Package 'tidyextreme'

February 4, 2026

Title A Tidy Toolbox for Climate Extreme Indices

Version 1.0.0

URL https://github.com/Mauritia-flexuosa/tidyextreme

BugReports https://github.com/Mauritia-flexuosa/tidyextreme/issues

Description
Calculate Expert Team on Climate Change Detection and Indices (ETCCDI) <-- (acronym) climate indices from daily or hourly temperature and precipitation data. Provides flexible data handling.

Depends R (>= 4.1.0)

License MIT + file LICENSE

Encoding UTF-8

Imports dplyr (>= 1.0.0), lubridate (>= 1.8.0), zoo (>= 1.8.0), data.table (>= 1.14.0), cli (>= 3.0.0), rlang (>= 1.0.0), tidyselect (>= 1.2.1)

Suggests testthat (>= 3.0.0), knitr, rmarkdown, ggplot2, tidyr, DT, tibble

RoxygenNote 7.3.3

Config/testthat/edition 3

VignetteBuilder knitr

NeedsCompilation no

Author Marcio Baldissera Cure [aut, cre, cph]

Maintainer Marcio Baldissera Cure <marciobcure@gmail.com>

Repository CRAN

Date/Publication 2026-02-04 18:00:03 UTC

Contents

`aggregate_hourly_precipitation` *Aggregate hourly data to daily precipitation statistics*

Description

Aggregate hourly data to daily precipitation statistics

Usage

```
aggregate_hourly_precipitation(
  df_hourly,
  time_col = "datetime",
  precip_col = "precipitation",
  tz = "UTC"
)
```

Arguments

`df_hourly`	Data frame with hourly precipitation data
`time_col`	Name of the datetime column (must be POSIXct) (string)
`precip_col`	Name of the hourly precipitation column (string)
`tz`	Timezone (default: "UTC")

Value

A tibble with columns: date, prcp

`aggregate_hourly_temperature` *Aggregate hourly data to daily temperature statistics*

Description

Aggregate hourly data to daily temperature statistics

Usage

```
aggregate_hourly_temperature(
  df_hourly,
  time_col = "datetime",
  temp_col = "temperature",
  tz = "UTC"
)
```

Arguments

`df_hourly`	Data frame with hourly temperature data
`time_col`	Name of the datetime column (must be POSIXct) (string)
`temp_col`	Name of the hourly temperature column (string)
`tz`	Timezone (default: "UTC")

Value

A tibble with columns: date, tmax, tmin

`calculate_CDD` *Calculate consecutive dry days (CDD)*

Description

Calculates statistics for dry spells (consecutive days with precipitation < 1 mm), following ETCCDI definition CDD.

Usage

```
calculate_CDD(
  df,
  frequency = "daily",
  time_col = NULL,
  prcp_col = NULL,
  precip_col = NULL,
  dry_threshold = 1
)
```

Arguments

df	Data frame with precipitation data
frequency	Temporal frequency: "daily" or "hourly" (string)
time_col	Name of the time column (string). Must be in a format recognizable by lubridate (e.g., Date for daily data, POSIXct for hourly data). Recommended formats: - Daily: YYYY-MM-DD (e.g., "2023-01-15") - Hourly: YYYY-MM-DD HH:MM:SS (e.g., "2023-01-15 14:30:00")
prcp_col	Name of precipitation column (daily data) (string)
precip_col	Name of precipitation column (hourly data) (string)
dry_threshold	Threshold for dry day in mm (default: 1)

Value

A data.frame with columns: year, CDD_max, CDD_mean, CDD_median, n_dry_spells

Examples

```
# Daily precipitation data
daily_prcp <- data.frame(
  date = seq(as.Date("2000-01-01"), as.Date("2002-12-31"), by = "day"),
  rainfall = pmax(0, rgamma(1096, shape = 0.5, scale = 10))
)

# Calculate consecutive dry days statistics
calculate_CDD(
  df = daily_prcp,
  frequency = "daily",
  time_col = "date",
  prcp_col = "rainfall"
)

# With custom dry threshold (0.5mm instead of 1mm)
calculate_CDD(
  df = daily_prcp,
  frequency = "daily",
  time_col = "date",
  prcp_col = "rainfall",
  dry_threshold = 0.5
```

```
)
```

`calculate_CSDI` *Calculate Cold Spell Duration Index (CSDI)*

Description

Calculates the number of days with at least 6 consecutive days where temperature is below the 10th percentile, following ETCCDI definition CSDI.

