with NAs in any cells are omited. All-tests, sub-sampling, and explicit simulation calls also return a normality_tests attribute containing test results and any sampling/calibration metadata.

Value

An unnamed two-element list containing the selected transformed dataset for each variable (or cleaned data when return = FALSE) and the specified plots. The additive rita_plotting attribute retains plot-ready observations; see `rita_plotting`. Compact/QQ return a list of trellis pages; full and manual modes retain their historical nested collections.

Structured result retention

Every successful call also has a rita_analysis attribute containing settings, cleaning records, unrounded original-scale descriptives, transformation selection information and all requested test results. See rita_analysis for its schema.

Examples

```
values <- rnorm(100)
x <- Rita(data = values, test = "shapiro-wilk", xform = "rankit")
analysis <- attr(x, "rita_analysis")
analysis$cleaning$retained_rows
analysis$tests[[1]][["shapiro-wilk"]]$p.value
# Generate compact pages without displaying; retrieve and print deliberately.
fit <- Rita(mtcars, display_plots=FALSE)
print(fit[[2]][[1]])
# Reuse saved observations; no tests, transformations or selection are rerun.
full <- RitaPlots(fit, "full")
qq <- RitaPlots(fit, "qq", plots_per_page=2)
```

```
# Historical manual flags; no automatic profile or display.
manual <- Rita(mtcars, autoPlot=FALSE, histPlot=TRUE, display_plots=FALSE)
# No generated plots, but plotting data remain available.
no_plots <- Rita(mtcars, autoPlot=FALSE)
attr(no_plots, "rita_plotting")$data[[1]]$original_rows
```

`RitaPlots` *Regenerate Rita plots from retained observations*

Description

Builds lattice plots without rerunning cleaning, transformations, selection or normality tests. The input must be a Rita result with rita_plotting metadata.

Usage

```
RitaPlots(result, plot_profile = "compact", plots_per_page = 3,
  display_plots = FALSE)
```

Arguments

`result`	A result from Rita().
`plot_profile`	One of "compact", "full", or "qq".
`plots_per_page`	Positive integer; variables per compact or QQ page.
`display_plots`	Whether to print generated plots. Default FALSE.

Value

A list of lattice trellis pages (compact/QQ), or the historical nested plot collection (full). See `rita_plotting` for metadata and RNG details.

Examples

```
fit <- Rita(mtcars, xform="rankit", autoPlot=FALSE)
pages <- RitaPlots(fit, "qq")
print(pages[[1]])
```

`rita_analysis` *Retained Rita Analysis Metadata*

Description

Structured records retained by Rita and precision metadata supplied by its test wrappers.

Details

The additive rita_analysis attribute is present on every successful Rita() call. Its schema_version is integer 1. All indices are one-based. No S3 class is added. The first two positional return elements keep their existing meanings.

settings contains canonical test and transformation names, alpha, j, threshold, seed, return_transformed, plotting flags, and active SW/KSL options. Inactive SW or KSL settings are NULL. A NULL seed means the caller's RNG stream was used; no copy of the caller's full RNG state is retained.

cleaning contains input_rows, input_columns, final_rows, final_columns, retained_columns (original positions), retained_rows and omitted_rows (original row positions). Its variables data frame has one row per original input column: original_position, variable, missing_original, missing_coercion, missing_total, considered_for_complete_cases, exclusion_stage, exclusion_reason and retained. Missingness fields are counts, not observation-level masks. Exclusion stages are before-complete-cases or after-complete-cases, with NA for retained variables. The considered_for_complete_cases flag separates early-excluded missingness from missingness relevant to the actual row filter. Counts include NA and NaN. Metadata records the executed cleaning sequence. Later exclusions never recover rows and do not alter the common complete-case sample.

descriptives contains scale (original-cleaned), values (an unrounded data frame with statistics in rows and retained variables in columns), skewness and kurtosis convention labels. Undefined small-sample moments remain undefined.

selection is a list in retained-variable order. Each record contains original_position, variable, method (canonical transformation), mode (requested or automatic), criterion, score, candidates, tied_methods and tie_resolution. Automatic candidates is an ordered data frame of method and score. All exact maxima are listed in tied_methods; the first is selected. tie_resolution is unique-maximum or first-exact-maximum. Requested transformations have NULL candidates, empty tied_methods and not-applicable tie_resolution. Their score is still the already-computed QQ correlation squared. Scores use the full cleaned data and the existing candidate order and formula.

