

Package ‘Rfactor’

September 12, 2026

Title Rainfall Erosivity Calculations from Precipitation Records

Version 0.2.0

Description Calculates rainfall erosivity from timestamped precipitation records.

The package identifies rainfall events, calculates continuous rolling rainfall intensities, rainfall kinetic energy, and event EI30 erosivity, aggregates contributing event erosivity to monthly and yearly totals, and calculates multi-year mean monthly and annual rainfall erosivity. Storm separation, event omission criteria, intensity durations, and rainfall kinetic-energy equations are configurable. Rainfall kinetic energy can be calculated using the formulations of Brown and Foster (1987) [<doi:10.13031/2013.31957>](https://doi.org/10.13031/2013.31957), McGregor et al. (1995) [<doi:10.13031/2013.27921>](https://doi.org/10.13031/2013.27921), and Laws and Parsons (1943) [<doi:10.1029/TR024i002p00452>](https://doi.org/10.1029/TR024i002p00452).

License GPL (>= 3)

URL <https://github.com/adrianira/Rfactor>

BugReports <https://github.com/adrianira/Rfactor/issues>

Encoding UTF-8

Depends R (>= 4.0.0)

Imports clock, data.table

Suggests knitr, rmarkdown, testthat (>= 3.0.0)

VignetteBuilder knitr

Config/testthat/edition 3

Config/roxygen2/version 8.1.0

NeedsCompilation no

Author Adrian Iraşoc [aut, cre, cph] (ORCID:
 [<https://orcid.org/0000-0002-9496-5588>](https://orcid.org/0000-0002-9496-5588))

Maintainer Adrian Iraşoc <adrian.irasoc@gmail.com>

Repository CRAN

Date/Publication 2026-09-12 13:50:09 UTC

Contents

rf_calculate_ei30	2
rf_calculate_energy	5
rf_calculate_mean_rfactor	7
rf_calculate_rfactor	9
rf_identify_storms	11
rf_max_intensity	13
rf_read_rainfall	15
rf_settings	17
rf_validate_rainfall	19
Index	22

rf_calculate_ei30	<i>Calculate event rainfall erosivity and EI30</i>
-------------------	--

Description

Calculates rainfall amount, maximum rolling rainfall intensities, kinetic energy, EI30 erosivity, and erosive-event classification for rainfall events produced by [rf_identify_storms\(\)](#).

Usage

```
rf_calculate_ei30(storms, settings = NULL, interval_min = NULL)
```

Arguments

storms	A rainfall-event data frame produced by rf_identify_storms() .
settings	An rf_settings object created by rf_settings() . If NULL, the settings stored in storms are used. If no stored settings are available, default settings from rf_settings() are used.
interval_min	Positive finite number giving the temporal resolution of the rainfall-event grid, in minutes. If NULL, the interval_min attribute stored by rf_identify_storms() is used. The interval must divide 30 minutes exactly, and all requested intensity durations must be exact multiples of the interval.

Value

A data frame of class `rf_events` with one row per rainfall event. It contains:

- storm_id: event identifier;
- event_start: timestamp of the first event interval;
- event_end: timestamp of the final event interval;
- duration_min: represented event duration, in minutes;
- precip_mm: total event precipitation, in mm;
- energy_mj_ha: total rainfall kinetic energy, in MJ/ha;

- ei30: event EI30 erosivity, in MJ mm/(ha h);
- omit_precip_condition: result of the precipitation omission condition, or NA when disabled;
- omit_intensity_condition: result of the intensity omission condition, or NA when disabled;
- omitted: whether the event satisfies the configured omission rule;
- erosive: logical inverse of omitted;
- intensity columns named i<duration>_mm_h for each duration requested in settings\$intensity_durations_min.

The returned object also stores settings and interval_min attributes.

If storms contains no events, an empty rf_events object with the corresponding output structure is returned.

