

Package: climatestatsr (via r-universe)

June 18, 2026

Type Package

Title Statistical Tools for Climate Change Analysis

Version 0.1.2

Date 2026-05-21

Maintainer Sadikul Islam <sadikul.islamiasri@gmail.com>

Description A comprehensive collection of statistical functions for climate change research. Provides tools for temporal trend detection based on the Mann-Kendall (MK) test (Mann 1945 <doi:10.2307/1907187>; Kendall 1975, ISBN:0852641990) and Sen's slope (Sen 1968 <doi:10.2307/2285891>), spatial autocorrelation using Moran's I (Moran 1950 <doi:10.2307/2332142>), extreme value analysis using the Generalised Extreme Value (GEV) distribution and Peaks-Over-Threshold (POT) method (Coles 2001 <doi:10.1007/978-1-4471-3675-0>), standardised drought indices including the Standardised Precipitation Index (SPI; McKee et al. 1993) and the Standardised Precipitation Evapotranspiration Index (SPEI; Vicente-Serrano et al. 2010 <doi:10.1175/2009JCLI2909.1>), and formal detection-attribution methods via optimal fingerprint regression and Empirical Orthogonal Function (EOF) analysis (Allen and Tett 1999 <doi:10.1007/s003820050291>), and apparent temperature via the heat index (Steadman 1979 <doi:10.1175/1520-0450(1979)018%3C0861:TAOSPI%3E2.0.CO;2>). Suitable for both station-level time series and gridded climate fields.

License GPL-3

Encoding UTF-8

Language en-US

Depends R (>= 4.0.0)

Imports stats, graphics, utils

Suggests ggplot2 (>= 3.4.0), sf (>= 1.0-0), ncd4 (>= 1.21), testthat (>= 3.0.0), knitr (>= 1.42), rmarkdown (>= 2.20)

VignetteBuilder knitr

NeedsCompilation no
ByteCompile true
RoxygenNote 7.3.3
Author Sadikul Islam [aut, cre] (ORCID:
<https://orcid.org/0000-0003-2924-7122>), Rajesh Kaushal [aut]
Repository <https://cran.r-universe.dev>
Date/Publication 2026-06-18 17:35:46 UTC
RemoteUrl <https://github.com/cran/climatestatsr>
RemoteRef HEAD
RemoteSha 84bfc86c036f23f11a3e7468bf641909562a99fd

Contents

climatestatsr-package	3
aggregate_climate	4
anomaly_baseline	5
autocorrelation_climate	6
change_point_detection	7
climate_summary	8
climate_test_methods	9
cluster_climate_zones	10
cold_spell_detection	11
detection_attribution	12
diurnal_temp_range	13
drought_spell	14
elevation_lapse_rate	15
extreme_value_index	16
fill_gaps_climate	17
fingerprint_analysis	18
fit_gev	19
frost_days	21
growing_degree_days	21
heat_index	22
heat_wave_detection	23
homogenize_series	24
hot_cold_spots	25
mk_test	26
morans_i	28
optimal_fingerprint	30
pdsi_simple	31
peaks_over_threshold	32
print.gev_fit	33
return_period	34
rgev_sim	35
rolling_trend	35

seasonal_decompose_climate	36
sens_slope	38
spatial_anomaly	39
spatial_interpolate	40
spatial_trend_field	41
spei	42
spi	43
standardize_climate	44
temporal_homogeneity	45
trend_significance	46
wind_chill	47
Index	49

climatestatsr-package *Statistical Tools for Climate Change Analysis*

Description

A comprehensive collection of statistical functions for climate change research. Provides tools for temporal trend detection, spatial analysis, extreme event statistics, standardised climate indices, and formal detection-attribution methods.

Details

The package is organised into five function families:

- **Temporal analysis:** `mk_test`, `sens_slope`, `change_point_detection`, `seasonal_decompose_climate`, `rolling_trend`, `temporal_homogeneity`, `trend_significance`, `autocorrelation_climate`
- **Spatial analysis:** `morans_i`, `hot_cold_spots`, `spatial_interpolate`, `spatial_trend_field`, `cluster_climate_zones`, `spatial_anomaly`, `elevation_lapse_rate`
- **Extreme events:** `fit_gev`, `return_period`, `peaks_over_threshold`, `heat_wave_detection`, `cold_spell_detection`, `drought_spell`, `extreme_value_index`
- **Climate indices:** `spi`, `spei`, `pdsi_simple`, `heat_index`, `wind_chill`, `frost_days`, `growing_degree_days`, `diurnal_temp_range`
- **Attribution and utilities:** `detection_attribution`, `fingerprint_analysis`, `optimal_fingerprint`, `fill_gaps_climate`, `homogenize_series`, `aggregate_climate`, `anomaly_baseline`, `standardize_climate`, `climate_summary`

Author(s)

Sadikul Islam <sadikul.islam@climate-research.org> ([ORCID](#))

Maintainer: Sadikul Islam <sadikul.islam@climate-research.org>

References

- Mann, H. B. (1945). Nonparametric tests against trend. *Econometrica* **13**(3), 245–259. doi:[10.2307/1907187](https://doi.org/10.2307/1907187)
- Sen, P. K. (1968). Estimates of the regression coefficient based on Kendall's tau. *Journal of the American Statistical Association* **63**(324), 1379–1389. doi:[10.2307/2285891](https://doi.org/10.2307/2285891)
- Coles, S. (2001). *An Introduction to Statistical Modeling of Extreme Values*. Springer-Verlag, London. doi:[10.1007/9781447136750](https://doi.org/10.1007/9781447136750)
- McKee, T. B., Doesken, N. J. and Kleist, J. (1993). The relationship of drought frequency and duration to time scales. In: *Proceedings of the 8th Conference on Applied Climatology*, 17–22 January 1993, Anaheim, California, pp. 179–184. American Meteorological Society.
- Vicente-Serrano, S. M., Begueria, S. and Lopez-Moreno, J. I. (2010). A Multiscalar Drought Index Sensitive to Global Warming: The Standardized Precipitation Evapotranspiration Index. *Journal of Climate* **23**(7), 1696–1718. doi:[10.1175/2009JCLI2909.1](https://doi.org/10.1175/2009JCLI2909.1)

aggregate_climate

Aggregate Climate Data to Coarser Time Steps

Description

Aggregates sub-monthly climate data (e.g., daily) to monthly, seasonal (DJF, MAM, JJA, SON), or annual statistics using a user-supplied aggregation function.

Usage

```
aggregate_climate(x, dates, to = c("monthly", "seasonal", "annual"),
                 FUN = mean)
```

Arguments

x	Numeric vector of climate observations.
dates	Date vector aligned with x.
to	Character. Target time resolution: "monthly", "seasonal", or "annual" (default).
FUN	Function. Aggregation statistic. Default is mean. Use sum for precipitation totals.

Value

A data frame with columns `period` (character labels such as "2000", "2000-01", or "2000-JJA") and `value`.

See Also

[anomaly_baseline](#), [standardize_climate](#).

Examples

```

dates <- seq(as.Date("2000-01-01"), by = "day", length.out = 365 * 5)
temp <- stats::rnorm(length(dates), 15, 5)
ann <- aggregate_climate(temp, dates, to = "annual")
head(ann)
seas <- aggregate_climate(temp, dates, to = "seasonal")
head(seas)

```

anomaly_baseline

Climate Anomaly Relative to a Baseline Period

Description

Computes absolute or standardised anomalies for a climate series relative to a user-defined baseline (reference) period.

Usage

```

anomaly_baseline(x, time_index, baseline_start, baseline_end,
                type = c("absolute", "standardised"))

```

Arguments

x	Numeric vector of climate observations.
time_index	Numeric, integer, or Date vector aligned with x (e.g., years).
baseline_start	Start of the baseline period (same class as time_index).
baseline_end	End of the baseline period.
type	Character. "absolute" (default): $x - \bar{x}_{\text{ref}}$; "standardised": $(x - \bar{x}_{\text{ref}}) / s_{\text{ref}}$.

Value

Numeric vector of anomalies.

See Also

[spatial_anomaly](#), [standardize_climate](#).

Examples

```

years <- 1950:2020
temp <- 14 + 0.02 * (years - 1950) + stats::rnorm(71)
anom <- anomaly_baseline(temp, years, 1961, 1990)
## Mean over baseline period should be zero
cat("Baseline mean:", round(mean(anom[years >= 1961 & years <= 1990]), 10), "\n")

```

`autocorrelation_climate`*Autocorrelation Analysis for Climate Series*

Description

Computes the autocorrelation function (ACF) and partial autocorrelation function (PACF) for a climate series, and performs the Ljung-Box portmanteau test for significant autocorrelation.

Usage

```
autocorrelation_climate(x, max.lag = NULL, plot = TRUE)
```

Arguments

<code>x</code>	Numeric vector (missing values removed).
<code>max.lag</code>	Integer. Maximum lag to compute. Defaults to $\min(30, \text{floor}(n / 4))$.
<code>plot</code>	Logical. If TRUE (default), ACF and PACF plots are produced.