tests is a list in retained-variable order; each element is a named list of the requested canonical tests, with the same structure in single- and all-tests mode. Each test record contains original_position, variable, transformation, test, statistic_name, statistic, p.value, precision, usable_n, tested_n, alpha, j, threshold, significant, decision, status, reason, calibration, calibration_details, monte_carlo, sampling, warnings, limitations and result. The result field is the actual wrapper result, including its attributes, also used by the pre-existing normality_tests attribute where that attribute applies. Descriptive list names may repeat if input names repeat; original_position is the unambiguous identity.

statistic and p.value use the computed unrounded values supplied by rita_test. precision is unrounded-computation for usable numerical results and unavailable otherwise. decision is reject, do not reject, or unavailable. significant is NA when inference is unavailable. The raw legacy flag is in result[[3]] when present. Usable n describes the common cleaned sample. Tested n records the actual sample

used for attempted computations, including failures and SW subsets. It is zero when the test did not run. No test is rerun to construct these fields.

status is computed, monte-carlo-floor (zero exceedances), approximation-boundary (clipped Molin-Abdi output), numerical-zero, invalid-numerical-result, or unavailable. A numerical zero may reflect underflow. An attempted numerical failure has status invalid-numerical-result and an informative reason. The original statistic and p.value remain available for diagnosis. The main console table masks invalid values and shows the decision as unavailable. Failure to reject does not establish normality. Nominal probabilities do not account for fitting or selecting a transformation on the same data.

calibration identifies stats-shapiro-wilk, molin-abdi, normal-null-monte-carlo, stephens-1986, chi-square-2, pearson-chi-square-approximation, or not-computed. calibration_details includes the raw Molin-Abdi probability and clipped flag, chi-square df where applicable, and Pearson cell count and expected frequency. monte_carlo retains the wrapper's existing replicate counts, exceedances, resolution and any available interval or other metadata; it is NULL for non-simulation methods. The plus-one resolution floor is not a bound on the true tail probability. sampling retains SW metadata with indices mapped to original rows, original_n describing the original input size, and complete_case_indices mapped to the cleaned sample. The nested result keeps the wrapper's original coordinate convention unchanged.

warnings records warning messages emitted during that individual test call without suppressing them. Global Rita warnings, such as shared subsampling, are not duplicated here. limitations separately records applicable qualifications, including subsampling, approximate calibration and small-sample restrictions. Existing method-specific numerical and calibration limitations remain.

The wrapper attribute rita_test contains statistic, p.value, n_tested, precision, calibration, details, status, reason and decision. The metadata decision is unavailable when the raw result is invalid. The legacy list elements keep their original rounding and flags. Small-sample sentinels may lack this attribute. A numerical zero or one alone does not establish an exact tail probability or approximation clipping.

See Also

Rita

`rita_plotting` *Rita plotting profiles and retained plotting data*

Description

Rita retains the original and selected transformed observations in an additive `rita_plotting` attribute. This is separate from the unchanged `rita_analysis` schema. Both positional return elements retain their meanings: data, followed by lattice plotting objects. No analysis-result class is added.

Details

With `autoPlot=TRUE`, `plot_profile="compact"` is the default. Each variable has an original histogram (using Rita's existing integer-valued feature detection) or density on the left and its selected transformed density on the right. Variables follow retained input-column order. Three variable pairs per page is the default; `plots_per_page` changes this. Both axes have panel-specific ranges; original and transformed units differ. Histograms show percent per bin; densities show density in the corresponding measurement units. No strip or violin plots are included in compact mode.

`plot_profile="full"` retains the historical repertoire and grouping: paired original/transformed main plots (integer-valued originals first), followed by original strip plots for at most 20 distinct values and 150 observations, otherwise violin/boxplots. It returns the historical two-group nested list, without pagination. Nine variables therefore yield 27 trellis objects in full mode versus three six-panel trellis pages in compact mode.

`plot_profile="qq"` uses the same pagination and variable/stage order as compact. Sorted observations are plotted against `qnorm(ppoints(n))`, with standard-normal theoretical quantiles. The dashed reference line joins the observed 25th and 75th percentiles (R quantile type 7) at the corresponding standard-normal quantiles. It is not a fitted least-squares line or a confidence envelope. The transformed panel labels the already-retained QQ-correlation-squared score. Better visual alignment does not establish normality or validate inference after selecting a transformation on the same observations.