EI30 erosivity

Event rainfall erosivity is calculated as

$$EI_{30} = E \times I_{30}$$

where E is total event rainfall kinetic energy in MJ/ha and I30 is the maximum continuous 30-minute rainfall intensity in mm/h. EI30 therefore has units of MJ mm/(ha h).

Maximum rainfall intensities are calculated with `rf_max_intensity()` using continuous rolling windows rather than fixed clock-period blocks.

Temporal resolution

Calculation of EI30 requires rainfall data capable of representing an exact 30-minute intensity window. interval_min must therefore divide 30 minutes exactly.

Every duration requested in settings\$intensity_durations_min must also be an exact multiple of interval_min.

Fixed-interval calculations have been validated against RIST 3.99.10 using 1-, 5-, 10-, 15-, and 30-minute rainfall data. Rainfall data with intervals greater than 30 minutes cannot provide a genuine I30 and therefore cannot be used for EI30 calculation.

Temporal aggregation can change maximum intensity, kinetic energy, and EI30 even when total rainfall is unchanged. Results calculated from different source resolutions should therefore not be expected to be identical.

Event omission

Event omission is controlled by `rf_settings()`.

The precipitation criterion is enabled with omit_precip and is satisfied when total event precipitation is strictly less than omit_precip_below_mm.

The intensity criterion is enabled with omit_intensity and is satisfied when the selected maximum rolling intensity is strictly less than omit_intensity_below_mm_h. The intensity duration is selected with omit_intensity_duration_min.

When both criteria are enabled:

- omit_logic = "all" omits an event only when both conditions are satisfied;
- omit_logic = "any" omits an event when either condition is satisfied.

Exact threshold values do not satisfy a criterion because the comparisons are strict (<).

If only one omission criterion is enabled, only that criterion is evaluated. If both omit_precip and omit_intensity are FALSE, no events are omitted. This corresponds to including all identified rainfall events.

For disabled criteria, the corresponding omit_precip_condition or omit_intensity_condition column is returned as NA, indicating that the condition was not evaluated.

Single-record events

RIST 3.99.10 was observed during fixed-interval validation to assign zero kinetic energy and zero EI30 to events represented by only one rainfall record.

Set single_record_energy = "rist_zero" in `rf_settings()` to reproduce this behaviour. With the default single_record_energy = "calculate", the selected kinetic-energy equation is applied normally.

Examples

```
rainfall <- data.frame(
  datetime = as.POSIXct(
    "2025-01-01 00:00:00",
    tz = "UTC"
  ) +
  seq(
    from = 0,
    by = 60,
    length.out = 30
  ),
  precip_mm = rep(1, 30)
)

storms <- rf_identify_storms(
  rainfall,
  interval_min = 1
)

events <- rf_calculate_ei30(
  storms
)

events
```

rf_calculate_energy *Calculate rainfall kinetic energy*

Description

Calculates total rainfall kinetic energy for a rainfall event from fixed-interval precipitation depths.

Usage

```
rf_calculate_energy(
  precip_mm,
  interval_min = 1,
  energy_equation = "brown_foster_1987"
)
```

Arguments

precip_mm	Numeric vector containing precipitation depth, in mm, for each consecutive fixed observation interval. Values must be finite, non-missing, and non-negative.
interval_min	Positive finite number giving the duration of each observation interval, in minutes. Default is 1.
energy_equation	Character string selecting the kinetic-energy equation. One of: • "brown_foster_1987"; • "mcgregor_1995"; • "laws_parsons_1943". Default is "brown_foster_1987".

Details

Rainfall intensity for each observation interval is calculated as

$$i_k = P_k \frac{60}{\Delta t}$$

where P_k is interval precipitation in mm and Δt is the interval duration in minutes.

Three kinetic-energy equations are available.

Value

Total rainfall kinetic energy for the supplied event, in MJ/ha.

Brown and Foster (1987)

$$e = 0.29[1 - 0.72 \exp(-0.05i)]$$

McGregor et al. (1995)

$$e = 0.29[1 - 0.72 \exp(-0.082i)]$$

The coefficient 0.082 is used. Although the RIST 3.99.10 graphical interface displays this coefficient as 0.08, dedicated validation showed that RIST energy calculations agree with the 0.082 formulation.