Value

A list with:

`acf` Numeric vector: sample ACF at lags 1 to `max.lag`.
`pacf` Numeric vector: sample PACF at lags 1 to `max.lag`.
`lags` Integer vector: lag indices.
`ljung_box` An "htest" object from [Box.test](#).
`ar1` Numeric: sample ACF at lag 1.

See Also

[mk_test](#).

Examples

```
set.seed(5)
temp <- as.numeric(stats::arima.sim(list(ar = 0.7), n = 100)) + 15
ac <- autocorrelation_climate(temp, plot = FALSE)
cat("AR(1) coefficient:", round(ac$ar1, 3), "\n")
print(ac$ljung_box)
```

`change_point_detection`*Change-Point Detection for Climate Series*

Description

Detects an abrupt shift (change point) in the mean of a climate time series using either Pettitt's non-parametric test or a CUSUM-based approach with bootstrap significance assessment.

Usage

```
change_point_detection(x, method = c("pettitt", "cusum"), alpha = 0.05)
```

Arguments

<code>x</code>	Numeric vector of observations (missing values removed internally).
<code>method</code>	Character. "pettitt" (default) uses Pettitt's rank-based test; "cusum" uses the cumulative sum of deviations from the mean with a bootstrap permutation p-value.
<code>alpha</code>	Numeric in (0, 1). Significance level for declaring a change point. Default is 0.05.

Value

A list containing:

`method` Character: name of the test applied.

`change_point` Integer: index of the detected change point.

`p.value` Numeric: approximate p-value.

`U_stat` Numeric: test statistic (K for Pettitt; max|CUSUM| for CUSUM).

`mean_before` Numeric: mean of observations up to the change point.

`mean_after` Numeric: mean after the change point.

`significant` Logical: whether the shift is significant at alpha.

`n` Integer: number of observations used.

(Pettitt only) `U_series` Integer vector of Pettitt U statistics.

(CUSUM only) `cusum` Numeric vector: cumulative sum series.

References

Pettitt, A.N. (1979). A non-parametric approach to the change-point problem. *Applied Statistics* **28**, 126–135. doi:[10.2307/2346729](https://doi.org/10.2307/2346729)

Page, E.S. (1954). Continuous inspection schemes. *Biometrika* **41**, 100–115.

See Also

[temporal_homogeneity](#) for SNHT-based inhomogeneity detection; [mk_test](#) for gradual trend testing.

Examples

```
## Known shift at observation 30
set.seed(3)
x <- c(stats::rnorm(30, 14, 1), stats::rnorm(30, 16, 1))
cp <- change_point_detection(x, method = "pettitt")
cat("Change point at index:", cp$change_point, "\n")
cat("Mean before:", round(cp$mean_before, 2),
    " / after:", round(cp$mean_after, 2), "\n")

## CUSUM method
change_point_detection(x, method = "cusum")
```

climate_summary

Comprehensive Climate Series Summary

Description

Prints a formatted statistical summary of a climate series including descriptive statistics, Mann-Kendall trend test results, and Sen's slope.

Usage

```
climate_summary(x, dates = NULL, variable_name = "Climate Variable")
```

Arguments

x	Numeric vector of climate observations.
dates	Date vector (currently unused; reserved for future seasonal breakdowns).
variable_name	Character. Label for the variable in the printed output. Default is "Climate Variable".

Value

Invisibly returns a named list with elements mean, sd, mk_tau, mk_p, sens_slope, slope_decade, and trend.

See Also

[mk_test](#), [sens_slope](#).

Examples

```
set.seed(99)
years <- 1970:2020
temp <- 14 + 0.025 * (years - 1970) + stats::rnorm(51, 0, 0.4)
res <- climate_summary(temp, variable_name = "Annual Mean Temperature (C)")
```

climate_test_methods *S3 Methods for "climate_test" Objects*

Description

Print, summary, and plot methods for objects of class "climate_test", which is the base class returned by [mk_test](#) and [seasonal_decompose_climate](#).

Usage

```
## S3 method for class 'climate_test'
print(x, ...)
## S3 method for class 'climate_test'
summary(object, ...)
## S3 method for class 'climate_test'
plot(x, ...)
```

Arguments

x	An object of class "climate_test".
object	An object of class "climate_test" (for summary).
...	Further arguments (currently ignored).

Details

print and summary display a formatted table of the test statistic, p-value, and interpretation.

plot behaviour depends on the subclass:

"mk_test" Two-panel plot: (1) time series with Sen's slope trend line; (2) histogram with mean line.

"climate_decomp" Four-panel plot: original series, trend, seasonal, and remainder components.

Value

print and summary return x invisibly. plot returns x invisibly.

See Also

[mk_test](#), [seasonal_decompose_climate](#).

Examples

```
set.seed(1)
r <- mk_test(1:50 + stats::rnorm(50, 0, 3))
print(r)

plot(r)
```

cluster_climate_zones *K-Means Climate Zone Classification*

Description

Partitions a set of locations into climate zones using k-means clustering on multi-variable climate normals (e.g., mean temperature and total precipitation).

Usage

```
cluster_climate_zones(clim_matrix, k = 5, scale = TRUE, seed = 42)
```

Arguments

clim_matrix	Numeric matrix with rows as locations and columns as climate variables (e.g., mean temperature, total precipitation).
k	Integer. Number of clusters. Default is 5.
scale	Logical. If TRUE (default), each variable is standardised to zero mean and unit variance before clustering.
seed	Integer. Random seed for reproducibility. Default is 42.

Value

A list with:

cluster Integer vector: cluster assignment (1 to k) for each location.

centers Numeric matrix: cluster centres in the (possibly scaled) variable space.

within_ss Numeric: total within-cluster sum of squares.

between_ss Numeric: between-cluster sum of squares.

total_ss Numeric: total sum of squares.

k Integer: number of clusters used.

See Also

[spatial_anomaly](#), [fingerprint_analysis](#).

Examples

```
set.seed(99)
cm <- matrix(c(stats::rnorm(100, 20, 5),
                 stats::rnorm(100, 800, 200)), ncol = 2)
cz <- cluster_climate_zones(cm, k = 3)
table(cz$cluster)
```

cold_spell_detection *Cold Spell Detection*

Description

Identifies cold spell events as periods during which daily minimum temperature falls below a threshold for a minimum number of consecutive days. This is the low-temperature counterpart of [heat_wave_detection](#).

Usage

```
cold_spell_detection(tmin, dates, threshold = "p10", min_days = 3)
```

Arguments

tmin	Numeric vector of daily minimum temperatures (°C).
dates	Date vector aligned with tmin.
threshold	Numeric or character percentile string (e.g., "p10" or "p05"). Default is "p10".
min_days	Integer. Minimum consecutive days. Default is 3.

Value

A data.frame with columns start_date, end_date, duration, min_temp, mean_temp, and intensity (mean deficit below the threshold).

See Also

[heat_wave_detection](#), [frost_days](#).

Examples

```
set.seed(9)
dates <- seq(as.Date("2000-01-01"), by = "day", length.out = 365 * 5)
doy <- as.integer(format(dates, "%j"))
tmin <- 5 - 12 * sin(2 * pi * doym / 365) +
        stats::rnorm(length(dates), 0, 2)
cs <- cold_spell_detection(tmin, dates, threshold = "p10")
cat("Cold spell events:", nrow(cs), "\n")
```

detection_attribution *Climate Change Detection and Attribution Test*

Description

Tests whether an observed climate change can be statistically distinguished from natural internal variability using a signal-to-noise ratio approach based on projections onto a forced signal.

Usage

```
detection_attribution(observed, natural_ensemble,
                      forced_signal = NULL, conf.level = 0.95)
```

Arguments

observed	Numeric vector of observed anomalies (e.g., annual mean temperature anomalies relative to a pre-industrial baseline).
natural_ensemble	Numeric matrix where each column is a control-run (natural internal variability) time series of the same length as observed.
forced_signal	Optional numeric vector (same length as observed) representing the model-simulated forced signal. If NULL, a linear ramp is used.
conf.level	Numeric. Confidence level. Default is 0.95.

Value

A list with:

detected	Logical: whether the signal is detected at conf.level.
z_score	Numeric: standardised score relative to natural variability.
p.value	Numeric: two-tailed p-value.
attribution_fraction	Numeric: fraction of observed variance explained by the forced signal.
noise_sd	Numeric: mean standard deviation of the natural ensemble.
projection_observed	Numeric: Pearson correlation of observed with the forced signal.
projection_natural	Numeric vector: correlations for each ensemble member.
conf.level	Numeric: confidence level used.
method	Character: "Optimal Fingerprint Detection".

References

- Hasselmann, K. (1979). On the signal-to-noise problem in atmospheric response studies. In: Shaw, D. B. (ed.) *Meteorology Over the Tropical Oceans*, pp. 251–259. Royal Meteorological Society, Bracknell.
- Allen, M. R. and Tett, S. F. B. (1999). Checking for model consistency in optimal fingerprinting. *Climate Dynamics* **15**(6), 419–434. doi:10.1007/s003820050291
- Hegerl, G. C. and Zwiers, F. (2011). Use of models in detection and attribution of climate change. *Wiley Interdisciplinary Reviews: Climate Change* **2**(4), 570–591. doi:10.1002/wcc.121

See Also

[fingerprint_analysis](#), [optimal_fingerprint](#).

