

Package: arkhaia (via r-universe)

October 21, 2026

Title Archaeological and Historical Analysis

Version 0.6.12

Date 2026-04-11

Description Tools for quantitative analysis related to archaeological and historical problems for irregularly spaced time indexed observations, toward evaluating linear dependence and homogeneity over time. Methods include effect sizes for measuring homogeneity, simulation from a truncated Poisson distribution for random right-censoring of count data, and least-squares spectral analysis by lowest frequency iteration for model fitting. Collins-Elliott (2026)
<https://volweb.utk.edu/~scolli46/sce_aqysuppl2026.pdf>.

License GPL (>= 3)

Imports Rcpp (>= 1.1.0), Rdpack, stats, utils

RdMacros Rdpack

Encoding UTF-8

LinkingTo Rcpp, RcppArmadillo

RoxygenNote 7.3.2

Suggests knitr, rmarkdown, testthat (>= 3.0.0)

Config/testthat/edition 3

VignetteBuilder knitr

Repository <https://scollinselliott.r-universe.dev>

Date/Publication 2026-10-21 15:38:57 UTC

RemoteUrl <https://github.com/scollinselliott/arkhaia>

RemoteRef HEAD

RemoteSha ada03289699d8f911be94dea282bde5eadedb0c5

Contents

CR	2
dtem	4
homogeneity	5
homogeneity_LOO	7
idt_ks	9
idt_wass	10
ks_arkhaia	12
log_OR_pair	12
LSSA	13
LSSA_LFI	15
LSSA_LFI_candidates	16
LSSA_LFI_comp	17
LSSA_LFI_epoch	17
LSSA_LFI_model	19
LSSA_LFI_multi	19
LSSA_LFI_pairwise	20
LSSA_LFI_validated	21
pa_matrix	21
pois_rcens	22
repartition	23
th_idt_ks	24
th_idt_wass	25
time_series	27
trace_pair	29
trim_epoch	30
trunc_pois	31
VB	32
VB_LOO_type	33
VB_pair	34
VB_trunc_pois	35
wass_R1	37
Index	39

CR

Cressie-Read Power-Divergence Statistic

Description

For a matrix of cross-tabulated counts of observations, computes the Cressie-Read power-divergence statistic according to the selection of a parameter λ (Cressie and Read 1984; Read and Cressie 1988).

Usage

```
CR(x, lambda = 2/3)

## S3 method for class 'matrix'
CR(x, lambda = 2/3)

## S3 method for class 'data.frame'
CR(x, lambda = 2/3)

## S3 method for class 'xtabs'
CR(x, lambda = 2/3)

## S3 method for class 'table'
CR(x, lambda = 2/3)
```

Arguments

x	A matrix of cross-tabulated counts.
lambda	The parameter of the Cressie-Read power-divergence statistic. Default is the recommended value of 2/3. To use Pearson's method, set lambda = 1.

Value

The Cressie-Read power-divergence statistic.

References

Cressie NAC, Read TRC (1984). "Multinomial Goodness-of-Fit Tests." *Journal of the Royal Statistical Society. Series B (Methodological)*, **46**, 440–464. doi:10.1111/j.25176161.1984.tb01318.x.

Read TRC, Cressie NAC (1988). *Goodness-of-Fit Statistics for Discrete Multivariate Data*. Springer, New York.

Examples

```
x1 <- c(2, 0, 10, 11, 5)
x2 <- c(1, 1, 17, 23, 3)
x3 <- c(0, 0, 2, 81, 11)

x <- matrix(c(x1, x2, x3), ncol = 3)

CR(x)
CR(x, lambda = 1)
```

Description

For multiple time-indexed observations and a number of events, $r = 1, \dots, n_r$, that occur at a given time index, this function performs a regression of the form $x(t) = \beta_0 + \sum_{r=1}^{n_r} \gamma t I(t \leq t_r) e^{-\lambda t}$, where $I(t \geq t_r)$ is an indicator function equaling 1 if $t \geq t_r$. Estimation of γ , λ , and β_0 is performed numerically on specified ranges of potential values. The intercept term is not used where cases where a zero intercept is expected (e.g., standardized observations).

Usage

```
dtem(
  x,
  events,
  gamma = c(-10, 10),
  lambda = c(0, 5),
  intercept = FALSE,
  incr = 0.01
)

## S3 method for class 'time_series'
dtem(
  x,
  events,
  gamma = c(-10, 10),
  lambda = c(0, 5),
  intercept = FALSE,
  incr = 0.01
)
```

Arguments

x	A data frame of two columns, with the first column containing time indices and the second containing values.
events	A vector of the time indices of the events, $t_1, \dots, t_{n_r}$.
gamma	The range of values of γ to evaluate. Default is <code>c(-10, 10)</code> .
lambda	The range of values of λ to evaluate. Default is <code>c(0, 5)</code> .
intercept	The range of values of the intercept β_0 to evaluate. If <code>FALSE</code> is given instead (the default) no intercept term is included.
incr	Increment by which to sample values of gamma and lambda. Default is <code>0.01</code> .

Value

A list containing

- A vector of coefficients and the residual sum of squares.
- A matrix of the residual sum of squares for samples within the ranges of gamma and lambda.

References

There are no references for Rd macro \insertAllCites on this help page.

Examples

```
# generate example of positive trend (gamma = 2)
# after event with slow decay (lambda = 0.05)
t_ <- runif(100, 0, 100)
events <- c(0, 54, 89)
gam <- 2
lam <- 0.05
x_ev1 <- (t_ - events[1]) * gam * exp(- lam * (t_ - events[1]))
x_ev2 <- (t_ - events[2]) * gam * exp(- lam * (t_ - events[2]))
x_ev3 <- (t_ - events[3]) * gam * exp(- lam * (t_ - events[3]))
x_ev1[t_ < events[1]] <- 0
x_ev2[t_ < events[2]] <- 0
x_ev3[t_ < events[3]] <- 0
y <- x_ev1 + x_ev2 + x_ev3 + rnorm(length(t_), 0, 0.25)
dat <- time_series(t = t_, x = y)

# fit model
res <- dtem(dat, events, gamma = c(-2.5,2.5), lam = c(0,0.1))

# coefficients
res$coefs
```

Description

Given a contingency table of cross-tabulated counts, with contexts along the columns and types along rows, this function estimates the distribution of effect sizes between pairs of "related" assemblages (determined *a priori*), as compared against a distribution of "unrelated" assemblages (if not specified, are supplied as all pairs which are not included in the "related" set). Hence, the practical significance of the level of homogeneity between related deposits is evaluated against that of unrelated deposits.