Laws and Parsons (1943)

$$e = 0.119 + 0.0873 \log_{10}(i)$$

For compatibility with the behaviour validated against RIST 3.99.10, this equation is applied directly to positive rainfall intensities without an upper intensity cap. Validation included rainfall intensity of 120 mm/h.

The logarithmic equation is undefined at zero intensity. Zero-rainfall intervals are therefore assigned zero unit energy and contribute zero kinetic energy.

Event energy

For all equations, e is unit rainfall energy in MJ/ha/mm and i is rainfall intensity in mm/h. Total rainfall energy is

$$E = \sum_k e_k P_k$$

where E is total event energy in MJ/ha.

precip_mm is assumed to represent consecutive observations at the fixed interval specified by interval_min. This function does not contain timestamps and therefore does not detect temporal gaps. In the normal Rfactor workflow, `rf_identify_storms()` prepares the regular within-event rainfall sequence before energy is calculated.

Examples

```
# Thirty minutes at 1 mm/min:
rf_calculate_energy(
  precip_mm = rep(1, 30),
  interval_min = 1
)

# The same rainfall using the McGregor equation
rf_calculate_energy(
  precip_mm = rep(1, 30),
  interval_min = 1,
  energy_equation = "mcgregor_1995"
)

# Ten-minute fixed-interval rainfall data
rf_calculate_energy(
```

```
precip_mm = c(2, 5, 3),
interval_min = 10,
energy_equation = "laws_parsons_1943"
)
```

rf_calculate_mean_rfactor

Calculate multi-year mean rainfall erosivity

Description

Calculates multi-year mean rainfall-runoff erosivity from monthly or yearly R-factor values, typically produced by `rf_calculate_rfactor()`.

Usage

```
rf_calculate_mean_rfactor(rfactor, period = c("yearly", "monthly"))
```

Arguments

rfactor	A data frame containing monthly or yearly R-factor values. The input must contain a numeric R column and a year column. For period = "monthly", a month column is also required.
period	Character string specifying the temporal scale of the input. One of "yearly" or "monthly".

Details

The function calculates the arithmetic multi-year mean of the available R-factor values.

Missing R values are excluded from the calculation. Genuine zero values are retained because $R = 0$ represents a valid period with no contributing erosive events.

For yearly input, the function calculates

$$\bar{R} = \frac{1}{n} \sum_{y=1}^n R_y$$

where R_y is the available rainfall erosivity for year y and n is the number of non-missing yearly values.

For monthly input, a separate multi-year mean is calculated for each calendar month:

$$\bar{R}_m = \frac{1}{n_m} \sum_{y=1}^{n_m} R_{y,m}$$

where $R_{y,m}$ is the available rainfall erosivity for month m in year y , and n_m is the number of non-missing values available for that calendar month.

Missing years or month-year combinations are not generated or imputed. An NA value is not included in either the mean or `n_years`.

A value of $R = 0$, however, is included in the mean and in `n_years`.

The function does not determine whether the underlying rainfall record is climatologically complete. An available monthly or yearly R-factor value is treated as valid unless the user removes it or marks it as NA. Assessment of record completeness and representativeness remains the responsibility of the user.

The mean annual R-factor is calculated directly from yearly R-factor values. It is not calculated by summing the twelve multi-year monthly means. This distinction is important when data availability differs among calendar months.

Value

For `period = "yearly"`, a data frame with:

mean_R Multi-year mean annual rainfall erosivity.

n_years Number of non-missing yearly R-factor values used.

For `period = "monthly"`, a data frame with:

month Calendar month as an integer from 1 to 12.

mean_R Multi-year mean rainfall erosivity for that month.

n_years Number of non-missing R-factor values used for that month.

Calendar months that are completely absent from the supplied input are not generated.