Examples

```
set.seed(10)
obs <- cumsum(stats::rnorm(50, 0.025, 0.15))
nat <- matrix(stats::rnorm(50 * 20, 0, 0.5), ncol = 20)
sig <- seq(0, 1.2, length.out = 50)
da <- detection_attribution(obs, nat, sig)
cat("Signal detected:", da$detected, "\n")
cat("Attribution fraction:", round(da$attribution_fraction, 2), "\n")
```

diurnal_temp_range	<i>Diurnal Temperature Range</i>
--------------------	----------------------------------

Description

Computes the mean Diurnal Temperature Range (DTR = Tmax - Tmin) aggregated by year or calendar month. DTR is widely used as an indicator of cloud cover, land use change, and climate variability.

Usage

```
diurnal_temp_range(tmax, tmin, dates, by = c("year", "month"))
```

Arguments

tmax	Numeric vector of daily maximum temperature (°C).
tmin	Numeric vector of daily minimum temperature.
dates	Date vector aligned with tmax and tmin.
by	Character. "year" (default) or "month".

Value

Named numeric vector of mean DTR values.

See Also

[growing_degree_days](#), [frost_days](#).

Examples

```
dates <- seq(as.Date("1980-01-01"), by = "day", length.out = 365 * 10)
tmax <- rep(25, length(dates))
tmin <- rep(12, length(dates))
dtr <- diurnal_temp_range(tmax, tmin, dates)
cat("Mean DTR:", round(mean(dtr), 1), "\u00b0C\n")
```

drought_spell

Drought Spell Detection from Standardised Indices

Description

Identifies drought episodes as periods during which a standardised drought index (SPI, SPEI, or any continuous series) remains below a threshold for a minimum number of consecutive time steps. The severity integrates the area between the index series and the threshold.

Usage

```
drought_spell(index_series, dates, threshold = -1.0, min_duration = 2)
```

Arguments

<code>index_series</code>	Numeric vector (e.g., SPI or SPEI values).
<code>dates</code>	Date or integer vector aligned with <code>index_series</code> .
<code>threshold</code>	Numeric. Drought onset threshold. Default is <code>-1.0</code> (moderate drought in the McKee SPI classification).
<code>min_duration</code>	Integer. Minimum duration in time steps. Default is 2.

Value

A data.frame with columns `start`, `end`, `duration`, `min_index`, `mean_index`, and `severity` (positive number: sum of `threshold - index` over the event). Returns an empty data frame if no events are found.

References

McKee, T.B., Doesken, N.J. and Kleist, J. (1993). The relationship of drought frequency and duration to time scales. *Proceedings of the 8th Conference on Applied Climatology*, 179–184.

See Also

[spi](#), [spei](#).

Examples

```
set.seed(7)
spi_vals <- stats::rnorm(360)
dates_m <- seq(as.Date("1990-01-01"), by = "month",
               length.out = 360)
droughts <- drought_spell(spi_vals, dates_m, threshold = -1)
cat("Drought events:", nrow(droughts), "\n")
```

elevation_lapse_rate *Environmental Lapse Rate Estimation*

Description

Estimates the temperature lapse rate (change in temperature per 1000 m elevation gain) from station temperature and elevation data via ordinary least squares regression. Useful for statistical down-scaling and data quality assessment.

Usage

```
elevation_lapse_rate(temp, elevation)
```

Arguments

temp Numeric vector of air temperature observations ($^{\circ}\text{C}$).

elevation Numeric vector of station elevations (metres above sea level), aligned with temp.

Value

A list with:

lapse_rate_per_m Numeric: OLS slope ($^{\circ}\text{C m}^{-1}$).

lapse_rate_per_1000m Numeric: lapse rate per 1000 m ($^{\circ}\text{C km}^{-1}$).

intercept Numeric: OLS intercept.

r_squared Numeric: coefficient of determination.

data Data frame with columns elevation, temp, and fitted.

Examples

```
elev <- seq(100, 3000, by = 100)
temp <- 25 - 0.0065 * elev + stats::rnorm(30, 0, 0.5)
lr <- elevation_lapse_rate(temp, elev)
cat("Lapse rate:", round(lr$lapse_rate_per_1000m, 2),
    "\u00b0C / 1000 m\n")
cat("R-squared:", round(lr$r_squared, 3), "\n")
```

extreme_value_index *Extreme Value Index (Hill Estimator)*

Description

Estimates the tail index of a heavy-tailed climate variable (e.g., wind speed, precipitation intensity) using the Hill estimator, with a stability plot for choosing the order statistic k .

Usage

```
extreme_value_index(x, k = NULL)
```

Arguments

x Numeric vector of positive observations.
k Integer. Number of upper order statistics to use. Defaults to $\max(2, \text{floor}(\sqrt{n}))$.

Value

A list with:

hill_index Numeric: Hill estimate of the tail index at k .
xi_estimate Numeric: same as **hill_index** (GEV shape parameterisation).
k_used Integer: order statistic used.
n Integer: sample size.
hill_plot Data frame with columns **k** and **hill**: Hill estimates across a range of k for stability assessment.

References

Hill, B.M. (1975). A simple general approach to inference about the tail of a distribution. *Annals of Statistics* **3**, 1163–1174. doi:10.1214/aos/1176343247

See Also

[fit_gev](#), [peaks_over_threshold](#).

Examples

```
set.seed(9)
ws <- abs(stats::rnorm(500, 10, 4))^1.5
ev <- extreme_value_index(ws)
cat("Hill tail index:", round(ev$hill_index, 3), "\n")

plot(ev$hill_plot$k, ev$hill_plot$hill, type = "l",
     xlab = "k", ylab = "Hill estimate")
```

fill_gaps_climate *Gap-Filling for Climate Series*

Description

Fills missing values in a climate series using linear interpolation, monthly climatological means, or regression against a nearby reference station.

Usage

```
fill_gaps_climate(x, method = c("linear", "climatology", "reference"),
                 dates = NULL, ref = NULL)
```

Arguments

x	Numeric vector containing NA values to be filled.
method	Character. "linear" (default): piecewise linear interpolation; "climatology": replace with the calendar-month mean; "reference": simple linear regression on a parallel station series.
dates	Date vector, required for method = "climatology".
ref	Numeric reference series of the same length as x, required for method = "reference".

Value

Numeric vector with missing values filled.

See Also

[homogenize_series](#), [standardize_climate](#).

Examples

```
x <- c(10, NA, NA, 13, 14, NA, 16)
xf <- fill_gaps_climate(x)
print(xf)
```

fingerprint_analysis *EOF-Based Spatial Fingerprint Analysis*

Description

Extracts the leading Empirical Orthogonal Functions (EOFs) and corresponding Principal Components (PCs) from a climate anomaly field as spatial fingerprints of climate change or variability modes.

Usage

```
fingerprint_analysis(data, n_eof = 3)
```

Arguments

data	Numeric matrix with rows as time steps and columns as spatial locations (grid cells or stations). Column means are removed internally.
n_eof	Integer. Number of leading EOFs to extract. Default is 3.

Value

A list with:

eof Numeric matrix ($p \times n_eof$): spatial EOF patterns.

pc Numeric matrix ($n \times n_eof$): principal component time series.

var_explained Numeric vector: fraction of variance explained by each EOF.

cumvar Numeric vector: cumulative explained variance.

n_eof Integer: number of EOFs extracted.

References

Lorenz, E. N. (1956). *Empirical Orthogonal Functions and Statistical Weather Prediction*. Scientific Report No. 1, Statistical Forecasting Project. Massachusetts Institute of Technology, Department of Meteorology, Cambridge, Massachusetts.

Von Storch, H. and Zwiers, F. W. (1999). *Statistical Analysis in Climate Research*. Cambridge University Press, Cambridge. doi:[10.1017/CBO9780511612336](https://doi.org/10.1017/CBO9780511612336)

See Also

[detection_attribution](#), [optimal_fingerprint](#).

Examples

```

set.seed(5)
mat <- matrix(stats::rnorm(50 * 100), nrow = 50)
## Add a forced spatially coherent signal
mat[, 1:30] <- mat[, 1:30] +
  outer(seq(0, 1, length.out = 50), rep(1, 30))
fp <- fingerprint_analysis(mat, n_eof = 3)
cat("Variance explained by EOF1:",
    round(fp$var_explained[1] * 100, 1), "%\n")

plot(fp$pc[, 1], type = "l", main = "Leading PC",
     xlab = "Year", ylab = "PC score")

```

fit_gev

*Fit Generalised Extreme Value Distribution***Description**

Fits the three-parameter Generalised Extreme Value (GEV) distribution to a vector of block maxima (e.g., annual maximum daily temperature or precipitation) via maximum likelihood estimation (MLE).