Usage

```
homogeneity(x, related = NULL, unrelated = NULL, direction = "UW")

## S3 method for class 'effect_sizes'
homogeneity(x, related = NULL, unrelated = NULL, direction = "UW")
```

Arguments

<code>x</code>	An <code>effect_sizes</code> object as returned by <code>VB_pair</code> .
<code>related</code>	The related pairs of contexts as a two-column matrix or data frame (contexts between which one anticipates a meaningful relationship). Names must match <code>colnames(x)</code> .
<code>unrelated</code>	The unrelated pairs of contexts as a two-column matrix or data frame (contexts between which one does not anticipate a meaningful relationship). Names must match <code>colnames(x)</code> . May be left <code>NULL</code> , in which event all pairs not expressed in <code>related</code> are created
<code>direction</code>	Whether the related or unrelated effect size should come first. Default is "UW"; alternative is "WU".

Value

A list of results:

11

- `n` - A vector of the number of related and unrelated pairs of contexts, respectively n_W and n_U .
- `U` - The effect sizes between unrelated pairs of contexts.
- `W` - The effect sizes between related pairs of contexts.
- `Q` - The quantile indicating the proportion of related contexts more homogenous than unrelated contexts (if `direction` is "UW"); less homogenous if `direction` is set to "WU".
- `D` - The distribution of differences, $D_{ij} = U_j - W_i$, if `direction` is set to "UW". The proportion of $D > 0$ is equivalent to the mean of `Q`.

Examples

```
x1 <- c(2, 0, 10, 11, 5)
x2 <- c(1, 1, 17, 23, 3)
x3 <- c(0, 0, 2, 81, 11)
x4 <- c(3, 18, 9, 0, 23)
x <- matrix(c(x1, x2, x3, x4), ncol = 4)
colnames(x) <- c("surface1", "subsurface1", "surface2", "subsurface2")
rownames(x) <- LETTERS[1:5]

# related pairs
W_contexts <- matrix(c("surface1", "surface2", "subsurface1", "subsurface2"), ncol = 2)

# unrelated pairs (will be automatically created if left NULL)
U_contexts <- matrix(c("surface1", "surface1", "surface2", "subsurface1",
                      "surface2", "subsurface2", "surface1", "subsurface2"), ncol = 2)
```

```
effect_sizes <- VB_pair(x)
homogeneity(effect_sizes, related = W_contexts, unrelated = U_contexts)
homogeneity(effect_sizes, related = W_contexts)
```

homogeneity_LOO*Leave-One-Out Routine for Homogeneity of Related Assemblages*

Description

To evaluate the stability of estimates of effect sizes between archaeological contexts in light of the inclusion or exclusion of types and contexts, this routine computes the homogeneity of related assemblages iteratively in two ways, first by leaving out a type on each iteration and second by leaving out a context on each iteration. For details on the arguments, see also [homogeneity](#) and [VB](#)).

Usage

```
homogeneity_LOO(
  x,
  related = NULL,
  unrelated = NULL,
  direction = "UW",
  lambda = 2/3
)

## S3 method for class 'matrix'
homogeneity_LOO(
  x,
  related = NULL,
  unrelated = NULL,
  direction = "UW",
  lambda = 2/3
)

## S3 method for class 'data.frame'
homogeneity_LOO(
  x,
  related = NULL,
  unrelated = NULL,
  direction = "UW",
  lambda = 2/3
)

## S3 method for class 'table'
```

```

homogeneity_LOO(
  x,
  related = NULL,
  unrelated = NULL,
  direction = "UW",
  lambda = 2/3
)

## S3 method for class 'xtabs'
homogeneity_LOO(
  x,
  related = NULL,
  unrelated = NULL,
  direction = "UW",
  lambda = 2/3
)

```

Arguments

<code>x</code>	A data frame or matrix representing a contingency table of counts, with contexts along the columns and types along rows.
<code>related</code>	The related pairs of contexts as a two-column matrix or data frame (contexts between which one anticipates a meaningful relationship). Names must match <code>colnames(x)</code> .
<code>unrelated</code>	The unrelated pairs of contexts as a two-column matrix or data frame (contexts between which one does not anticipate a meaningful relationship). Names must match <code>colnames(x)</code> .
<code>direction</code>	Whether the related or unrelated effect size should come first. Default is "UW"; alternative is "WU".
<code>lambda</code>	Parameter of the Cressie-Read power-divergence statistic. Default is the recommended value of $2/3$.

Value

A list containing:

- `EQ` - The mean quantile expressing the degree of homogeneity among related contexts.
- `EQ_T_mean` - The mean quantile upon iterating over the omission of each type (of the LOO samples).
- `EQ_T_var` - The variance of the LOO type samples.
- `EQ_C_mean` - The mean quantile upon iterating over the omission of each context.
- `EQ_C_var` - The variance of the LOO context samples.

Examples

```

x1 <- c(2, 0, 10, 11, 5)
x2 <- c(1, 1, 17, 23, 3)

```

```

x3 <- c(0, 0, 2, 81, 11)
x4 <- c(3, 18, 9, 0, 23)
x <- matrix(c(x1, x2, x3, x4), ncol = 4)
colnames(x) <- c("surface1", "subsurface1", "surface2", "subsurface2")
rownames(x) <- LETTERS[1:5]

# related pairs
W_contexts <- matrix(c("surface1", "surface2", "subsurface1", "subsurface2"), ncol = 2)

# unrelated pairs (will be automatically created if left NULL)
U_contexts <- matrix(c("surface1", "surface1", "surface2", "subsurface1",
                      "surface2", "subsurface2", "surface1", "subsurface2"), ncol = 2)

homogeneity_L00(x, related = W_contexts, unrelated = U_contexts)

```

idt_ks

Integration in Distribution over Time (KS)

Description

For two one-dimensional temporal point processes X_1 and X_2 , this function estimates the condition of the two processes becoming "integrated in distribution over time" (i.d.t.) at a moment in time t . This is achieved by computing the Kolmogorov-Smirnov distance D between their empirical cumulative distribution functions (e.c.d.f.), as a function $d(t, h)$ that encompasses observations on the support $(t - h, t + h)$. For practical significance in the estimation of d , Monte Carlo permutation of samples is used to estimate the probability that the observed value of d is greater than the distribution of permuted samples (i.e., that observations are interchangeable between X_1 and X_2).

Usage

```

idt_ks(x1, x2 = NULL, stat = FALSE, incr = 0.5, perm = FALSE, n_perm = 1e+05)

## S3 method for class 'numeric'
idt_ks(x1, x2 = NULL, stat = FALSE, incr = 0.5, perm = FALSE, n_perm = 1e+05)

## S3 method for class 'list'
idt_ks(x1, x2 = NULL, stat = FALSE, incr = 0.5, perm = FALSE, n_perm = 1e+06)

```

Arguments

x1	Either a list of vectors or a vector containing observations of sample 1.
x2	If x1 is vector, x2 is vector containing observations of sample 2.
stat	Whether to multiply the distance by $\sqrt{\frac{n_1 n_2}{n_1 + n_2}}$. Default is FALSE.
incr	The increment at which to sample values of t and h . Default is 0.5.

perm	Whether to perform a Monte Carlo permutation across all samples in the input, which will return the probability of an integrating threshold at time t . Default is FALSE.
n_perm	Number of Monte Carlo permutations to perform. Default is 1e5.