Examples

```
yearly <- data.frame(
  year = 2020:2024,
  R = c(
    820.4,
    910.2,
    NA,
    760.1,
    0
  )
)

rf_calculate_mean_rfactor(
  yearly,
  period = "yearly"
)

monthly <- data.frame(
  year = c(
    2020,
    2021,
    2022,
    2020,
```

```

      2021,
      2022
    ),
    month = c(
      1,
      1,
      1,
      2,
      2,
      2
    ),
    R = c(
      30,
      40,
      NA,
      50,
      0,
      70
    )
  )
)

rf_calculate_mean_rfactor(
  monthly,
  period = "monthly"
)

```

`rf_calculate_rfactor` *Aggregate event rainfall erosivity by calendar period*

Description

Aggregates event EI30 erosivity into monthly or yearly calendar totals.

Usage

```
rf_calculate_rfactor(events, period = c("yearly", "monthly"))
```

Arguments

<code>events</code>	A data frame containing at least: • <code>event_start</code>: event starting time as POSIXct; • <code>ei30</code>: event EI30 erosivity in MJ mm/(ha h); • <code>erosive</code>: logical event classification. Normally this is an <code>rf_events</code> object returned by rf_calculate_ei30().
<code>period</code>	Calendar aggregation period. Either "yearly" or "monthly". Default is "yearly".

Details

Only events with `erosive == TRUE` and a non-missing `ei30` value contribute to the erosivity total. Non-erosive or omitted events contribute zero.

Calendar periods represented by at least one event are retained even when none of their events contribute erosivity. Such periods are returned with `R = 0` and `n_events = 0`.

Calendar months or years that are completely absent from the supplied event data are not created, filled, or imputed.

Value

A data frame ordered chronologically.

For `period = "yearly"`, the columns are:

- `year`: calendar year;
- `R`: sum of contributing event `EI30` values for that year, in MJ mm/(ha h);
- `n_events`: number of erosive events with non-missing `EI30` that contribute to the total.

For `period = "monthly"`, the result additionally contains:

- `month`: calendar month as an integer from 1 to 12.

An empty event table returns an empty data frame with the corresponding yearly or monthly structure.

Interpretation of R

The `R` column returned by this function is the sum of contributing event `EI30` values for the requested calendar period:

$$R_p = \sum_{j \in p} EI_{30,j}$$

where the sum includes only events classified as erosive and having a non-missing `EI30` value.

A yearly value returned by this function represents the rainfall erosivity calculated from the available events for that calendar year. It is not, by itself, the climatological long-term mean annual rainfall-runoff erosivity factor (R-factor) used by USLE or RUSLE.

Calculation of a representative long-term R-factor requires an appropriate multi-year rainfall record and assessment of record completeness and representativeness outside this aggregation function.

Calendar assignment

Each event is assigned to the calendar month or year containing its `event_start` timestamp. Consequently, an event that crosses a month or year boundary is assigned in full to the period in which it begins.

Examples

```
events <- data.frame(
  event_start = as.POSIXct(
    c(
      "2025-05-10 12:00:00",
      "2025-05-20 15:00:00",
      "2025-06-05 09:00:00"
    ),
    tz = "UTC"
  ),
  ei30 = c(
    100,
    25,
    40
  ),
  erosive = c(
    TRUE,
    FALSE,
    FALSE
  )
)

# May is represented with R = 100.
# June is represented with R = 0 because it contains rainfall
# events but none classified as erosive.
rf_calculate_rfactor(
  events,
  period = "monthly"
)

rf_calculate_rfactor(
  events,
  period = "yearly"
)
```

rf_identify_storms	<i>Identify independent rainfall events</i>
--------------------	---

Description

Splits a timestamped rainfall series into independent rainfall events using the elapsed time between consecutive observations containing positive rainfall.