Precedence is explicit: with `autoPlot=TRUE`, the selected profile controls plots and all five manual flags are ignored. With `autoPlot=FALSE`, the profile and pagination arguments are ignored, and only `histPlot`, `densPlot`, `stripPlot`, `violinPlot`, and `xformPlot` apply. The five historical groups (including empty groups) remain in position 2. With all flags FALSE, no plots are generated, but plotting data are still retained. `display_plots=FALSE` suppresses automatic drawing and keeps the console statistics and plot objects. For batch analysis, assign the result, for example `fit <- Rita(mtcars, display_plots=FALSE)`. Printing the plot collection explicitly or through R's auto-printing can open the default graphics device.

`RitaPlots(result, plot_profile, plots_per_page, display_plots=FALSE)` builds another profile using saved observations, even after a no-plot call or a call with `return=FALSE`. It never reruns transformations, selection, normality tests or sampling. Density estimation and plotting coordinates are prepared as needed. Print an individual returned trellis object to draw or export it deliberately. For three pairs per page, a device approximately 10 by 10 inches is recommended. A 5-by-5-inch device uses fewer ticks and fits labels to the width of each strip. Long names may show their input column number and an abbreviated name. Use fewer pairs on shorter devices. Very narrow strips may abbreviate the transformation or show only the QQ score when space permits. Full labels and scores remain in `rita_plotting`.

Generation alone does not draw random numbers. Existing lattice rendering of jittered density rugs and strip plots may advance the caller's RNG, including on reprinting. Jitter is not redesigned here. Compact omits supplemental strips and changes plotting order, so its rendering-time RNG consumption can differ from full. QQ rendering consumes no RNG. Rita's explicit seed remains scoped to the Rita call; later printing or RitaPlots display is outside that scope. Plots use all cleaned observations, not just a Shapiro-Wilk random subsample.

Schema version 1

`attr(result,"rita_plotting")` is a list containing:

schema_version Integer 1.

profile Active profile: compact, full, qq or manual.

plots_per_page Configured variables per page for compact/QQ; NULL otherwise.

display Whether automatic drawing was requested.

data One record per retained variable in input order.

Each data record contains `original_position` and `variable`, matching `rita_analysis$selection`; `original_rows`, matching `rita_analysis$cleaning$retained_rows`; numeric `original` and `transformed` vectors in that row order; canonical `transformation`; `transformed_label`; and the existing `selection_score`. `stages` is original then transformed. `original_plot`, `transformed_plot`, and `supplemental_plot` describe the existing feature-detection choices. `qq` documents probabilities, theoretical distribution and reference-line convention. These observations and conventions suffice to regenerate QQ coordinates deterministically. Original column position, not a possibly duplicated variable name, identifies a record.

Each compact/QQ trellis page also has a `rita_panels` attribute, ordered left to right, top to bottom. Records retain original position, variable, stage, transformation, type, original rows and observations. QQ panels additionally retain a `coordinates` data frame (original_row, theoretical, observed) and `reference` (intercept, slope). Tied observations remain individually identified in sorted order. Non-QQ coordinates/reference are NULL. Actual lattice panel arguments are constructed from these same observations.

Examples

```
fit <- Rita(mtcars, display_plots=FALSE)
length(fit[[2]]) # three compact pages
attr(fit,"rita_plotting")$data[[1]]$original
print(fit[[2]][[1]])
full <- RitaPlots(fit,"full")
qq <- RitaPlots(fit,"qq",plots_per_page=2)
attr(qq[[1]],"rita_panels")[[1]]$coordinates
```

---

skewCoeff — *Adjusted Fisher-Pearson Skewness Coefficient with Sample-size Correction Factor*

Description

Adjusted Fisher-Pearson Skewness Coefficient with Sample-size Correction Factor

Usage

```
skewCoeff(data, sd)
```

Arguments

`data`	The data for which skewness is computed (vector)
`sd`	The supplied standard deviation; sample SD gives the adjusted Fisher-Pearson coefficient (scalar)

Value

The skewness value (scalar)

References

Shreve, Joni N. and Donna Dea Holland . 2018. SAS® Certification Prep Guide: Statistical Business Analysis Using SAS®9. Cary, NC: SAS Institute Inc.