Usage

```
fit_gev(x, conf.level = 0.95)
```

Arguments

x	Numeric vector of block maxima (at least 10 non-missing values).
conf.level	Numeric in (0, 1). Confidence level for parameter confidence intervals derived by the delta method. Default is 0.95.

Details

The GEV cumulative distribution function is

$$F(x; \mu, \sigma, \xi) = \exp\left\{-\left[1 + \xi\left(\frac{x - \mu}{\sigma}\right)\right]^{-1/\xi}\right\}$$

for $1 + \xi(x - \mu)/\sigma > 0$. When $\xi \rightarrow 0$ this reduces to the Gumbel distribution.

Starting values for optimisation are obtained by Gumbel method-of-moments. The Hessian at the MLE is used to construct the parameter covariance matrix and delta-method confidence intervals.

Value

An object of class `c("gev_fit", "climate_test")` (a list) with:

`mu` Numeric: location parameter μ .
`sigma` Numeric: scale parameter $\sigma > 0$.
`xi` Numeric: shape parameter ξ .
`se` Numeric vector of length 3: standard errors.
`ci` 3-by-2 matrix: lower and upper confidence limits.
`loglik` Numeric: log-likelihood at the MLE.
`aic` Numeric: Akaike information criterion.
`bic` Numeric: Bayesian information criterion.
`n` Integer: number of block maxima used.
`data` Numeric: the input data.
`cov_mat` 3-by-3 numeric matrix: parameter covariance.
`method` Character: "GEV maximum likelihood".
`conf.level` Numeric: the confidence level used.

Calling `print()` on this object shows a formatted parameter table.

References

Coles, S. (2001). *An Introduction to Statistical Modeling of Extreme Values*. Springer, London.
[doi:10.1007/9781447136750](https://doi.org/10.1007/9781447136750)

Jenkinson, A.F. (1955). The frequency distribution of the annual maximum (or minimum) values of meteorological elements. *Quarterly Journal of the Royal Meteorological Society* **81**, 158–171.

See Also

[return_period](#) for return-level estimation; [peaks_over_threshold](#) for the POT alternative; [rgev_sim](#) for simulating GEV random variates.

Examples

```
set.seed(42)
ann_max <- rgev_sim(50, mu = 35, sigma = 4, xi = 0.1)
gev <- fit_gev(ann_max)
print(gev)
```

frost_days	<i>Frost Day Count</i>
------------	------------------------

Description

Counts the number of frost days (days on which daily minimum temperature falls below 0°C) aggregated by year or calendar month.

Usage

```
frost_days(tmin, dates, by = c("year", "month"))
```

Arguments

tmin	Numeric vector of daily minimum temperatures (°C).
dates	Date vector aligned with tmin.
by	Character. Aggregate by "year" (default) or "month".

Value

Named integer vector of frost day counts.

See Also

[growing_degree_days](#), [cold_spell_detection](#).

Examples

```
dates <- seq(as.Date("2000-01-01"), by = "day", length.out = 365)
tmin <- rep(5, 365)
tmin[1:30] <- -2 ## 30 frost days in January
fd <- frost_days(tmin, dates, by = "year")
cat("Annual frost days:", fd["2000"], "\n")
```

growing_degree_days	<i>Growing Degree Days</i>
---------------------	----------------------------

Description

Computes daily Growing Degree Days (GDD) as the amount by which the daily mean temperature exceeds a base threshold, with an optional upper temperature cap.

Usage

```
growing_degree_days(tmax, tmin, base_temp = 10,
                    upper_temp = Inf, dates = NULL,
                    cumulative = FALSE)
```

Arguments

tmax	Numeric vector of daily maximum temperature (°C).
tmin	Numeric vector of daily minimum temperature.
base_temp	Numeric. Base temperature threshold. Default is 10 °C.
upper_temp	Numeric. Upper temperature cap. Default is Inf (no cap).
dates	Date vector. Currently unused but retained for future aggregation features.
cumulative	Logical. If TRUE, returns cumulative GDD. Default is FALSE.

Value

Numeric vector of daily GDD (or cumulative GDD if `cumulative = TRUE`).

See Also

[frost_days](#), [diurnal_temp_range](#).

Examples

```
n <- 365
tmax <- rep(25, n)
tmin <- rep(15, n)
## Mean = 20, base = 10 -> 10 GDD per day
gdd <- growing_degree_days(tmax, tmin, base_temp = 10)
cat("Annual GDD:", sum(gdd), "\n") ## should be 3650
```

heat_index

Heat Index (Apparent Temperature)

Description

Computes the Heat Index (apparent temperature, also called the “feels like” temperature) from air temperature and relative humidity using the Rothfus regression equation (National Weather Service).

Usage

```
heat_index(temp, rh, unit = "C")
```

Arguments

temp	Numeric vector of air temperature.
rh	Numeric vector of relative humidity (percent, 0–100).
unit	Character. "C" (Celsius, default) or "F" (Fahrenheit).

Value

Numeric vector of heat index values in the same unit as temp.

References

Rothfus, L. P. (1990). *The Heat Index Equation*. Technical Attachment SR/SSD 90-23. National Weather Service Southern Region, Fort Worth, Texas.

Steadman, R. G. (1979). The assessment of sultriness. Part I: A temperature-humidity index based on human physiology and clothing science. *Journal of Applied Meteorology* **18**(7), 861–873. doi:10.1175/15200450(1979)018<0861:TAOSPI>2.0.CO;2

See Also

[wind_chill](#), [heat_wave_detection](#).

Examples

```
## At 35 C, RH 80%
hi <- heat_index(temp = 35, rh = 80)
print(paste("Heat index:", round(hi, 1), "deg C"))
## Higher humidity feels hotter
heat_index(35, 90) > heat_index(35, 40)
```

heat_wave_detection *Heat Wave Detection*

Description

Identifies heat wave events as periods during which daily maximum temperature exceeds a threshold for a minimum number of consecutive days.

Usage

```
heat_wave_detection(tmax, dates, threshold = "p90", min_days = 3)
```

Arguments

tmax	Numeric vector of daily maximum temperatures (°C).
dates	Date vector aligned with tmax.
threshold	Numeric or character. A fixed temperature threshold (e.g., 35), or a percentile string such as "p90" or "p95". The percentile is computed over the full record.
min_days	Integer. Minimum number of consecutive days above the threshold required to qualify as a heat wave. Default is 3.

Value

A data.frame with one row per event and columns:

start_date Date: first day of the event.

end_date Date: last day.

duration Integer: length in days.

peak_temp Numeric: maximum temperature during the event.

mean_temp Numeric: mean temperature.

intensity Numeric: mean excess above the threshold (°C-days).

If no events are found, an empty data.frame with the above columns is returned invisibly.

See Also

[cold_spell_detection](#), [heat_index](#).

Examples

```
set.seed(6)
dates <- seq(as.Date("1990-01-01"), by = "day", length.out = 365 * 10)
doy <- as.integer(format(dates, "%j"))
tmax <- 25 + 10 * sin(2 * pi * doy / 365) +
  stats::rnorm(length(dates), 0, 3)
hw <- heat_wave_detection(tmax, dates, threshold = "p90", min_days = 3)
cat("Heat wave events detected:", nrow(hw), "\n")
head(hw)
```

homogenize_series

Homogenise a Climate Series Using SNHT

Description

Applies a mean-shift correction to a climate series based on the break point detected by the Standard Normal Homogeneity Test (SNHT). The segment before the detected break is shifted upward or downward so that its mean equals the mean of the segment after the break.

Usage

```
homogenize_series(x, alpha = 0.05)
```

Arguments

x Numeric vector. The climate series to homogenise.

alpha Numeric. Significance level for the SNHT. Default is 0.05. If no significant break is found, x is returned unchanged with a message.

Value

Numeric vector: the homogenised series (same length as x).

See Also

[temporal_homogeneity](#), [fill_gaps_climate](#).

Examples

```
set.seed(20)
x_inhom <- c(stats::rnorm(40, 0, 1), stats::rnorm(40, 1.5, 1))
x_hom <- homogenize_series(x_inhom)
cat("Mean before correction:", round(mean(x_inhom[1:40]), 2), "\n")
cat("Mean after correction:", round(mean(x_hom[1:40]), 2), "\n")
```

hot_cold_spots

Local Spatial Hot-Spot and Cold-Spot Detection (Getis-Ord G_i^)*

Description

Identifies statistically significant spatial clusters of high values (hot spots) or low values (cold spots) in a climate field using the Getis-Ord G_i^* local statistic.

Usage

```
hot_cold_spots(x, coords, dist_threshold = NULL, alpha = 0.05)
```

Arguments

x Numeric vector of climate values at n locations.
 $coords$ Matrix or data frame of coordinates (two columns).
 $dist_threshold$ Numeric. Neighbourhood radius. Defaults to the median pairwise distance.
 $alpha$ Numeric. Significance level. Default is 0.05.

Details

G_i^* includes the focal location itself in the sum, making it suited to detecting local hot and cold spots rather than spatial outliers. Z scores are derived under the assumption of normality.