Usage

```
rf_identify_storms(data, settings = rf_settings(), interval_min = NULL)
```

Arguments

data	A data frame containing datetime and precip_mm. The data are validated with <code>rf_validate_rainfall()</code> .
settings	An <code>rf_settings</code> object created by <code>rf_settings()</code> . Default settings are used when this argument is omitted.
interval_min	Positive finite number giving the base temporal resolution of the rainfall record, in minutes. If NULL, the <code>expected_interval_min</code> attribute stored by <code>rf_read_rainfall()</code> is used. If that metadata are not available, <code>interval_min</code> must be supplied explicitly.

Details

The input rainfall series does not need to contain every expected time step. Sparse rainfall records and temporal gaps are allowed.

In the fixed-interval storm-separation behaviour validated against RIST 3.99.10, a new event begins when the elapsed time between consecutive positive-rainfall observations is greater than `storm_break_hours`.

With the default setting of 6 hours:

- a gap of 6 hours or less remains within the same event;
- a gap greater than 6 hours starts a new event.

Any positive rainfall observation resets the storm-break clock, regardless of its precipitation amount.

The `storm_break_precip_mm` setting in `rf_settings()` is retained as RIST configuration meta-data but does not affect event separation in the validated fixed-interval algorithm used by Rfactor. Dedicated experiments with RIST 3.99.10 using 1-minute fixed-interval rainfall data produced identical event grouping when this setting was varied from 0 to 10 mm.

Value

A data frame of class `rf_storms` containing:

- `storm_id`: sequential event identifier;
- `datetime`: timestamps on the reconstructed within-event grid;
- `precip_mm`: precipitation amount at each grid position, in mm.

The returned object also stores the calculation settings and temporal resolution in the attributes `settings` and `interval_min`.

An entirely dry rainfall series returns an empty `rf_storms` object.

Reconstruction of sparse events

After an event has been identified, a regular time grid is reconstructed from its first through its last positive-rainfall observation using `interval_min`.

Observations supplied by the user, including explicitly recorded zeros, are placed on this grid. Expected time positions absent from the supplied record are assigned a precipitation amount of zero for the calculation.

This computational reconstruction preserves elapsed time for rolling-intensity and energy calculations. An inserted zero does not assert that the corresponding interval was observed and dry, and the procedure does not estimate missing precipitation.

Regular grids are reconstructed independently for each event. Time steps lying between separate events are not generated.

Examples

```
rainfall <- data.frame(
  datetime = as.POSIXct(
    c(
      "2025-01-01 00:00:00",
      "2025-01-01 03:00:00",
      "2025-01-01 07:00:00"
    ),
    tz = "UTC"
  ),
  precip_mm = c(5, 0.1, 5)
)

# The positive rainfall at 03:00 resets the six-hour break clock,
# so all three observations belong to one event.
storms <- rf_identify_storms(
  rainfall,
  interval_min = 1
)

storms
```

rf_max_intensity	<i>Calculate maximum rolling rainfall intensity</i>
------------------	---

Description

Calculates the maximum rainfall intensity over a continuous rolling time window.

Usage

```
rf_max_intensity(precip_mm, duration_min = 30, interval_min = 1)
```

Arguments

precip_mm	Numeric vector of precipitation depths, in mm, for consecutive regular rainfall intervals. Values must be finite, non-missing, and non-negative.
duration_min	Positive finite number giving the duration of the rolling intensity window, in minutes. It must be an exact multiple of interval_min. Default is 30.
interval_min	Positive finite number giving the temporal resolution represented by each element of precip_mm, in minutes. Default is 1.

Details

The calculation evaluates all possible consecutive windows represented by the rainfall vector; windows are not restricted to fixed clock boundaries. For example, with 1-minute rainfall data and `duration_min = 30`, the maximum depth occurring in any consecutive 30-minute window is used.

The requested duration must be an exact multiple of the rainfall interval. Thus, 10-minute rainfall data can be used directly for 10-, 20-, 30-, or 60-minute intensities, but not for a true 5- or 15-minute intensity.

`precip_mm` is assumed to represent consecutive observations on a regular time grid with spacing `interval_min`. The function does not contain timestamps and therefore cannot detect gaps. In the normal Rfactor workflow, `rf_identify_storms()` reconstructs the regular within-event grid before rainfall intensities are calculated.