Usage

```
calculate_CSDI(
  df,
  frequency = "daily",
  time_col = NULL,
  tmin_col = NULL,
  temp_col = NULL,
  window_days = 30,
  min_consecutive = 6
)
```

Arguments

`df`	Data frame with climate data
`frequency`	Temporal frequency: "daily" or "hourly" (string)
`time_col`	Name of the time column (string). For daily frequency, the column should be of class Date or a string in the format YYYY-MM-DD. For hourly frequency, the column should be of class POSIXct or a string in the format YYYY-MM-DD HH:MM:SS.
`tmin_col`	Name of minimum temperature column (daily data) (string)
`temp_col`	Name of temperature column (for single temp or hourly) (string)
`window_days`	Window size for percentile calculation (default: 30)
`min_consecutive`	Minimum consecutive days for cold spell (default: 6)

Value

A tibble with columns: year, CSDI, n_spells, mean_spell_length

Examples

```
# Daily data with minimum temperature
set.seed(123)
daily_data <- data.frame(
  date = seq(as.Date("2000-01-01"), as.Date("2002-12-31"), by = "day"),
  tmin = 15 + 8 * sin(seq(0, 4*pi, length.out = 1096)) + rnorm(1096, 0, 3)
)

calculate_CSDI(
  df = daily_data,
  frequency = "daily",
  time_col = "date",
  tmin_col = "tmin"
)

# With custom window and consecutive days
calculate_CSDI(
  df = daily_data,
  frequency = "daily",
  time_col = "date",
  tmin_col = "tmin",
  window_days = 15,
  min_consecutive = 5
)
```

---

calculate_CWD *Calculate consecutive wet days (CWD)*

Description

Calculates statistics for wet spells (consecutive days with precipitation $\geq$ 1 mm), following ETCCDI definition CWD.

Usage

```
calculate_CWD(
  df,
  frequency = "daily",
  time_col = NULL,
  prcp_col = NULL,
  precip_col = NULL,
  wet_threshold = 1
)
```

Arguments

df	Data frame with precipitation data
frequency	Temporal frequency: "daily" or "hourly" (string)

time_col	Name of the time column (string). Must be in a format recognizable by lubridate (e.g., Date for daily data, POSIXct for hourly data). Recommended formats: - Daily: YYYY-MM-DD (e.g., "2023-01-15") - Hourly: YYYY-MM-DD HH:MM:SS (e.g., "2023-01-15 14:30:00")
prcp_col	Name of precipitation column (daily data) (string)
precip_col	Name of precipitation column (hourly data) (string)
wet_threshold	Threshold for wet day in mm (default: 1)

Value

A data.frame with columns: year, CWD_max, CWD_mean, CWD_median, n_wet_spells

Examples

```
# Daily precipitation data
daily_prcp <- data.frame(
  date = seq(as.Date("2000-01-01"), as.Date("2002-12-31"), by = "day"),
  rainfall = pmax(0, rgamma(1096, shape = 0.5, scale = 10))
)

# Calculate consecutive wet days statistics
calculate_CWD(
  df = daily_prcp,
  frequency = "daily",
  time_col = "date",
  prcp_col = "rainfall"
)

# With custom wet threshold (5mm instead of 1mm)
calculate_CWD(
  df = daily_prcp,
  frequency = "daily",
  time_col = "date",
  prcp_col = "rainfall",
  wet_threshold = 5
)
```

---

calculate_DTR *Calculate diurnal temperature range (DTR)*

Description

Calculates the mean and standard deviation of daily temperature range (difference between maximum and minimum temperature) per year.

Usage

```
calculate_DTR(
  df,
  frequency = "daily",
  time_col = NULL,
  tmax_col = NULL,
  tmin_col = NULL,
  temp_col = NULL
)
```

Arguments

df	Data frame with climate data
frequency	Temporal frequency: "daily" or "hourly" (string)
time_col	Name of the time column (string). For daily frequency, the column should be of class Date or a string in the format YYYY-MM-DD. For hourly frequency, the column should be of class POSIXct or a string in the format YYYY-MM-DD HH:MM:SS.
tmax_col	Name of maximum temperature column (daily data) (string)
tmin_col	Name of minimum temperature column (daily data) (string)
temp_col	Name of temperature column (hourly data) (string)