Details

The Rcpp-based function `ks_arkhaia` relied upon for faster estimation of the KS distance than `ks.test`.

Value

A list containing a data frame two vectors `x1` and `x2`, the probability that Monte Carlo permutation of samples results in a lower distance than the one obtained. For a list of vectors input in `x1`, a matrix of probabilities pairwise between all vectors.

idt_wass	<i>Integration in Distribution over Time (Wasserstein)</i>
----------	--

Description

For two one-dimensional point processes, this function computes their Wasserstein R1 distances as a function $d(t, h)$, which encompasses observations on the support $(t - h, t + h)$. Monte Carlo permutation of samples is used to estimate the probability that the obtain distances is of practical significance.

Usage

```
idt_wass(
  x1,
  x2 = NULL,
  stat = FALSE,
  scaled = FALSE,
  scaled_by = 0,
  quad_by = 0.001,
  incr = 0.5,
  perm = FALSE,
  n_perm = 10^5
)

## S3 method for class 'numeric'
idt_wass(
  x1,
  x2 = NULL,
  stat = FALSE,
  scaled = FALSE,
  scaled_by = 0,
```

```

    quad_by = 0.001,
    incr = 0.5,
    perm = FALSE,
    n_perm = 10^5
)

## S3 method for class 'list'
idt_wass(
  x1,
  x2 = NULL,
  stat = FALSE,
  scaled = FALSE,
  scaled_by = 0,
  quad_by = 0.001,
  incr = 0.5,
  perm = FALSE,
  n_perm = 10^5
)

```

Arguments

x1	Either a list of vectors or a vector containing observations of sample 1.
x2	If x1 is vector, x2 is vector containing observations of sample 2.
stat	Whether to multiply the distance by $\sqrt{\frac{n_1 n_2}{n_1 + n_2}}$. Default is FALSE.
scaled	Whether to scale the Wasserstein R1 distance used in the course of estimation. Default is FALSE.
scaled_by	The magnitude by which to scale the Wasserstein R1 distance, if scaled is TRUE. Entering 0 (the default) will use the range of the maximum and minimum of the pair of point processes, while entering -1 will use the value of h .
quad_by	The increment of sampling for performing quadrature. Default is 1e-3.
incr	The increment at which to sample values of t and h . Default is 0.5.
perm	Whether to perform a Monte Carlo permutation across all samples in the input, which will return the probability of an integrating threshold at time t . Default is FALSE.
n_perm	Number of Monte Carlo permutations to perform. Default is 10^5.

Value

For two vectors x1 and x2, the probability that Monte Carlo permutation of samples results in a lower empirical scaled Wasserstein R1 distance than the one obtained. For a list of vectors input in x1, a matrix of probabilities pairwise between all vectors.

ks_arkhaia	<i>Kolmogorov-Smirnov Distance</i>
------------	------------------------------------

Description

This function computes the Kolmogorov-Smirnov (KS) distance between two empirical cumulative distribution functions. For more functionality, use [ks.test](#).

Usage

```
ks_arkhaia(x1, x2 = NULL)

## S3 method for class 'numeric'
ks_arkhaia(x1, x2 = NULL)

## S3 method for class 'list'
ks_arkhaia(x1, x2 = NULL)
```

Arguments

x1	A vector of observations, or a list containing both vectors.
x2	A vector of observations, if x1 is not a list.

Details

The Rcpp-based function [ks_arkhaia](#) is used for faster estimation of the KS distance than [ks.test](#).

Value

The KS distance between the two vectors.

log_OR_pair	<i>Log Odds Ratio with Haldane-Anscombe's Correction Pairwise between Columns</i>
-------------	---

Description

For a contingency table with columns representing contexts and rows representing types, computes the log odds ratio of a 2 x 2 contingency table of the presence-absences across each pair of columns. The addition of 0.5 to cells of the contingency table is performed (Anscombe 1956; Haldane 1956).

Usage

```
log_OR_pair(x)

## S3 method for class 'matrix'
log_OR_pair(x)

## S3 method for class 'data.frame'
log_OR_pair(x)

## S3 method for class 'xtabs'
log_OR_pair(x)

## S3 method for class 'table'
log_OR_pair(x)
```

Arguments

x A matrix or data frame with contexts along columns and types along rows.

Value

A matrix giving the log odds ratio between all columns of the input.

Examples

```
x1 <- c(2, 0, 0, 11, 5, 0, 2, 0, 4)
x2 <- c(1, 1, 0, 23, 3, 3, 0, 0, 0)
x3 <- c(1, 0, 0, 0, 10, 0, 4, 0, 1)

x <- data.frame(S1 = x1, S2 = x2, S3 = x3)
rownames(x) <- LETTERS[1:nrow(x)]

log_OR_pair(x)
```

Description

Performs a simple least squares fitting to time indexed data of the form $x(t) = \beta_0 + \sum_{i=1}^n \beta_{1i} \cos(2\pi f t) + \beta_{2i} \sin(2\pi f t)$, using a range of potential frequencies (Vaníček 1969, 1971). Intercept may be omitted.

Usage

```
LSSA(
  x,
  freqs = seq(0.001, 0.5, by = 5e-04),
  intercept = TRUE,
  type = "frequency"
)

## S3 method for class 'time_series'
LSSA(
  x,
  freqs = seq(0.001, 0.5, by = 5e-04),
  intercept = TRUE,
  type = "frequency"
)
```

Arguments

<code>x</code>	A <code>time_series</code> object (see time_series) with only one variate.
<code>freqs</code>	A vector of frequencies to evaluate. By default a grid from 0.001 to 0.5 is tested at an interval of 0.005.
<code>intercept</code>	Whether to include the intercept. Default is TRUE.
<code>type</code>	Type of output. Can be either "frequency" (the default) or "period".

Value

A data frame containing the power and residual sum of squares for each frequency, as well as coefficients.

References

Vaniček P (1969). "Approximate Spectral Analysis by Least-Squares Fit." *Astrophysics and Space Science*, **4**, 387–391. doi:[10.1007/BF00651344](#).

Vaniček P (1971). "Further development and properties of the spectral analysis by least-squares fit." *Astrophysics and Space Science*, **12**, 10–33. doi:[10.1007/BF00656134](#).

Examples

```
t_ <- runif(100, 0, 100)
y <- 5 * sin(0.1 * t_) - 2.3 * cos(0.3 * t_ + 1)
dat <- time_series(t = t_, x = y)
LSSA(dat)
```

LSSA_LFI	<i>Least Squares Spectral Analysis via Lowest Frequency Iteration (LSSA-LFI)</i>
----------	--

Description

Performs a simple least squares fitting to time indexed data of the form $x(t) = \beta_0 + \sum_{i=1}^n \beta_{1i} \cos(2\pi f t) + \beta_{2i} \sin(2\pi f t)$, using an input of frequencies; the lowest frequency peak (not the highest power frequency) is chosen for regression, up to a chosen number of iterations. Intercept may be omitted. The lowest frequency is equivalent to the longest period.