Partial rolling windows are permitted. When fewer observations are available than required for a complete window, the available rainfall depth is divided by the full requested duration. For example, 10 mm falling during a 10-minute event gives a 30-minute intensity of 20 mm/h:

$$I_{30} = 10 \times 60/30 = 20 \text{ mm/h}$$

This treatment reproduces the behaviour observed during validation against RIST 3.99.10 fixed-interval rainfall input.

Value

The maximum rolling rainfall intensity, in mm/h.

Examples

```
# Thirty minutes of rainfall at 1 mm/min:
# 30 mm in 30 minutes = 60 mm/h
rf_max_intensity(
  precip_mm = rep(1, 30),
  duration_min = 30,
  interval_min = 1
)

# Ten-minute rainfall data can be used directly for I30.
rf_max_intensity(
  precip_mm = c(1, 2, 3),
  duration_min = 30,
  interval_min = 10
)

# A rainfall event shorter than the requested window uses a
# partial window. Ten mm in a 10-minute event gives I30 = 20 mm/h.
rf_max_intensity(
  precip_mm = rep(1, 10),
  duration_min = 30,
  interval_min = 1
)
```

rf_read_rainfall	<i>Read timestamped rainfall data</i>
------------------	---------------------------------------

Description

Reads a delimited precipitation file and standardizes the selected timestamp and precipitation columns to datetime and precip_mm.

Usage

```
rf_read_rainfall(  
  file,  
  datetime_col,  
  precip_col,  
  tz = "UTC",  
  expected_interval_min = 1,  
  ambiguous = "error",  
  nonexistent = "error",  
  validate = TRUE  
)
```

Arguments

file	Path to a delimited precipitation file readable by <code>data.table::fread()</code> .
datetime_col	Name of the column containing rainfall timestamps.
precip_col	Name of the column containing precipitation amounts in millimetres.
tz	Time zone in which input timestamp values should be interpreted. Default is "UTC".
expected_interval_min	Positive finite number giving the base temporal resolution of the rainfall record, in minutes. Default is 1. Gaps consisting of positive whole multiples of this interval are allowed.
ambiguous	Handling of ambiguous local times, such as those occurring during a daylight-saving-time transition. Passed to <code>clock::date_time_parse()</code> . Default is "error".
nonexistent	Handling of nonexistent local times, such as those occurring during a daylight-saving-time transition. Passed to <code>clock::date_time_parse()</code> . Default is "error".
validate	Logical. If TRUE, the default, the standardized rainfall record is checked with <code>rf_validate_rainfall()</code> . If FALSE, this validation step is skipped.

Details

The input file does not need to contain every expected time step. Sparse rainfall records and other gaps between observations are accepted, provided timestamp differences remain compatible with the declared base temporal resolution.

The reader preserves the rainfall observations supplied by the user. It does not insert missing time steps, interpolate precipitation, estimate missing rainfall, or assess the climatological completeness of the record.

Accepted timestamp formats are:

- "DD.MM.YYYY HH:MM:SS";
- "DD.MM.YYYY HH:MM";
- "YYYY-MM-DD HH:MM:SS";
- "YYYY-MM-DD HH:MM";
- "DD.MM.YYYY";
- "YYYY-MM-DD".

Date-only values are interpreted as midnight (00:00:00) in the time zone supplied through `tz`.

Input rows are sorted chronologically when necessary. A warning is issued when sorting changes the original row order.

Value

A data frame with:

- `datetime`: parsed rainfall timestamps as POSIXct;
- `precip_mm`: precipitation amounts in millimetres.

The returned object also stores the input time zone and declared temporal resolution in the attributes `input_timezone` and `expected_interval_min`.