Value

A tibble with columns: year, DTR_mean, DTR_sd, n_days

Examples

```
# Daily data with maximum and minimum temperature
daily_data <- data.frame(
  date = seq(as.Date("2000-01-01"), as.Date("2002-12-31"), by = "day"),
  tmax = rnorm(1096, mean = 25, sd = 5),
  tmin = rnorm(1096, mean = 15, sd = 5)
)

calculate_DTR(
  df = daily_data,
  frequency = "daily",
  time_col = "date",
  tmax_col = "tmax",
  tmin_col = "tmin"
)
```

`calculate_PRCPstats` *Calculate annual precipitation totals and statistics*

Description

Calculates comprehensive annual precipitation statistics including total precipitation, number of wet days, mean daily precipitation, and maximum daily precipitation.

Usage

```
calculate_PRCPstats(
  df,
  frequency = "daily",
  time_col = NULL,
  prcp_col = NULL,
  precip_col = NULL,
  wet_threshold = 1
)
```

Arguments

`df`	Data frame with precipitation data
`frequency`	Temporal frequency: "daily" or "hourly" (string)
`time_col`	Name of the time column (string). Must be in a format recognizable by lubridate (e.g., Date for daily data, POSIXct for hourly data). Recommended formats: - Daily: YYYY-MM-DD (e.g., "2023-01-15") - Hourly: YYYY-MM-DD HH:MM:SS (e.g., "2023-01-15 14:30:00")
`prcp_col`	Name of precipitation column (daily data) (string)
`precip_col`	Name of precipitation column (hourly data) (string)
`wet_threshold`	Threshold for wet day in mm (default: 1)

Value

A data.frame with columns: year, PRCP_total, PRCP_days, PRCP_mean, PRCP_max

Examples

```
# Daily precipitation data
daily_prcp <- data.frame(
  date = seq(as.Date("2000-01-01"), as.Date("2002-12-31"), by = "day"),
  rainfall = pmax(0, rgamma(1096, shape = 0.5, scale = 10))
)

# Calculate comprehensive precipitation statistics
calculate_PRCPstats(
  df = daily_prcp,
  frequency = "daily",
```

```
  time_col = "date",
  prcp_col = "rainfall"
)

# With custom wet threshold (2mm instead of 1mm)
calculate_PRCPstats(
  df = daily_prcp,
  frequency = "daily",
  time_col = "date",
  prcp_col = "rainfall",
  wet_threshold = 2
)

# Hourly precipitation data (converted to daily)
hourly_prcp <- data.frame(
  datetime = seq(
    as.POSIXct("2000-01-01 00:00", tz = "UTC"),
    as.POSIXct("2000-01-31 23:00", tz = "UTC"),
    by = "hour"
  ),
  precip = pmax(0, rgamma(31*24, shape = 0.3, scale = 2))
)

calculate_PRCPstats(
  df = hourly_prcp,
  frequency = "hourly",
  time_col = "datetime",
  precip_col = "precip"
)
```

`calculate_R10mm` *Calculate number of heavy precipitation days (R10mm)*

Description

Counts the number of days per year when precipitation $\geq$ 10 mm, following ETCCDI definition R10mm.

Usage

```
calculate_R10mm(
  df,
  frequency = "daily",
  time_col = NULL,
  prcp_col = NULL,
  precip_col = NULL,
  threshold = 10
)
```

Arguments

df	Data frame with precipitation data
frequency	Temporal frequency: "daily" or "hourly" (string)
time_col	Name of the time column (string). Must be in a format recognizable by lubridate (e.g., Date for daily data, POSIXct for hourly data). Recommended formats: - Daily: YYYY-MM-DD (e.g., "2023-01-15") - Hourly: YYYY-MM-DD HH:MM:SS (e.g., "2023-01-15 14:30:00")
prcp_col	Name of precipitation column (daily data) (string)
precip_col	Name of precipitation column (hourly data) (string)
threshold	Precipitation threshold in mm (default: 10)

Value

A data.frame with columns: year, R10mm

Examples

```
# Daily precipitation data
daily_prcp <- data.frame(
  date = seq(as.Date("2000-01-01"), as.Date("2002-12-31"), by = "day"),
  rainfall = pmax(0, rgamma(1096, shape = 0.5, scale = 10))
)

# Calculate number of days with precipitation \eqn{\geq} 10mm
calculate_R10mm(
  df = daily_prcp,
  frequency = "daily",
  time_col = "date",
  prcp_col = "rainfall"
)

# With custom threshold (15mm instead of 10mm)
calculate_R10mm(
  df = daily_prcp,
  frequency = "daily",
  time_col = "date",
  prcp_col = "rainfall",
  threshold = 15
)
```

---

calculate_R1mm *Calculate number of days with precipitation $\geq$ 1mm (R1mm)*

Description

Counts the number of days per year when precipitation $\geq$ 1 mm, representing wet days.