Usage

```
LSSA_LFI(x, n_iter = 1, intercept = TRUE, AIC = FALSE)
```

```
## S3 method for class 'time_series'
```

```
LSSA_LFI(x, n_iter = 1, intercept = TRUE, AIC = FALSE)
```

Arguments

x	A time_series object (see time_series) with only one variate.
n_iter	The number of iterations to run. Default is 1.
intercept	Whether to include the intercept. Default is TRUE.
AIC	If TRUE, only the result that has yielded the lowest AIC is given (Akaike 1973). Default is FALSE.

Value

A list containing:

- A list of the coefficients for each iteration (the intercept is included in the first iteration).
- A vector of the frequencies.
- The residual sum of squares (RSS) after each iteration (decreasing).
- The AIC upon each iteration. (If the parameter AIC is TRUE, this will stop at the lowest AIC value produced by the frequencies tested).

References

Akaike H (1973). "Information Theory and an Extension of the Maximum Likelihood Principle." In Petrov BN, Caski F (eds.), *Proceeding of the Second International Symposium on Information Theory*, 267–281. Akademiai Kiado, Budapest.

LSSA_LFI_candidates *Linear Dependence of LSSA-LFI Candidate Models via AIC*

Description

For a set of time series (namely partitions of set of series) contained in a list, will compute the Akaike Information Criterion (AIC) for each candidate set (Akaike 1973).

Usage

```
LSSA_LFI_candidates(x, sets = NULL, n_iter = 1, intercept = TRUE)

## S3 method for class 'time_series'
LSSA_LFI_candidates(x, sets = NULL, n_iter = 1, intercept = TRUE)
```

Arguments

x	A time_series object (see time_series) with more than one variate.
sets	Candidate sets of variates to evaluate; must be a list of lists containing the indices of the levels in x\$variate. If left NULL, two sets are evaluated: all series pooled together [1] and all series kept separate [2].
n_iter	The number of iterations to run for the least squares spectral analysis via lowest frequency iteration (LSSA-LFI). Default is 1.
intercept	Whether to include the intercept in the least squares spectral analysis via lowest frequency iteration (LSSA-LFI). Default is TRUE.

Value

The index of the set yielding the lowest AIC. If sets is NULL, the output of [1] indicates linear dependence; if the output is [2], linear independence.

References

Akaike H (1973). “Information Theory and an Extension of the Maximum Likelihood Principle.” In Petrov BN, Caski F (eds.), *Proceeding of the Second International Symposium on Information Theory*, 267–281. Akademiai Kiado, Budapest.

LSSA_LFI_comp	<i>Partitioned LSSA-LFI Model Terms and AIC: Composite Number of Parameters</i>
---------------	---

Description

For multiple time series contained in a list, this function will compute the Akaike Information Criterion (AIC) for the best-fitting models particular to each series, including the case of an intercept-only model. This function can be performed after [LSSA_LFI_candidates](#) and [repartition](#) to determine the appropriate number of terms. For the situation where all time series must have the same number of terms, [LSSA_LFI_multi](#) should be used, but this function should choose a better fitting model, since it adjusts the number of parameters for each dataset.

Usage

```
LSSA_LFI_comp(x, intercept = TRUE)

## S3 method for class 'time_series'
LSSA_LFI_comp(x, intercept = TRUE)
```

Arguments

x	A time_series object (see time_series).
intercept	Whether to include the intercept in the least squares spectral analysis via lowest frequency iteration (LSSA-LFI). Default is TRUE.

Value

A list giving the number of parameters (k) and residual sum of squares (rss) for each dataset in x, along with the AIC for the partitioned model.

LSSA_LFI_epoch	<i>Period-Variable Pairwise Selection of Linearly Dependent LSSA-LFI Models</i>
----------------	---

Description

Evaluate pairwise linear dependence between two time series using a LSSA-LFI validated model selection (see [LSSA_LFI_validated](#)), with variable length time period.

Usage

```

LSSA_LFI_epoch(
  x,
  pair = NULL,
  n_iter = 1,
  intercept = TRUE,
  t_range = NULL,
  h_range = NULL,
  t_grid = 1,
  h_grid = 1
)

## S3 method for class 'list'
LSSA_LFI_epoch(
  x,
  pair = NULL,
  n_iter = 1,
  intercept = TRUE,
  t_range = NULL,
  h_range = NULL,
  t_grid = 1,
  h_grid = 1
)

```

Arguments

<code>x</code>	A list of data frames.
<code>pair</code>	The pair of commodities to evaluate in the list <code>x</code> , given as indices.
<code>n_iter</code>	The number of iterations to run. Default is 1.
<code>intercept</code>	Whether to include the intercept in the least squares spectral analysis via lowest frequency iteration (LSSA-LFI). Default is TRUE.
<code>t_range</code>	The range of the time period. Default are the minimum and maximum dates spanned by the data in <code>x</code> .
<code>h_range</code>	The range of the potential windows of h . Default is from 0 to entire span of <code>t_range</code> .
<code>t_grid</code>	The length of interval along which to sample t . Default is 1.
<code>h_grid</code>	The length of interval along which to sample h . Default is 1.

Value

A matrix giving the probability of linear dependence, predicated upon time t and window length h .

LSSA_LFI_model	<i>LSSA-LFI Model</i>
----------------	-----------------------

Description

Generates a data frame of values $x(t)$ of the model generated by LSSA-LFI (see [LSSA_LFI](#)).

Usage

```
LSSA_LFI_model(x, t_ = NULL, n_iter = 1, intercept = TRUE, label = "model")

## S3 method for class 'time_series'
LSSA_LFI_model(x, t_ = NULL, n_iter = 1, intercept = TRUE, label = "model")
```

Arguments

<code>x</code>	A <code>time_series</code> object (see time_series) with only one variate.
<code>t_</code>	A vector giving samples range of t for computing $x(t)$. Default is from the minimum to maximum time index sampled at 0.01 intervals.
<code>n_iter</code>	The number of iterations to run. Default is 1.
<code>intercept</code>	Whether to include the intercept in the least squares spectral analysis via lowest frequency iteration (LSSA-LFI). Default is TRUE.
<code>label</code>	Default is "model".

Value

A data frame containing $t, x(t)$.

LSSA_LFI_multi	<i>Partitioned LSSA-LFI Model Terms and AIC</i>
----------------	---

Description

For multiple time series contained in a [time_series](#) object, this function will compute the Akaike Information Criterion (AIC) as the number of terms (`n_iter`) is increased. This function can be performed after [LSSA_LFI_candidates](#) and [repartition](#) to determine the appropriate number of terms.

Usage

```
LSSA_LFI_multi(x, n_iter = 1, intercept = TRUE, AIC = FALSE)

## S3 method for class 'time_series'
LSSA_LFI_multi(x, n_iter = 1, intercept = TRUE, AIC = FALSE)
```

Arguments

x	A time_series object (see time_series).
n_iter	The number of iterations to run for the least squares spectral analysis via lowest frequency iteration (LSSA-LFI). Default is 1.
intercept	Whether to include the intercept in the least squares spectral analysis via lowest frequency iteration (LSSA-LFI). Default is TRUE.
AIC	If TRUE, only the result that has yeilded the lowest AIC is given (Akaike 1973). Default is FALSE.