Examples

```
rainfall_file <- tempfile(fileext = ".csv")

writeLines(
  c(
    "time,rain",
    "2025-01-01 00:00:00,0.0",
    "2025-01-01 00:05:00,0.4",
    "2025-01-01 00:10:00,0.2"
  ),
  rainfall_file
)

rainfall <- rf_read_rainfall(
  rainfall_file,
  datetime_col = "time",
  precip_col = "rain",
  tz = "UTC",
  expected_interval_min = 5
)

rainfall
```

rf_settings

Create rainfall erosivity calculation settings

Description

Creates a settings object controlling storm separation, storm omission criteria, rainfall-intensity durations, kinetic-energy calculation, and handling of single-record rainfall events.

Usage

```
rf_settings(
  storm_break_hours = 6,
  storm_break_precip_mm = 1.27,
  omit_precip_below_mm = 12.7,
  omit_intensity_below_mm_h = 25.4,
  omit_intensity_duration_min = 15,
  omit_logic = c("all", "any"),
  energy_equation = "brown_foster_1987",
  intensity_durations_min = c(5, 10, 15, 20, 30, 60),
  single_record_energy = c("calculate", "rist_zero"),
  omit_precip = TRUE,
  omit_intensity = TRUE
)
```

Arguments

storm_break_hours

Positive number giving the elapsed time, in hours, used to separate rainfall events. Consecutive positive rainfall observations separated by no more than this duration belong to the same event. A greater separation starts a new event. Default is 6.

storm_break_precip_mm

Non-negative precipitation value associated with the RIST storm-break configuration, in mm. Default is 1.27.

Dedicated validation experiments with RIST 3.99.10 using fixed-interval rainfall input showed identical storm grouping when this value was varied from 0 to 10 mm. In the validated fixed-interval algorithm used by Rfactor, storm separation is therefore controlled by storm_break_hours; this value is retained as RIST configuration metadata and does not alter storm grouping.

omit_precip_below_mm

Non-negative precipitation threshold, in mm, used when the precipitation-based omission criterion is enabled. The condition is satisfied when event precipitation is strictly less than this value. Default is 12.70.

omit_intensity_below_mm_h

Non-negative rainfall-intensity threshold, in mm/h, used when the intensity-based omission criterion is enabled. The condition is satisfied when the selected event maximum intensity is strictly less than this value. Default is 25.40.

omit_intensity_duration_min	Duration, in minutes, of the maximum rainfall intensity used by the intensity-based omission criterion. Must be one of 5, 10, 15, 30, or 60. Default is 15. When omit_intensity = TRUE, this duration must also occur in intensity_durations_min.
omit_logic	How multiple enabled omission criteria are combined. Either "all" or "any". Default is "all". "all" omits an event only when all enabled omission conditions are satisfied. "any" omits an event when at least one enabled condition is satisfied. When only one criterion is enabled, the choice has no practical effect. When both omit_precip and omit_intensity are FALSE, no omission criterion is applied and all identified events are retained.
energy_equation	Kinetic-energy equation. One of: • "brown_foster_1987"; • "mcgregor_1995"; • "laws_parsons_1943". Default is "brown_foster_1987".
intensity_durations_min	Positive rainfall-intensity durations, in minutes, to calculate for each event. Values must be finite and must not contain duplicates. They are stored in increasing order. Default is c(5, 10, 15, 20, 30, 60). Requested durations must be compatible with the temporal resolution of the rainfall data when intensities are calculated. If the settings are used for EI30 calculation, 30 must be included.
single_record_energy	How kinetic energy should be handled for an event containing only one rainfall record. Either "calculate" or "rist_zero". Default is "calculate". "calculate" applies the selected kinetic-energy equation normally. "rist_zero" sets event energy and EI30 to zero for a one-record event, reproducing the behaviour observed during validation against RIST 3.99.10 fixed-interval input.
omit_precip	Logical. Should the precipitation-based omission criterion be enabled? Default is TRUE.
omit_intensity	Logical. Should the intensity-based omission criterion be enabled? Default is TRUE.