Examples

```
values <- rnorm(100)
x <- skewCoeff(data = values,sd = sd(values))
```

---

squareXform *Square-root Transformation*

Description

This function left anchors the minimum value to 0 per the recommendations of Osborne (2002) and subsequently transforms the data by taking the square-root of each value.

Usage

```
squareXform(sample)
```

Arguments

sample	Nonempty finite numeric vector, matrix, array, or data frame without missing values.

Details

Computes sqrt(sample - min(sample)), including for positive input.

Input must be nonempty and finite, with no NA or NaN. Numeric vectors, matrices, arrays and data frames are accepted; dimensions and names are retained. The minimum is taken over the whole input. Constants return zero.

Value

The square-root-transformed data, retaining input dimensions and names

References

Osborne, J. W. (2002). Notes on the use of data transformations. Practical Assessment, Research and Evaluation, 9(1), 42-50.

Osborne, J. W. (2002). The Effects of Minimum Values on Data Transformations. Retrieved from https://files.eric.ed.gov/fulltext/ED463313.pdf

Examples

```
values <- rnorm(100)
x <- squareXform(values)
```

`SWTest` *Shapiro-Wilk Test*

Description

This function is a wrapper for shapiro.test() from the stats package. Options added include an ability to toggle a Bonferonni correction for significance, a corresponding significance flag, and reorganized output to facilitate integration with the Rita package.

Usage

```
SWTest(data, alpha = 0.05, j = 1, warn = T,
  large_n = "subsample", subsample_n = 5000)
```

Arguments

`data`	Data of a univariate distribution for which the test statistic is computed (vector)
`alpha`	The two-sided decision threshold used for hypothesis-testing
`j`	The # hypotheses tested; used to compute a Bonferonni correction, if applicable; should remain at its default if multiple testing is not an issue (scalar)
`warn`	Logical; emit a sample-size diagnostic for random subsampling (enabled by default) or termination below three usable observations.
`large_n`	Exact policy for more than 5000 usable observations: "subsample" (default) or "error".
`subsample_n`	Integer from 3 to 5000; subset size when large_n="subsample" and usable n exceeds 5000. Default 5000.

Details

With 3 to 5000 usable observations, the ordinary stats::shapiro.test is performed. When 'data' has N > 5000 and subsample_n = 5000 (default), a sample of size N = 5000 is constructed by taking a random subsample without replacement from 'data' to approximate a Shapiro-Wilk p-value. Select large_n = "error" to terminate instead of sampling. With fewer than three usable observations, testing is terminated without computed results. NA/NaN are omitted before checking sample size.

Subsampling advances R's RNG without resetting it; use set.seed() before calling. Existing ties in the data are not removed. The sampling attribute records original, usable and analyzed sizes, original-input indices, and a random-subsample method label, even when warn=FALSE. Rita's sw_large_n option shares one set across selected transformations in a call. Significance uses the unrounded p-value against alpha/j; displayed values retain three-decimal rounding.

Value

An object including the test statistic, p-value, and a significance flag (list)

Precision metadata

Successful computed results also carry a rita_test attribute with the unrounded statistic and p.value, n_tested, precision, calibration and details. It records existing computations without rerunning the test. Unsupported or undefined results retain their existing sentinel/NA representation and may lack this attribute. See rita_analysis for reporting conventions.

References

Patrick Royston (1982). An extension of Shapiro and Wilk's W test for normality to large samples. Applied Statistics, 31, 115–124. 10.2307/2347973

Patrick Royston (1982). Algorithm AS 181: The W test for Normality. Applied Statistics, 31, 176–180. 10.2307/2347986

Patrick Royston (1995). Remark AS R94: A remark on Algorithm AS 181: The W test for normality. Applied Statistics, 44, 547–551. 10.2307/2986146

Examples

```
values <- rnorm(100)
x <- SWTest(data = values)
set.seed(2026)
large_values <- rnorm(6000)
sampled <- SWTest(large_values, warn = FALSE)
attr(sampled, "sampling")[c("original_n", "analyzed_n")]
```

Index

ADTest, 2
arcsineXform, 3

chisqTest, 4

DPTest, 6

inverseXform, 7

JBTest, 8

KSLTest, 9
kurtCoeff, 11

logitXform, 12
logXform, 14

MasterTest, 15
MasterXform, 15

popSD, 16

rankitXform, 17
Rita, 18, *24*
rita_analysis, 23
rita_plotting, *20–22*, 24
RitaPlots, 22

skewCoeff, 26
squareXform, 27
SWTest, 28