Value

A data frame giving the number of iterations and the AIC for each.

References

Akaike H (1973). “Information Theory and an Extension of the Maximum Likelihood Principle.” In Petrov BN, Caski F (eds.), *Proceeding of the Second International Symposium on Information Theory*, 267–281. Akademiai Kiado, Budapest.

LSSA_LFI_pairwise	<i>Pairwise Selection of Linearly Dependent LSSA-LFI Models</i>
-------------------	---

Description

Evaluate pairwise linear dependence between observations using a LSSA-LFI validated model selection (see [LSSA_LFI_validated](#)).

Usage

```
LSSA_LFI_pairwise(x, n_iter = 1, intercept = TRUE)

## S3 method for class 'time_series'
LSSA_LFI_pairwise(x, n_iter = 1, intercept = TRUE)
```

Arguments

x	A time_series object (see time_series) with at least three variates.
n_iter	The number of iterations to run. Default is 1.
intercept	Whether to include the intercept in the least squares spectral analysis via lowest frequency iteration (LSSA-LFI). Default is TRUE.

Value

An upper-triangular matrix, containing the probability of linear dependence between series.

LSSA_LFI_validated	<i>Validated Linear Dependence via LSSA-LFI</i>
--------------------	---

Description

Probability of linear dependence between two groups of time series observations using a LSSA-LFI model selection (see [LSSA_LFI_candidates](#)), on the basis of the inclusion of a third "attendant" variate. The third attendant variate is selected from the remaining time series in the list, hence the input list must include at least three times series.

Usage

```
LSSA_LFI_validated(x, pair = NULL, n_iter = 1, intercept = TRUE)

## S3 method for class 'time_series'
LSSA_LFI_validated(x, pair = NULL, n_iter = 1, intercept = TRUE)
```

Arguments

x	A time_series object (see time_series) with at least three variates.
pair	The pair of series to evaluate in the x, given levels in x\$variate.
n_iter	The number of iterations to run. Default is 1.
intercept	Whether to include the intercept in the least squares spectral analysis via lowest frequency iteration (LSSA-LFI). Default is TRUE.

Value

The probability of linear dependence between two time series, in which 1 indicates linear dependence almost surely and 0 indicates independence.

pa_matrix	<i>Presence-Absence Matrix</i>
-----------	--------------------------------

Description

Create a 2 x 2 contingency table of the presence/absence of a given type

Usage

```
pa_matrix(x)

## S3 method for class 'matrix'
pa_matrix(x)

## S3 method for class 'data.frame'
```

```

pa_matrix(x)

## S3 method for class 'table'
pa_matrix(x)

## S3 method for class 'xtabs'
pa_matrix(x)

```

Arguments

x A matrix or data frame where the two columns indicate contexts and the rows indicate types, with each cell containing counts of types in a context.

Value

A 2 x 2 contingency table of the counts of types present in both, either, or neither context.

Examples

```

x1 <- c(2, 0, 10, 11, 5)
x2 <- c(1, 1, 17, 23, 3)

x <- data.frame(S1 = x1, S2 = x2)
rownames(x) <- LETTERS[1:5]

pa_matrix(x)

```

pois_rcens	<i>Least Squares Fit of Poisson Distribution for Random Right-Censored Data</i>
------------	---

Description

For a matrix of cross-tabulated counts of observations which constitute a minimum threshold, this function estimates the rate parameter column-wise, either retaining or omitting zeros, by a least-squares approach.

Usage

```

pois_rcens(x, lambda_grid = seq(0.01, 100, by = 0.01), omit_zero = TRUE)

## S3 method for class 'matrix'
pois_rcens(x, lambda_grid = seq(0.01, 100, by = 0.01), omit_zero = TRUE)

## S3 method for class 'data.frame'
pois_rcens(x, lambda_grid = seq(0.01, 100, by = 0.01), omit_zero = TRUE)

```

```
## S3 method for class 'xtabs'
pois_rcens(x, lambda_grid = seq(0.01, 100, by = 0.01), omit_zero = TRUE)

## S3 method for class 'table'
pois_rcens(x, lambda_grid = seq(0.01, 100, by = 0.01), omit_zero = TRUE)
```

Arguments

<code>x</code>	A matrix of cross-tabulated counts.
<code>lambda_grid</code>	The resolution at which to sample for the rate parameter. Default is <code>seq(0.1, 100, by = 0.01)</code> .
<code>omit_zero</code>	Whether to omit zeros. Default is <code>TRUE</code> .

Value

A vector of the rate parameters for each column.

Examples

```
x1 <- c(1,2,2,5,7,0,0)
x2 <- c(9,2,5,15,7,90,0)
x <- matrix(c(x1,x2), ncol = 2)

pois_rcens(x)
pois_rcens(x, omit_zero = FALSE)
```

repartition

Repartition (Regroup) Variates

Description

Given multiple time series in a `time_series` object (`x`), group the variates into a new set of variates, expressed as a list of the variate names (`set`). Returns a `time_series` object using the new partitioning of variates.

Usage

```
repartition(x, set = NULL)

## S3 method for class 'time_series'
repartition(x, set = NULL)
```

Arguments

<code>x</code>	A <code>time_series</code> object containing multiple time series, each of which should be a matrix or data frame with time index in the first column and value in the second.
<code>set</code>	A list of vectors containing the names of variates in each of the new groups.

Value

A list containing the data according to the partition structure described by set.

Examples

```
idx <- runif(40, 0, 100)
x1 <- runif(10, 0, 1) + 3
x2 <- runif(10, 0, 1) + 2
x3 <- runif(10, 0, 1) + 1
x4 <- runif(10, 0, 1)
x <- time_series(t = idx, x = c(x1,x2,x3,x4),
  variate = c(rep("x1", 10), rep("x2", 10),
    rep("x3", 10), rep("x4", 10)))

repartition(x, set = list(c("x1", "x4"), "x2", "x3"))
```

th_idt_ks

Thresholds of Integration in Distribution over Time (KS)

Description

For two one-dimensional temporal point processes X_1 and X_2 , this function estimates the condition of the two processes becoming "integrated in distribution over time" (i.d.t.) at a moment in time t . This is achieved by computing the Kolmogorov-Smirnov distance D between their empirical cumulative distribution functions (e.c.d.f.), before and after t for some duration h , and taking the difference between the distances $\theta = D_{before} - D_{after}$. For practical significance in the estimation of θ , Monte Carlo permutation of samples is used to estimate the probability that the observed value of θ is greater than the distribution of permuted samples (i.e., that observations are interchangeable between X_1 and X_2).