Value

A data frame with one row per location and columns:

$index$ Integer: location index.
 x Numeric: original climate value.
 lon, lat Numeric: coordinates.

gi_star Numeric: G_i^* Z score.
 p.value Numeric: two-tailed p-value.
 n_neighbours Numeric: sum of spatial weights (number of neighbours including self).
 classification Character: "hot spot", "cold spot", or "not significant".

References

Getis, A. and Ord, J.K. (1992). The analysis of spatial association by use of distance statistics. *Geographical Analysis* **24**, 189–206. doi:10.1111/j.15384632.1992.tb00261.x

See Also

[morans_i](#), [spatial_anomaly](#).

Examples

```
set.seed(3)
n      <- 50
coords <- data.frame(x = stats::runif(n, 0, 10),
                    y = stats::runif(n, 0, 10))
vals   <- ifelse(coords$x > 7,
                 stats::rnorm(n, 30, 2),
                 stats::rnorm(n, 15, 2))
hs <- hot_cold_spots(vals, coords, dist_threshold = 2)
table(hs$classification)
```

mk_test

Mann-Kendall Trend Test

Description

Performs the non-parametric Mann-Kendall test for a monotonic trend in a climate time series. Missing values are silently removed. An optional AR(1) pre-whitening step (Yue and Wang 2002) removes serial autocorrelation before the test statistic is computed.

Usage

```
mk_test(x, prewhiten = FALSE, conf.level = 0.95,
        alternative = c("two.sided", "greater", "less"))
```

Arguments

x	Numeric vector of observations ordered in time.
prewhiten	Logical. If TRUE, AR(1) pre-whitening is applied to remove autocorrelation before computing the test statistic (modified Mann-Kendall test). Default is FALSE.
conf.level	Numeric in (0, 1). Confidence level used to determine the trend direction label. Default is 0.95.
alternative	Character string specifying the alternative hypothesis. One of "two.sided" (default), "greater", or "less".

Details

The Mann-Kendall statistic S is the sum of signs of all pairwise differences $\text{sgn}(x_j - x_i)$ for $j > i$. Under the null hypothesis of no trend, S is asymptotically normal with mean zero and variance adjusted for tied values.

The standardised statistic Z is obtained by continuity correction (± 1 adjustment to S).

When `prewhiten = TRUE` and the sample AR(1) coefficient exceeds 0.05, the series is filtered as $x'_t = x_t - \hat{\rho}x_{t-1}$ before computing S .

Value

An object of class `c("mk_test", "climate_test")` (a list) containing:

`statistic` Named numeric: the standardised Z statistic.

`p.value` Numeric: p-value for the chosen alternative.

`tau` Numeric: Kendall's tau.

`S` Integer: Mann-Kendall score.

`var.S` Numeric: variance of S adjusted for ties.

`n` Integer: number of non-missing observations used.

`trend` Character: "increasing", "decreasing", or "no trend".

`alternative` Character: the alternative hypothesis.

`conf.level` Numeric: the confidence level used.

`prewhiten` Logical: whether pre-whitening was applied.

`data` Numeric: the original input vector (including NAs).

`method` Character: description of the method.

References

Mann, H.B. (1945). Nonparametric tests against trend. *Econometrica* **13**, 245–259. doi:10.2307/1907187

Kendall, M.G. (1975). *Rank Correlation Methods*. Griffin, London.

Yue, S. and Wang, C.Y. (2002). Applicability of prewhitening to eliminate the influence of serial correlation on the Mann-Kendall test. *Water Resources Research* **38**, 1068. doi:10.1029/2001WR000861

See Also

[sens_slope](#) for the slope magnitude; [trend_significance](#) for simultaneous testing of multiple stations.

Examples

```
## Basic usage on a warming temperature series
set.seed(42)
years <- 1971:2020
temp <- 14.0 + 0.025 * (years - 1971) + stats::rnorm(50, 0, 0.4)
result <- mk_test(temp)
print(result)

## One-sided test
mk_test(temp, alternative = "greater")

## Pre-whitened variant for autocorrelated series
ar_temp <- as.numeric(stats::arima.sim(list(ar = 0.6), n = 60)) +
  seq(0, 3, length.out = 60) + 14
mk_test(ar_temp, prewhiten = TRUE)

## Plot the result (time series + distribution)
plot(result)
```

morans_i

Moran's I Test for Spatial Autocorrelation

Description

Computes the global Moran's I statistic for spatial autocorrelation in a climate field (e.g., temperature or precipitation anomalies at observation stations), with both analytical and permutation-based p-values.

Usage

```
morans_i(x, coords, dist_threshold = NULL, n_perm = 999)
```

Arguments

x	Numeric vector of climate values at n locations.
coords	Matrix or data frame with two columns giving the coordinates (longitude/latitude or x/y) for each location.
dist_threshold	Numeric. Maximum distance defining spatial neighbours. If NULL (default), the median pairwise distance is used.
n_perm	Integer. Number of random permutations for the permutation-based p-value. Default is 999.

Details

Moran's I is defined as

$$I = \frac{n}{S_0} \frac{\mathbf{z}^\top W \mathbf{z}}{\mathbf{z}^\top \mathbf{z}}$$

where \mathbf{z} are mean-centred values, W is the row-standardised binary spatial weight matrix, and S_0 is the sum of all weights.

Under the randomisation assumption the expected value is $E[I] = -1/(n - 1)$. The analytical variance uses the formulas of Cliff and Ord (1981). The permutation p-value randomly reassigns values to locations n_{perm} times.

Value

A list with:

I Numeric: Moran's I statistic.
 expected Numeric: expected value $-1/(n - 1)$.
 variance Numeric: analytical variance.
 Z Numeric: standardised Z score.
 p.value Numeric: analytical two-tailed p-value.
 p.perm Numeric: permutation p-value.
 n_perm Integer: number of permutations used.
 dist_threshold Numeric: neighbourhood radius used.
 interpretation Character: plain-English summary.

References

Moran, P.A.P. (1950). Notes on continuous stochastic phenomena. *Biometrika* **37**, 17–23. doi:10.2307/2332142

Cliff, A.D. and Ord, J.K. (1981). *Spatial Processes: Models and Applications*. Pion, London.

See Also

[hot_cold_spots](#) for local spatial clustering; [spatial_trend_field](#) for per-location trend analysis.

Examples

```
set.seed(7)
n <- 30
coords <- data.frame(lon = stats::runif(n, -10, 10),
                     lat = stats::runif(n, 40, 60))
## Temperature decreasing with latitude -> positive autocorrelation
x <- 25 - 0.3 * coords$lat + stats::rnorm(n, 0, 1)
mi <- morans_i(x, coords, n_perm = 199)
cat("Moran's I:", round(mi$I, 3), " p =", round(mi$p.value, 4), "\n")
cat(mi$interpretation, "\n")
```

optimal_fingerprint *Optimal Fingerprint Regression*

Description

Estimates scaling factors for anthropogenic (ANT) and natural (NAT) climate signals by regressing observed climate changes onto model-simulated response patterns using Generalised Least Squares (GLS), with an optional noise covariance matrix estimated from control runs.

Usage

```
optimal_fingerprint(obs, all_forcing, nat_forcing = NULL,
                    noise_cov = NULL)
```

Arguments

obs	Numeric vector of observed climate changes.
all_forcing	Numeric vector of model-simulated changes under all forcings (ANT + NAT), same length as obs.
nat_forcing	Optional numeric vector of model-simulated changes under natural forcing only. If supplied, the ANT signal is computed as all_forcing - nat_forcing.
noise_cov	Optional positive-definite covariance matrix of internal variability noise ($n \times n$). Defaults to the identity matrix.

Value

A list with:

beta_all Numeric: scaling factor for the ALL-forcing signal (or ANT if nat_forcing is supplied).

beta_nat Numeric: scaling factor for the NAT signal (NA if nat_forcing is NULL).

residual_variance Numeric: residual variance per degree of freedom.

scaling_factors Numeric vector or matrix: all estimated scaling factors.

method Character: "Optimal Fingerprint (GLS)".

References

Allen, M.R. and Tett, S.F.B. (1999). Checking for model consistency in optimal fingerprinting. *Climate Dynamics* **15**, 419–434. doi:10.1007/s003820050291

See Also

[detection_attribution](#), [fingerprint_analysis](#).

Examples

```
set.seed(42)
obs_c <- cumsum(stats::rnorm(50, 0.03, 0.1))
all_c <- cumsum(stats::rnorm(50, 0.025, 0.05)) +
  cumsum(stats::rnorm(50, 0.01, 0.05))
nat_c <- cumsum(stats::rnorm(50, 0, 0.1))
ofp <- optimal_fingerprint(obs_c, all_c, nat_c)
cat("ANT scaling factor:", round(ofp$beta_all, 2), "\n")
cat("NAT scaling factor:", round(ofp$beta_nat, 2), "\n")
```

pdsi_simple

Simplified Palmer Drought Severity Index

Description

Computes a simplified, self-calibrated Palmer Drought Severity Index (PDSI) from monthly mean temperature and precipitation using the Thornthwaite potential evapotranspiration method.