Details

The principal defaults correspond to the RIST configuration used during validation of this package:

- storm-break duration: 6 hours;
- RIST storm-break precipitation setting: 1.27 mm;
- precipitation omission threshold: 12.70 mm;
- intensity omission threshold: 25.40 mm/h;
- omission intensity duration: 15 minutes;

- both omission criteria enabled;
- both enabled omission conditions must be satisfied;
- Brown and Foster (1987) kinetic-energy equation.

With the default omission settings, a storm is omitted only when both

$$P < 12.70 \text{ mm}$$

and

$$I_{15} < 25.40 \text{ mm/h}$$

are satisfied. The inequalities are strict: values equal to either threshold do not satisfy that omission condition.

The default `single_record_energy = "calculate"` differs from the behaviour observed for single-record storms in RIST 3.99.10. Set `single_record_energy = "rist_zero"` when that specific RIST behaviour is required.

Value

An object of class `rf_settings` containing the validated calculation settings.

Examples

```
# Default settings
settings <- rf_settings()

# Include every identified rainfall event
include_all <- rf_settings(
  omit_precip = FALSE,
  omit_intensity = FALSE
)

# Example settings for 10-minute rainfall data
settings_10min <- rf_settings(
  intensity_durations_min = c(10, 20, 30, 60),
  omit_intensity_duration_min = 10
)
```

rf_validate_rainfall *Validate a rainfall time series*

Description

Checks the structure, precipitation values, timestamp order, and alignment with a declared base temporal resolution before rainfall erosivity calculations are performed.

Usage

```
rf_validate_rainfall(data, expected_interval_min = 1)
```

Arguments

data A data frame containing columns named `datetime` and `precip_mm`.

expected_interval_min Positive finite number giving the base temporal resolution of the rainfall record, in minutes. Consecutive timestamp differences may be positive whole multiples of this value. Default is 1.

Details

The rainfall record does not need to be temporally complete. Consecutive timestamps may be separated by gaps, provided each separation is a positive whole multiple of `expected_interval_min`. For example, with `expected_interval_min = 1`, timestamp differences of 1, 2, 10, or 120 minutes are valid, whereas a difference of 1.5 minutes is not.

Validation does not assess whether a rainfall record is climatologically complete. Missing time intervals, months, or years are not filled, interpolated, estimated, or otherwise modified.

The function requires:

- a POSIXct `datetime` column;
- a numeric `precip_mm` column;
- no missing or duplicate timestamps;
- strictly increasing timestamps;
- finite, non-missing, non-negative precipitation values;
- timestamp differences compatible with the declared base interval.

Value

TRUE, invisibly, when validation succeeds. An error is raised when the rainfall record is invalid.

Examples

```
rainfall <- data.frame(
  datetime = as.POSIXct(
    c(
      "2025-01-01 00:00:00",
      "2025-01-01 00:01:00",
      "2025-01-01 00:10:00"
    ),
    tz = "UTC"
  ),
  precip_mm = c(0, 0.2, 0.1)
)

# The nine-minute gap is allowed because it is a whole
# multiple of the one-minute base interval.
```

```
rf_validate_rainfall(  
  rainfall,  
  expected_interval_min = 1  
)
```

Index

`clock::date_time_parse()`, [15](#)
`data.table::fread()`, [15](#)

`rf_calculate_ei30`, [2](#)
`rf_calculate_ei30()`, [9](#)
`rf_calculate_energy`, [5](#)
`rf_calculate_mean_rfactor`, [7](#)
`rf_calculate_rfactor`, [9](#)
`rf_calculate_rfactor()`, [7](#)
`rf_identify_storms`, [11](#)
`rf_identify_storms()`, [2](#), [6](#), [14](#)
`rf_max_intensity`, [13](#)
`rf_max_intensity()`, [3](#)
`rf_read_rainfall`, [15](#)
`rf_read_rainfall()`, [12](#)
`rf_settings`, [17](#)
`rf_settings()`, [2–4](#), [12](#)
`rf_validate_rainfall`, [19](#)
`rf_validate_rainfall()`, [12](#), [15](#)