Usage

```
th_idt_ks(
  x1,
  x2 = NULL,
  stat = FALSE,
  incr = 0.5,
  perm = FALSE,
  n_perm = 1e+05
)

## S3 method for class 'numeric'
th_idt_ks(
  x1,
  x2 = NULL,
  stat = FALSE,
  incr = 0.5,
```

```

    perm = FALSE,
    n_perm = 1e+05
  )

  ## S3 method for class 'list'
  th_idt_ks(
    x1,
    x2 = NULL,
    stat = FALSE,
    incr = 0.5,
    perm = FALSE,
    n_perm = 1e+06
  )

```

Arguments

x1	Either a list of vectors or a vector containing observations of sample 1.
x2	If x1 is vector, x2 is vector containing observations of sample 2.
stat	Whether to multiply the distance by $\sqrt{\frac{n_1 n_2}{n_1 + n_2}}$. Default is FALSE.
incr	The increment at which to sample values of t and h . Default is 0.5.
perm	Whether to perform a Monte Carlo permutation across all samples in the input, which will return the probability of an integrating threshold at time t . Default is FALSE.
n_perm	Number of Monte Carlo permutations to perform. Default is 1e5.

Value

A list containing a data frame two vectors x1 and x2, the probability that Monte Carlo permutation of samples results in a lower distance than the one obtained. For a list of vectors input in x1, a matrix of probabilities pairwise between all vectors.

th_idt_wass

Thresholds of Integration in Distribution over Time (Wasserstein)

Description

For two one-dimensional point processes, this function estimates the condition of the two processes becoming i.d.t. at a moment of time t considering a window of time around that moment, $(t - h, t + h)$. This is achieved by computing the difference of the empirical scaled Wasserstein R1 distances between the two processes before and after t . Monte Carlo permutation of samples is used to estimate the probability that the obtain distances is of practical significance.

Usage

```

th_idt_wass(
  x1,
  x2 = NULL,
  stat = FALSE,
  scaled = FALSE,
  scaled_by = 0,
  quad_by = 0.001,
  incr = 0.5,
  perm = FALSE,
  n_perm = 10^5
)

## S3 method for class 'numeric'
th_idt_wass(
  x1,
  x2 = NULL,
  stat = FALSE,
  scaled = FALSE,
  scaled_by = 0,
  quad_by = 0.001,
  incr = 0.5,
  perm = FALSE,
  n_perm = 10^5
)

## S3 method for class 'list'
th_idt_wass(
  x1,
  x2 = NULL,
  stat = FALSE,
  scaled = FALSE,
  scaled_by = 0,
  quad_by = 0.001,
  incr = 0.5,
  perm = FALSE,
  n_perm = 10^5
)

```

Arguments

x1	Either a list of vectors or a vector containing observations of sample 1.
x2	If x1 is vector, x2 is vector containing observations of sample 2.
stat	Whether to multiply the distance by $\sqrt{\frac{n_1 n_2}{n_1 + n_2}}$. Default is FALSE.
scaled	Whether to scale the Wasserstein R1 distance used in the course of estimation. Default is FALSE.

scaled_by	The magnitude by which to scale the Wasserstein R1 distance, if scaled is TRUE. Entering 0 (the default) will use the range of the maximum and minimum of the pair of point processes, while entering -1 will use the value of h .
quad_by	The increment of sampling for performing quadrature. Default is $1e-3$.
incr	The increment at which to sample values of t and h . Default is 0.5.
perm	Whether to perform a Monte Carlo permutation across all samples in the input, which will return the probability of an integrating threshold at time t . Default is FALSE.
n_perm	Number of Monte Carlo permutations to perform. Default is 10^5 .

Value

For two vectors x_1 and x_2 , the probability that Monte Carlo permutation of samples results in a lower empirical scaled Wasserstein R1 distance than the one obtained. For a list of vectors input in x_1 , a matrix of probabilities pairwise between all vectors.

time_series	<i>Create a Time Series Object</i>
-------------	------------------------------------

Description

Creates an object of class `time_series` from a matrix or data frame, or by giving a filename for a csv file in the working directory. The first column contains time indices (t), the second the value of observations (x), both of which must be numeric, and the third contains a label as a factor column (variate). Multiple time series are contained in the same `time_series` object, distinguished by different labels.

Usage

```
time_series(
  input,
  t = NULL,
  x = NULL,
  variate = NULL,
  log = FALSE,
  inverse = FALSE,
  standard = FALSE,
  ...
)

## S3 method for class 'matrix'
time_series(
  input,
  t = NULL,
  x = NULL,
  variate = NULL,
```

```

    log = FALSE,
    inverse = FALSE,
    standard = FALSE,
    ...
)

## S3 method for class 'data.frame'
time_series(
  input,
  t = NULL,
  x = NULL,
  variate = NULL,
  log = FALSE,
  inverse = FALSE,
  standard = FALSE,
  ...
)

## S3 method for class 'character'
time_series(
  input,
  t = NULL,
  x = NULL,
  variate = NULL,
  log = FALSE,
  inverse = FALSE,
  standard = FALSE,
  ...
)

## S3 method for class 'numeric'
time_series(
  input = NULL,
  t = NULL,
  x = NULL,
  variate = NULL,
  log = FALSE,
  inverse = FALSE,
  standard = FALSE,
  ...
)

```

Arguments

input	A matrix, data frame, or a filename given as character.
t	If used, ignores ime indices of the observations; requires also input x
x	The values of the
variate	The label of the variate if

log	Whether to take the logarithm of observations for x (default is FALSE).
inverse	Whether to use the inverse of observations (default is FALSE). If both log and inverse are true, the log inverse ($\log(1/x)$) is used.
standard	Whether to standardize the observations, using $z = (x - \bar{x})/s$, grouped by variate (default is FALSE). This operation is performed after log and/or inverse.
...	Additional arguments for importing from csv. • header If a filename, whether the csv file contains a header (default TRUE). • tabular If a filename, whether the data are presented in a tabular format, as opposed to "long" (tidyverse) format (default TRUE).

Value

A time_series object.

Examples

```
idx <- runif(100,0,100)
x <- cumsum(rnorm(100,0,1))
label <- rep("x", 100)
mat <- matrix(c(idx, x, label), ncol = 3)
time_series(mat)

dat <- data.frame(idx, x, label)
time_series(dat)

time_series(t = idx, x = x, variate = label)
```

trace_pair	<i>Trace Coefficient Pairwise between Columns</i>
------------	---

Description

For a contingency table with columns representing contexts and rows representing types, computes the trace of a 2 x 2 contingency table of the presence-absences across each pair of columns.

Usage

```
trace_pair(x)

## S3 method for class 'matrix'
trace_pair(x)

## S3 method for class 'data.frame'
trace_pair(x)

## S3 method for class 'xtabs'
```

```
trace_pair(x)

## S3 method for class 'table'
trace_pair(x)
```

Arguments

x A matrix or data frame with contexts along columns and types along rows.

Value

A matrix giving the trace between all columns of the input.