Usage

```
pdsi_simple(temp, precip, lat = 40, awc = 150)
```

Arguments

temp	Numeric vector of monthly mean temperature (°C).
precip	Numeric vector of monthly precipitation (mm), same length as temp.
lat	Numeric. Latitude in decimal degrees for day-length correction. Default is 40.
awc	Numeric. Available water capacity of the soil (mm). Default is 150.

Value

Numeric vector of simplified PDSI values. Negative values indicate drought; positive values indicate wet conditions.

References

Palmer, W. C. (1965). *Meteorological Drought*. Research Paper No. 45. US Weather Bureau, Washington, D.C.

Thornthwaite, C. W. (1948). An approach toward a rational classification of climate. *Geographical Review* **38**(1), 55–94. doi:10.2307/210739

See Also

[spi](#), [spei](#).

Examples

```

set.seed(1)
n <- 360
doy <- rep(1:12, 30)
temp <- 10 + 12 * sin(pi * doym / 6) + stats::rnorm(n, 0, 1)
pr <- pmax(0, 50 + 20 * cos(pi * doym / 6) + stats::rnorm(n, 0, 15))
pdsi_vals <- pdsi_simple(temp, pr, lat = 40)
plot(pdsi_vals, type = "l", ylab = "PDSI",
     xlab = "Month", main = "Simplified PDSI")
abline(h = 0, lty = 2)

```

peaks_over_threshold *Peaks-Over-Threshold Analysis*

Description

Selects independent peak exceedances above a user-defined threshold, fits the Generalised Pareto Distribution (GPD) to the excesses, and estimates return levels.

Usage

```

peaks_over_threshold(x, threshold, min_sep = 3,
                    return_periods = c(10, 50, 100),
                    n_years = NULL)

```

Arguments

x	Numeric vector of observations (e.g., daily or sub-daily data).
threshold	Numeric. Exceedance threshold.
min_sep	Integer. Minimum separation (in observations) between retained peaks to ensure independence. Default is 3.
return_periods	Numeric vector of return periods (years). Default is c(10, 50, 100).
n_years	Numeric. Length of the record in years. If NULL (default), estimated as length(x) / 365.25.

Value

A list with:

sigma	Numeric: GPD scale parameter.
xi	Numeric: GPD shape parameter.
threshold	Numeric: the threshold used.
n_excess	Integer: number of peaks retained.
lambda	Numeric: average number of exceedances per year.
n_years	Numeric: record length in years.
return_levels	Data frame with columns T and level.
excess	Numeric vector: retained peak excesses above the threshold.

References

Davison, A.C. and Smith, R.L. (1990). Models for exceedances over high thresholds (with discussion). *Journal of the Royal Statistical Society B* **52**, 393–442. doi:10.1111/j.25176161.1990.tb01796.x

See Also

[fit_gev](#), [extreme_value_index](#).

Examples

```
set.seed(2)
daily_precip <- stats::rexp(365 * 30, rate = 1 / 5)
pot <- peaks_over_threshold(daily_precip, threshold = 20,
                           n_years = 30,
                           return_periods = c(10, 50, 100))
print(pot$return_levels)
```

print.gev_fit

Print Method for GEV Fit Objects

Description

Prints a formatted parameter table for objects of class "gev_fit" returned by [fit_gev](#).

Usage

```
## S3 method for class 'gev_fit'
print(x, ...)
```

Arguments

x An object of class "gev_fit".
... Further arguments (currently ignored).

Value

x, invisibly.

See Also

[fit_gev](#).

Examples

```
set.seed(2)
gev <- fit_gev(rgev_sim(40, mu = 30, sigma = 5, xi = 0.1))
print(gev)
```

return_period	<i>Return Periods and Return Levels</i>
---------------	---

Description

Computes return levels (values expected to be exceeded on average once every T years) from a fitted GEV object with delta-method confidence intervals, or empirically using the Gringorten plotting position.

Usage

```
return_period(fit, return_periods = c(2, 5, 10, 25, 50, 100, 200),
              conf.level = 0.95)
```

Arguments

fit	Either a <code>gev_fit</code> object returned by <code>fit_gev</code> , or a numeric vector of annual maxima for the empirical method.
return_periods	Numeric vector of return periods (years). Default is <code>c(2, 5, 10, 25, 50, 100, 200)</code> .
conf.level	Numeric. Confidence level for the delta-method intervals. Default is 0.95. Ignored for the empirical method.

Value

A data.frame with columns T (return period), level (return level), lower, and upper (confidence bounds; NA for empirical method).

References

Gringorten, I.I. (1963). A plotting rule for extreme probability paper. *Journal of Geophysical Research* **68**, 813–814. doi:10.1029/JZ068i003p00813

See Also

`fit_gev`, `peaks_over_threshold`.

Examples

```
set.seed(1)
am <- rgev_sim(50, mu = 30, sigma = 5, xi = 0.05)
gev <- fit_gev(am)
rp <- return_period(gev, c(2, 10, 50, 100))
print(rp)
```

rgev_sim	<i>Simulate GEV Random Variates</i>
----------	-------------------------------------

Description

Generates random variates from the Generalised Extreme Value distribution using the probability-integral transform. Primarily intended for testing, simulation studies, and vignette examples.

Usage

```
rgev_sim(n, mu = 0, sigma = 1, xi = 0)
```

Arguments

n	Integer. Number of observations.
mu	Numeric. Location parameter. Default is 0.
sigma	Numeric. Scale parameter (must be positive). Default is 1.
xi	Numeric. Shape parameter. Default is 0 (Gumbel).

Value

Numeric vector of length n.

See Also

[fit_gev](#).

Examples

```
set.seed(1)
x <- rgev_sim(100, mu = 30, sigma = 5, xi = 0.1)
hist(x, main = "GEV sample", col = "lightblue")
```

rolling_trend	<i>Rolling-Window Trend Analysis</i>
---------------	--------------------------------------

Description

Applies Sen's slope estimator over a moving window to identify periods of accelerating or decelerating warming (or other climate trends).

Usage

```
rolling_trend(x, window = 20, step = 1)
```

Arguments

x	Numeric vector of observations.
window	Integer. Window length in observations. Default is 20.
step	Integer. Step size between consecutive windows. Default is 1.

Value

A data.frame with one row per window position and columns:

start_index Integer: first index of the window.

end_index Integer: last index of the window.

mid_index Numeric: mid-point of the window.

slope Numeric: Sen's slope within the window.

slope_decade Numeric: slope scaled to change per decade.

See Also

[sens_slope](#), [mk_test](#).

Examples

```
set.seed(1)
years <- 1950:2020
temp <- 14 + 0.015 * (years - 1950) + stats::rnorm(71, 0, 0.5)
rt <- rolling_trend(temp, window = 20, step = 2)
plot(rt$mid_index, rt$slope_decade, type = "l",
      xlab = "Mid-window index", ylab = "Trend per decade")
abline(h = 0, lty = 2)
```

seasonal_decompose_climate

Seasonal Climate Decomposition

Description

Decomposes a regularly-spaced climate series into trend, seasonal, and irregular (remainder) components using STL (Seasonal and Trend decomposition using Loess) or classical additive decomposition.

Usage

```
seasonal_decompose_climate(x, frequency = 12,
                           method = c("stl", "classical"),
                           s.window = "periodic")
```

Arguments

x	Numeric vector. Must be a complete, regularly-spaced series. Any NA values are linearly interpolated before decomposition.
frequency	Integer. Number of observations per cycle: 12 for monthly data, 4 for quarterly, etc. Default is 12.
method	Character. "stl" (default, robust) or "classical" (additive).
s.window	Passed to stl : "periodic" (default) or an odd integer giving the loess window for the seasonal extraction.

Value

An object of class `c("climate_decomp", "climate_test")` (a list) with elements:

method Character: decomposition method used.

frequency Integer: cycle length.

x Numeric: original (possibly interpolated) series.

trend Numeric: trend component.

seasonal Numeric: seasonal component.

remainder Numeric: irregular / residual component.

Calling `plot()` on this object produces a four-panel time series display (original, trend, seasonal, remainder).

See Also

[rolling_trend](#) to analyse trend acceleration; [mk_test](#) to test for monotonic trend in the extracted trend component.

Examples

```
## 30 years of synthetic monthly temperature
set.seed(7)
n <- 360
t_idx <- seq_along(numeric(n))
temp <- 15 + 0.002 * t_idx + 8 * sin(2 * pi * t_idx / 12) +
  stats::rnorm(n, 0, 0.5)
dc <- seasonal_decompose_climate(temp, frequency = 12)

## Inspect components

plot(dc)
```

sens_slope	<i>Sen's Slope Estimator</i>
------------	------------------------------

Description

Computes Sen's slope — the median of all pairwise slopes — as a robust, non-parametric estimator of linear trend magnitude for a climate time series. An optional confidence interval is derived via a normal approximation on the rank of the sorted pairwise slopes.