Examples

```
x1 <- c(2, 0, 0, 11, 5, 0, 2, 0, 4)
x2 <- c(1, 1, 0, 23, 3, 3, 1, 1, 1)
x3 <- c(1, 0, 0, 0, 10, 0, 4, 0, 0)

x <- data.frame(S1 = x1, S2 = x2, S3 = x3)
rownames(x) <- LETTERS[1:nrow(x)]

trace_pair(x)
```

trim_epoch	<i>Trim to Epoch</i>
------------	----------------------

Description

For a data frame in which the first column contains a time index and the second column observations, trim the data frame to include observations only with a given epoch (time period).

Usage

```
trim_epoch(x, epoch = NULL)

## S3 method for class 'time_series'
trim_epoch(x, epoch = NULL)
```

Arguments

x A time_series object (see [time_series](#)).

epoch A numeric vector given the start and end time indices of the epoch.

Value

A data frame containing only those observations with time indices within the epoch.

trunc_pois	<i>Resampled Contingency Table via a Truncated Poisson for Random Right-Censored Data</i>
------------	---

Description

For a matrix of cross-tabulated counts of observations which constitute a minimum threshold, returns a contingency table whose counts are sampled according to a truncated Poisson distribution, whose rate parameter is determined column-wise (see [pois_rcens](#)).

Usage

```
trunc_pois(x, lambda_grid = seq(0.01, 100, by = 0.01), omit_zero = TRUE)

## S3 method for class 'matrix'
trunc_pois(x, lambda_grid = seq(0.01, 100, by = 0.01), omit_zero = TRUE)

## S3 method for class 'data.frame'
trunc_pois(x, lambda_grid = seq(0.01, 100, by = 0.01), omit_zero = TRUE)

## S3 method for class 'xtabs'
trunc_pois(x, lambda_grid = seq(0.01, 100, by = 0.01), omit_zero = TRUE)

## S3 method for class 'table'
trunc_pois(x, lambda_grid = seq(0.01, 100, by = 0.01), omit_zero = TRUE)
```

Arguments

x	A matrix of cross-tabulated counts.
lambda_grid	The resolution at which to sample for the rate parameter. Default is <code>seq(0.1, 100, by = 0.01)</code> .
omit_zero	Whether to omit zeros. Default is <code>TRUE</code> .

Value

A contingency table Y of the same size as X , with y_{ij} drawn according to a truncated Poisson distribution that ensures $y_{ij} \geq x_{ij}$.

Examples

```
x1 <- c(1,2,2,5,7,0,0)
x2 <- c(9,2,5,15,7,90,0)
x <- matrix(c(x1,x2), ncol = 2)

pois_rcens(x)
pois_rcens(x, omit_zero = FALSE)
```

VB	<i>Bias-Corrected Cramer's V</i>
----	----------------------------------

Description

For a matrix of cross-tabulated counts of observations, estimates Cramer's V using Bergsma's bias correction (Bergsma 2013), using the Cressie-Read power divergence statistic (see [CR](#)).

Usage

```
VB(x, lambda = 2/3)

## S3 method for class 'matrix'
VB(x, lambda = 2/3)

## S3 method for class 'data.frame'
VB(x, lambda = 2/3)

## S3 method for class 'xtabs'
VB(x, lambda = 2/3)

## S3 method for class 'table'
VB(x, lambda = 2/3)
```

Arguments

x	A matrix or data frame of cross-tabulated counts.
lambda	Parameter of the Cressie-Read power-divergence statistic. Default is the recommended value of 2/3.

Value

Bergsma's bias-corrected estimate of Cramer's V .

References

Bergsma W (2013). "A Bias-Correction for Cramér's V and Tschuprow's ST ." *Journal of the Korean Statistical Society*, **42**, 323–328. doi:[10.1016/j.jkss.2012.10.002](https://doi.org/10.1016/j.jkss.2012.10.002).

Examples

```
x1 <- c(2, 0, 10, 11, 5)
x2 <- c(1, 1, 17, 23, 3)
x3 <- c(0, 0, 2, 81, 11)

x <- matrix(c(x1, x2, x3), ncol = 3)

VB_pair(x)
```

```
VB(x, lambda = 1)
```

VB_LOO_type	<i>Leave-One-Out Type Routine for Pairwise Effect Sizes of Archaeological Contexts</i>
-------------	--

Description

To evaluate the stability of estimates of effect sizes between archaeological contexts in light of the inclusion or exclusion of a given type, this routine computes bias-corrected Cramer's V [VB](#) omitting a type on each iteration.

Usage

```
VB_LOO_type(x, lambda = 2/3)

## S3 method for class 'matrix'
VB_LOO_type(x, lambda = 2/3)

## S3 method for class 'data.frame'
VB_LOO_type(x, lambda = 2/3)

## S3 method for class 'xtabs'
VB_LOO_type(x, lambda = 2/3)

## S3 method for class 'table'
VB_LOO_type(x, lambda = 2/3)
```

Arguments

x	A contingency table as a matrix or data frame expressing counts, with contexts along columns and types along rows.
lambda	Parameter of the Cressie-Read power-divergence statistic. Default is the recommended value of 2/3.

Value

A three-dimensional array of the pairwise context-by-context effect sizes given the omission of a type, for all types given in the input data frame.

Examples

```
x1 <- c(2, 0, 10, 11, 5)
x2 <- c(1, 1, 17, 23, 3)
x3 <- c(0, 0, 2, 81, 11)

x <- matrix(c(x1, x2, x3), ncol = 3)
```

```
rownames(x) <- LETTERS[1:nrow(x)]
colnames(x) <- c("S1", "S2", "S3")

VB_LOO_type(x)
```

VB_pair

Bias-Corrected Cramer's V Pairwise between Columns

Description

For a matrix or data frame of cross-tabulated counts of observations, estimates Cramer's V using Bergsma's bias correction (Bergsma 2013) by subsetting the table by pairs of columns (see [VB](#)). In subsetting, zero row/columns are automatically removed from the subset matrix.

Usage

```
VB_pair(x, lambda = 2/3)

## S3 method for class 'matrix'
VB_pair(x, lambda = 2/3)

## S3 method for class 'data.frame'
VB_pair(x, lambda = 2/3)

## S3 method for class 'xtabs'
VB_pair(x, lambda = 2/3)

## S3 method for class 'table'
VB_pair(x, lambda = 2/3)
```

Arguments

x	A matrix or data frame of cross-tabulated counts.
lambda	Parameter of the Cressie-Read power-divergence statistic. Default is the recommended value of 2/3.

Value

A matrix of Bergsma's bias-corrected estimate of Cramer's V, pairwise between columns of the input matrix, of class `effect_sizes`.

References

Bergsma W (2013). "A Bias-Correction for Cramér's V and Tschuprow's ST ." *Journal of the Korean Statistical Society*, **42**, 323–328. doi:[10.1016/j.jkss.2012.10.002](https://doi.org/10.1016/j.jkss.2012.10.002).