Usage

```
sens_slope(x = NULL, y, conf.level = 0.95)
```

Arguments

x	Optional numeric vector of time indices (e.g., years). If NULL, integer indices 1:length(y) are used.
y	Numeric vector of climate observations. Pairs with missing values in either x or y are removed.
conf.level	Numeric in (0, 1). Confidence level for the slope confidence interval. Default is 0.95.

Details

The estimator computes all $n(n - 1)/2$ pairwise slopes $Q_{ij} = (x_j - x_i)/(t_j - t_i)$ for $j > i$ and takes their median. The intercept is computed as $\hat{\beta}_0 = \text{median}(y) - \hat{\beta}_1 \text{median}(x)$.

The confidence interval uses a normal approximation based on the Mann-Kendall variance to identify the lower and upper rank positions in the sorted slope vector (Sen 1968).

The convenience field `slope_decade` multiplies the slope by 10, giving the trend per decade — a standard reporting unit in climate science.

Value

A list with elements:

`slope` Numeric: Sen's slope estimate.

`intercept` Numeric: corresponding intercept.

`slope_ci` Named numeric vector `c(lower, upper)`: confidence interval for the slope.

`slope_decade` Numeric: slope scaled to change per decade (`slope * 10`).

`n` Integer: number of complete pairs used.

`conf.level` Numeric: the confidence level used.

`x` Numeric: time indices used (after NA removal).

`y` Numeric: observations used (after NA removal).

`fitted` Numeric: fitted values $\hat{\beta}_0 + \hat{\beta}_1 x$.

References

Sen, P.K. (1968). Estimates of the regression coefficient based on Kendall's tau. *Journal of the American Statistical Association* **63**, 1379–1389. doi:10.2307/2285891

See Also

[mk_test](#) for the associated significance test; [rolling_trend](#) for moving-window application.

Examples

```
## Annual mean temperature 1970-2020
set.seed(1)
years <- 1970:2020
temp <- 14.0 + 0.03 * (years - 1970) + stats::rnorm(51, 0, 0.4)
ss <- sens_slope(years, temp)
cat("Warming rate:", round(ss$slope_decade, 3), "\u00b0C per decade\n")
cat("95% CI:      [", round(ss$slope_ci["lower"], 3), ", ",
    round(ss$slope_ci["upper"], 3), "]\n")

## Without explicit time index (uses 1:n)
sens_slope(y = temp)$slope_decade
```

spatial_anomaly

Spatial Climate Anomaly Field

Description

Computes standardised anomalies for each grid cell or station column relative to a user-defined climatological baseline period.

Usage

```
spatial_anomaly(data, time_index, baseline_start, baseline_end)
```

Arguments

data Numeric matrix with rows as time steps and columns as locations.

time_index Numeric or integer vector (e.g., years) aligned with the rows of data.

baseline_start Start of the baseline period (same class as `time_index`).

baseline_end End of the baseline period.

Value

Numeric matrix of standardised anomalies with the same dimensions as `data`. Columns with zero baseline standard deviation are returned as NA.

See Also

[anomaly_baseline](#) for a single time series.

Examples

```
set.seed(4)
mat <- matrix(stats::rnorm(50 * 20, 15, 3), nrow = 50)
idx <- 1971:2020
anm <- spatial_anomaly(mat, idx, 1971, 2000)
cat("Dimensions:", dim(anm), "\n")
```

spatial_interpolate *Spatial Interpolation for Climate Data*

Description

Interpolates scattered station observations onto a regular grid using Inverse Distance Weighting (IDW) or a two-dimensional loess spline.

Usage

```
spatial_interpolate(obs_coords, obs_values, grid_coords,
                    method = c("idw", "spline"), power = 2)
```

Arguments

obs_coords	Numeric matrix ($n \times 2$) of observation coordinates (longitude/latitude or x/y).
obs_values	Numeric vector of length n : observed climate values.
grid_coords	Numeric matrix ($m \times 2$) of target grid coordinates.
method	Character. "idw" (default) or "spline" (2-D loess).
power	Numeric. IDW distance-decay exponent. Default is 2.

Value

Numeric vector of length m : interpolated values at grid_coords.

References

Shepard, D. (1968). A two-dimensional interpolation function for irregularly-spaced data. *Proceedings of the 23rd ACM National Conference*, 517–524. doi:10.1145/800186.810616

See Also

[elevation_lapse_rate](#), [spatial_anomaly](#).

Examples

```

set.seed(5)
obs <- matrix(stats::runif(40), ncol = 2)
vals <- sin(obs[, 1] * 3) + cos(obs[, 2] * 3)
grd <- as.matrix(expand.grid(x = seq(0.1, 0.9, 0.1),
                             y = seq(0.1, 0.9, 0.1)))
pred <- spatial_interpolate(obs, vals, grd, method = "idw")
cat("Interpolated", length(pred), "grid points\n")

```

spatial_trend_field *Spatial Field of Climate Trends*

Description

Applies the Mann-Kendall test and Sen's slope independently to each column (grid cell or station) of a space-time matrix, producing a map of trend magnitudes and significance.

Usage

```
spatial_trend_field(data_3d, ...)
```

Arguments

data_3d	Numeric matrix with rows as time steps and columns as locations, <i>or</i> a three-dimensional array with dimensions [lon, lat, time].
...	Additional arguments passed to mk_test .

Value

A data.frame with one row per location and columns loc, slope, slope_decade, tau, p.value, and trend.

See Also

[mk_test](#), [trend_significance](#).

Examples

```

set.seed(11)
mat <- matrix(stats::rnorm(50 * 100), nrow = 50)
## Impose warming in second half of locations
mat[, 51:100] <- mat[, 51:100] +
  outer(seq_len(50), rep(0.03, 50))
sf <- spatial_trend_field(mat)
cat("Significant cells:", sum(sf$p.value < 0.05, na.rm = TRUE), "\n")

```


Examples

```
set.seed(5)
n <- 240
pr <- stats::rgamma(n, 5, 0.04)
tmin <- abs(stats::rnorm(n, 8, 3)) + 2
tmax <- tmin + stats::runif(n, 6, 12)
sp6 <- spei(precip = pr, tmin = tmin, tmax = tmax, lat = 45, scale = 6)
cat("SPEI-6 SD:", round(stats::sd(sp6, na.rm = TRUE), 3), "\n")
```

 spi

Standardised Precipitation Index (SPI)

Description

Computes the Standardised Precipitation Index at any accumulation time scale from a monthly precipitation series. A two-parameter gamma distribution is fitted to the positive accumulated values over a reference period, and probabilities are transformed to Z scores.

Usage

```
spi(precip, scale = 3, dates = NULL,
    ref_start = NULL, ref_end = NULL)
```

Arguments

precip	Numeric vector of monthly precipitation totals (mm), ordered chronologically.
scale	Integer. Accumulation period in months (e.g., 1, 3, 6, 12). Default is 3.
dates	Date or character vector aligned with precip. If NULL, the series is assumed to start in January.
ref_start	Integer year. Start of the reference period for distribution fitting. Defaults to the full record.
ref_end	Integer year. End of the reference period. Defaults to the full record.

Details

An n-month accumulation is computed by summing n consecutive monthly values, introducing n - 1 leading NAs. The proportion of zero-precipitation months is modelled explicitly as a point mass at zero; the gamma CDF is used for positive values. SPI values are obtained by inverting the normal distribution.

Typical SPI classifications (McKee et al. 1993):

≥ 2.0	Extremely wet
1.5 to 1.99	Very wet
1.0 to 1.49	Moderately wet
-0.99 to 0.99	Near normal

-1.0 to -1.49	Moderately dry
-1.5 to -1.99	Severely dry
≤ -2.0	Extremely dry

Value

Numeric vector of SPI values (same length as precip). The first scale - 1 elements are NA.

References

McKee, T.B., Doesken, N.J. and Kleist, J. (1993). The relationship of drought frequency and duration to time scales. *Proceedings of the 8th Conference on Applied Climatology*, 179–184.

See Also

[spei](#), [drought_spell](#).

Examples

```
set.seed(3)
precip <- stats::rgamma(360, shape = 2, rate = 0.05)
spi3 <- spi(precip, scale = 3)
## Distribution should be approximately standard normal
cat("Mean:", round(mean(spi3, na.rm = TRUE), 3),
    "SD:", round(stats::sd(spi3, na.rm = TRUE), 3), "\n")

plot(spi3, type = "h",
     col = ifelse(spi3 >= 0, "steelblue", "firebrick"),
     xlab = "Month", ylab = "SPI-3")
abline(h = c(-1, 1), lty = 2)
```

standardize_climate *Standardise a Climate Variable*

Description

Applies Z-score standardisation to a climate variable, optionally performing the standardisation separately within each calendar month to remove seasonality before further analysis (e.g., trend testing).

Usage

```
standardize_climate(x, by_month = FALSE, dates = NULL)
```

Arguments

x	Numeric vector.
by_month	Logical. If TRUE and dates is supplied, standardise within each calendar month. Default is FALSE.
dates	Date vector aligned with x. Required when by_month = TRUE.

Value

Numeric vector of standardised values.

See Also

[anomaly_baseline](#).

Examples

```
x <- stats::rnorm(120, 20, 5)
z <- standardize_climate(x)
cat("Mean:", round(mean(z), 10), " SD:", round(stats::sd(z), 6), "\n")
```

temporal_homogeneity *Standard Normal Homogeneity Test (SNHT)*

Description

Applies Alexandersson's Standard Normal Homogeneity Test to detect a single inhomogeneity (e.g., a station relocation or instrument change) in a climate record.

Usage

```
temporal_homogeneity(x, alpha = 0.05)
```

Arguments

x	Numeric vector of climate observations (at least 10 non-missing).
alpha	Numeric. Significance level. Default is 0.05.

Value

A list with elements:

method Character: "Standard Normal Homogeneity Test (SNHT)".

T0 Numeric: maximum SNHT statistic.

break_index Integer: index of the detected break.

critical Numeric: critical value from Alexandersson (1986).

significant Logical: TRUE if $T_0 > \text{critical}$.

T_series Numeric vector: SNHT statistic at each candidate break position.

n Integer: sample size.

References

Alexandersson, H. (1986). A homogeneity test applied to precipitation data. *International Journal of Climatology* **6**, 661–675. doi:10.1002/joc.3370060607

See Also

[homogenize_series](#) to apply the detected break correction; [change_point_detection](#) for the Pettitt / CUSUM alternatives.

Examples

```
set.seed(10)
x <- c(stats::rnorm(40, 0, 1), stats::rnorm(40, 0.8, 1))
snht <- temporal_homogeneity(x)
cat("Break at index:", snht$break_index,
    " Significant:", snht$significant, "\n")
```

trend_significance *Multiple-Station Trend Significance with FDR/Bonferroni Correction*

Description

Runs the Mann-Kendall test on each column of a climate data matrix (stations or grid cells) simultaneously and applies a false discovery rate (FDR) or Bonferroni correction to adjust for multiple comparisons.

Usage

```
trend_significance(data, correction = c("fdr", "bonferroni"), alpha = 0.05)
```

Arguments

data	A numeric matrix or data frame where each column is a station or grid-cell time series (rows = time steps). A vector is coerced to a single-column matrix.
correction	Character. "fdr" (Benjamini-Hochberg, default) or "bonferroni".
alpha	Numeric. Family-wise significance level. Default is 0.05.

Value

A data.frame with one row per station / grid cell and columns:

station Integer: column index.
tau Numeric: Kendall's tau.
p.raw Numeric: unadjusted p-value.
p.adjusted Numeric: adjusted p-value.
significant Logical: whether the adjusted p-value is below alpha.
trend Character: "increasing", "decreasing", or "no trend".

References

Benjamini, Y. and Hochberg, Y. (1995). Controlling the false discovery rate. *Journal of the Royal Statistical Society B* **57**, 289–300. doi:10.1111/j.25176161.1995.tb02031.x

See Also

[mk_test](#), [spatial_trend_field](#).

Examples

```
set.seed(42)
mat <- matrix(stats::rnorm(50 * 20), nrow = 50)
## Impose a trend in first 5 stations
mat[, 1:5] <- mat[, 1:5] + outer(seq_len(50), rep(0.05, 5))
ts_result <- trend_significance(mat)
table(ts_result$trend)
```

wind_chill

Wind Chill Temperature

Description

Computes the Wind Chill Temperature using the 2001 North American Joint Action Group for Temperature Indices (JAG/TI) formula, valid for air temperatures at or below 10 °C and wind speeds above 4.8 km/h.

Usage

```
wind_chill(temp, wind)
```

Arguments

temp	Numeric. Air temperature (°C).
wind	Numeric. Wind speed (km/h).

Value

Numeric: Wind Chill Temperature (°C).

References

Environment Canada and US National Weather Service (2001). *Report on Wind Chill Temperature and Extreme Heat Indices: Evaluation and Improvement Projects*. Office of the Federal Coordinator for Meteorological Services and Supporting Research (OFCM), Washington, D.C. Publication FCM-R19-2003.

Siple, P. A. and Passel, C. F. (1945). Measurements of dry atmospheric cooling in subfreezing temperatures. *Proceedings of the American Philosophical Society* **89**(1), 177–199.

See Also

[heat_index](#).

Examples

```
wind_chill(temp = -10, wind = 30)
## Should be substantially colder than -10
wind_chill(-10, 30) < -10
```

Index

- * **cluster**
 - cluster_climate_zones, 10
- * **distribution**
 - extreme_value_index, 16
 - fit_gev, 19
 - peaks_over_threshold, 32
 - return_period, 34
 - rgev_sim, 35
- * **htest**
 - change_point_detection, 7
 - detection_attribution, 12
 - mk_test, 26
 - temporal_homogeneity, 45
 - trend_significance, 46
- * **methods**
 - climate_test_methods, 9
 - print_gev_fit, 33
- * **multivariate**
 - fingerprint_analysis, 18
- * **nonparametric**
 - mk_test, 26
 - sens_slope, 38
- * **package**
 - climatestatsr-package, 3
- * **regression**
 - elevation_lapse_rate, 15
 - optimal_fingerprint, 30
 - sens_slope, 38
- * **spatial**
 - cluster_climate_zones, 10
 - elevation_lapse_rate, 15
 - hot_cold_spots, 25
 - morans_i, 28
 - spatial_anomaly, 39
 - spatial_interpolate, 40
 - spatial_trend_field, 41
- * **ts**
 - aggregate_climate, 4
 - anomaly_baseline, 5
 - autocorrelation_climate, 6
 - climate_summary, 8
 - cold_spell_detection, 11
 - diurnal_temp_range, 13
 - drought_spell, 14
 - fill_gaps_climate, 17
 - frost_days, 21
 - growing_degree_days, 21
 - heat_index, 22
 - heat_wave_detection, 23
 - homogenize_series, 24
 - pdsi_simple, 31
 - rolling_trend, 35
 - seasonal_decompose_climate, 36
 - spei, 42
 - spi, 43
 - standardize_climate, 44
 - wind_chill, 47
- aggregate_climate, 3, 4
- anomaly_baseline, 3, 4, 5, 40, 45
- autocorrelation_climate, 3, 6
- Box.test, 6
- change_point_detection, 3, 7, 46
- climate_summary, 3, 8
- climate_test_methods, 9
- climatestatsr (climatestatsr-package), 3
- climatestatsr-package, 3
- cluster_climate_zones, 3, 10
- cold_spell_detection, 3, 11, 21, 24
- detection_attribution, 3, 12, 18, 30
- diurnal_temp_range, 3, 13, 22
- drought_spell, 3, 14, 42, 44
- elevation_lapse_rate, 3, 15, 40
- extreme_value_index, 3, 16, 33
- fill_gaps_climate, 3, 17, 25

fingerprint_analysis, [3](#), [10](#), [13](#), [18](#), [30](#)
fit_gev, [3](#), [16](#), [19](#), [33–35](#)
frost_days, [3](#), [11](#), [14](#), [21](#), [22](#)

growing_degree_days, [3](#), [14](#), [21](#), [21](#)

heat_index, [3](#), [22](#), [24](#), [48](#)
heat_wave_detection, [3](#), [11](#), [23](#), [23](#)
homogenize_series, [3](#), [17](#), [24](#), [46](#)
hot_cold_spots, [3](#), [25](#), [29](#)

mk_test, [3](#), [6](#), [8](#), [9](#), [26](#), [36](#), [37](#), [39](#), [41](#), [47](#)
morans_i, [3](#), [26](#), [28](#)

optimal_fingerprint, [3](#), [13](#), [18](#), [30](#)

pdsi_simple, [3](#), [31](#)
peaks_over_threshold, [3](#), [16](#), [20](#), [32](#), [34](#)
plot.climate_test
 (climate_test_methods), [9](#)
print.climate_test
 (climate_test_methods), [9](#)
print.gev_fit, [33](#)

return_period, [3](#), [20](#), [34](#)
rgev_sim, [20](#), [35](#)
rolling_trend, [3](#), [35](#), [37](#), [39](#)

seasonal_decompose_climate, [3](#), [9](#), [36](#)
sens_slope, [3](#), [8](#), [27](#), [36](#), [38](#)
spatial_anomaly, [3](#), [5](#), [10](#), [26](#), [39](#), [40](#)
spatial_interpolate, [3](#), [40](#)
spatial_trend_field, [3](#), [29](#), [41](#), [47](#)
spei, [3](#), [14](#), [31](#), [42](#), [44](#)
spi, [3](#), [14](#), [31](#), [42](#), [43](#)
standardize_climate, [3–5](#), [17](#), [44](#)
stl, [37](#)
summary.climate_test
 (climate_test_methods), [9](#)

temporal_homogeneity, [3](#), [8](#), [25](#), [45](#)
trend_significance, [3](#), [27](#), [41](#), [46](#)

wind_chill, [3](#), [23](#), [47](#)