Examples

```

x1 <- c(2, 0, 10, 11, 5)
x2 <- c(1, 1, 17, 23, 3)
x3 <- c(0, 0, 2, 81, 11)

x <- matrix(c(x1, x2, x3), ncol = 3)

VB_pair(x)
VB_pair(x, lambda = 1)

```

VB_trunc_pois

Bias-Corrected Cramer's V for Random Right-Censored Data

Description

Given a matrix of cross-tabulated counts of observations which constitute a minimum threshold, returns the distribution of bias-corrected Cramer's V by resampling from a truncated Poisson distribution, with rates determined column-wise (see [VB, trunc_pois](#)).

Usage

```

VB_trunc_pois(
  x,
  lambda = 2/3,
  lambda_grid = seq(0.01, 100, by = 0.01),
  omit_zero = TRUE,
  n_iter = 10^5
)

## S3 method for class 'matrix'
VB_trunc_pois(
  x,
  lambda = 2/3,
  lambda_grid = seq(0.01, 100, by = 0.01),
  omit_zero = TRUE,
  n_iter = 10^5
)

## S3 method for class 'data.frame'
VB_trunc_pois(
  x,
  lambda = 2/3,
  lambda_grid = seq(0.01, 100, by = 0.01),
  omit_zero = TRUE,
  n_iter = 10^5
)

```

```
## S3 method for class 'xtabs'
VB_trunc_pois(
  x,
  lambda = 2/3,
  lambda_grid = seq(0.01, 100, by = 0.01),
  omit_zero = TRUE,
  n_iter = 10^5
)

## S3 method for class 'table'
VB_trunc_pois(
  x,
  lambda = 2/3,
  lambda_grid = seq(0.01, 100, by = 0.01),
  omit_zero = TRUE,
  n_iter = 10^5
)
```

Arguments

<code>x</code>	A matrix of cross-tabulated counts.
<code>lambda</code>	The value of <code>lambda</code> (default is 2/3) for the Cressie-Read power divergence statistic used to estimate the bias-corrected Cramer's V . Default is 2/3.
<code>lambda_grid</code>	The resolution at which to sample for the rate parameter. Default is <code>seq(0.1, 100, by = 0.01)</code> .
<code>omit_zero</code>	Whether to omit zeros. Default is TRUE.
<code>n_iter</code>	Number of samples of V to take. Default is 10^5 .

Value

A contingency table Y of the same size as X , with y_{ij} drawn according to a truncated Poisson distribution, $y_{ij} \geq x_{ij}$.

Examples

```
x1 <- c(1,2,2,5,7,0,0)
x2 <- c(9,2,5,15,7,90,0)
x <- matrix(c(x1,x2), ncol = 2)

VB_trunc_pois(x, n_iter = 10^2)
VB_trunc_pois(x, omit_zero = FALSE, n_iter = 10^2)
```

wass_R1

Wasserstein R1 Distance between Two e.c.d.f.

Description

For two or more one-dimensional point processes, computes their Wasserstein distance via their empirical cumulative density functions (e.c.d.f.), adjusting for sample size and the scale of their durations (Vasershtein 1969; Anderson 1962; Ramdas et al. 2017), using numerical quadrature:

$$W_{1s} = \frac{1}{s} \int_{\mathbf{R}} |F_1(x) - F_2(x)| dx$$

where s is a scaling coefficient, namely either the range of the point process or its support. More than two processes result in pairwise estimation of processes.

Usage

```
wass_R1(
  x1,
  x2 = NULL,
  scaled = FALSE,
  scaled_by = 0,
  quad_by = 0.001,
  perm = FALSE,
  n_perm = 10^5
)

## S3 method for class 'numeric'
wass_R1(
  x1,
  x2 = NULL,
  scaled = FALSE,
  scaled_by = 0,
  quad_by = 0.001,
  perm = FALSE,
  n_perm = 10^5
)

## S3 method for class 'list'
wass_R1(
  x1,
  x2 = NULL,
  scaled = FALSE,
  scaled_by = 0,
  quad_by = 0.001,
  perm = FALSE,
  n_perm = 10^5
)
```

Arguments

x1	Either a list of vectors or a vector containing observations of sample 1.
x2	If x1 is vector, x2 is vector containing observations of sample 2.
scaled	Whether to scale the resulting Wasserstein R1 distance. Default is FALSE.
scaled_by	The magnitude by which to scale the Wasserstein R1 distance, if scaled is TRUE. Entering 0 (the default) will use the range of the maximum and minimum of the pair of point processes.
quad_by	The increment of sampling for performing quadrature. Default is 1e-3.
perm	Whether to perform a Monte Carlo permutation across all samples in the input. Default is FALSE.
n_perm	Number of Monte Carlo permutations to perform. Default is 10^5.

Value

For two vectors x1 and x2, the empirical scaled Wasserstein R1 distance between the two e.c.d.f. of their samples. For a list of vectors input in x1, a matrix of the empirical scaled Wasserstein R1 distances pairwise between all vectors.

References

- Anderson TW (1962). “On the Distribution of the Two-Sample Cramer–von Mises Criterion.” *Annals of Mathematical Statistics*, **33**, 1148–1159. doi:[10.1214/aoms/1177704477](https://doi.org/10.1214/aoms/1177704477).
- Ramdas A, García Trillos N, Cuturi M (2017). “On Wasserstein Two-Sample Testing and Related Families of Nonparametric Tests.” *Entropy*, **19.47**, 1–15. doi:[10.3390/e19020047](https://doi.org/10.3390/e19020047).
- Vasershtein LN (1969). “Markov processes over denumerable products of spaces, describing large systems of automata.” *Problemy Peredači Informacii*, **5**, 64–72.

Index

CR, [2](#), [32](#)

dtem, [4](#)

homogeneity, [5](#), [7](#)

homogeneity_L00, [7](#)

idt_ks, [9](#)

idt_wass, [10](#)

ks.test, [10](#), [12](#)

ks_arkhaia, [10](#), [12](#), [12](#)

log_OR_pair, [12](#)

LSSA, [13](#)

LSSA_LFI, [15](#), [19](#)

LSSA_LFI_candidates, [16](#), [17](#), [19](#), [21](#)

LSSA_LFI_comp, [17](#)

LSSA_LFI_epoch, [17](#)

LSSA_LFI_model, [19](#)

LSSA_LFI_multi, [17](#), [19](#)

LSSA_LFI_pairwise, [20](#)

LSSA_LFI_validated, [17](#), [20](#), [21](#)

pa_matrix, [21](#)

pois_rcens, [22](#), [31](#)

repartition, [17](#), [19](#), [23](#)

th_idt_ks, [24](#)

th_idt_wass, [25](#)

time_series, [14–17](#), [19–21](#), [23](#), [27](#), [30](#)

trace_pair, [29](#)

trim_epoch, [30](#)

trunc_pois, [31](#), [35](#)

VB, [7](#), [32](#), [33–35](#)

VB_L00_type, [33](#)

VB_pair, [6](#), [34](#)

VB_trunc_pois, [35](#)

wass_R1, [37](#)