Usage

```
calculate_R1mm(
  df,
  frequency = "daily",
  time_col = NULL,
  prcp_col = NULL,
  precip_col = NULL,
  threshold = 1
)
```

Arguments

df	Data frame with precipitation data
frequency	Temporal frequency: "daily" or "hourly"
time_col	Name of the time column (string). Must be in a format recognizable by lubridate (e.g., Date for daily data, POSIXct for hourly data). Recommended formats: - Daily: YYYY-MM-DD (e.g., "2023-01-15") - Hourly: YYYY-MM-DD HH:MM:SS (e.g., "2023-01-15 14:30:00")
prcp_col	Name of precipitation column (daily data) (string)
precip_col	Name of precipitation column (hourly data) (string)
threshold	Precipitation threshold in mm (default: 1)

Value

A data.frame with columns: year, R1mm

Examples

```
# Daily precipitation data
daily_prcp <- data.frame(
  date = seq(as.Date("2000-01-01"), as.Date("2002-12-31"), by = "day"),
  rainfall = pmax(0, rgamma(1096, shape = 0.5, scale = 10))
)

# Calculate number of days with precipitation \eqn{\geq} 1mm (wet days)
calculate_R1mm(
  df = daily_prcp,
  frequency = "daily",
  time_col = "date",
  prcp_col = "rainfall"
)

# With custom threshold (0.5mm instead of 1mm)
calculate_R1mm(
  df = daily_prcp,
  frequency = "daily",
  time_col = "date",
  prcp_col = "rainfall",
  threshold = 0.5
```

```
)
```

`calculate_R20mm` *Calculate number of very heavy precipitation days (R20mm)*

Description

Counts the number of days per year when precipitation $\geq$ 20 mm, following ETCCDI definition R20mm.

Usage

```
calculate_R20mm(
  df,
  frequency = "daily",
  time_col = NULL,
  prcp_col = NULL,
  precip_col = NULL,
  threshold = 20
)
```

Arguments

`df`	Data frame with precipitation data
`frequency`	Temporal frequency: "daily" or "hourly"
`time_col`	Name of the time column (string). Must be in a format recognizable by lubridate (e.g., Date for daily data, POSIXct for hourly data). Recommended formats: - Daily: YYYY-MM-DD (e.g., "2023-01-15") - Hourly: YYYY-MM-DD HH:MM:SS (e.g., "2023-01-15 14:30:00")
`prcp_col`	Name of precipitation column (daily data) (string)
`precip_col`	Name of precipitation column (hourly data) (string)
`threshold`	Precipitation threshold in mm (default: 20)

Value

A data.frame with columns: year, R20mm

Examples

```
# Daily precipitation data
daily_prcp <- data.frame(
  date = seq(as.Date("2000-01-01"), as.Date("2002-12-31"), by = "day"),
  rainfall = pmax(0, rgamma(1096, shape = 0.5, scale = 10))
)

# Calculate number of days with precipitation \eqn{\geq} 20mm
calculate_R20mm(
```

```
  df = daily_prcp,
  frequency = "daily",
  time_col = "date",
  prcp_col = "rainfall"
)

# With custom threshold (25mm instead of 20mm)
calculate_R20mm(
  df = daily_prcp,
  frequency = "daily",
  time_col = "date",
  prcp_col = "rainfall",
  threshold = 25
)
```

`calculate_Rx1day` *Calculate maximum 1-day precipitation (Rx1day)*

Description

Calculates the annual maximum 1-day precipitation amount, following ETCCDI definition Rx1day.

Usage

```
calculate_Rx1day(
  df,
  frequency = "daily",
  time_col = NULL,
  prcp_col = NULL,
  precip_col = NULL,
  min_valid_years = 1
)
```

Arguments

`df`	Data frame with precipitation data
`frequency`	Temporal frequency: "daily" or "hourly" (string)
`time_col`	Name of the time column (string). For daily frequency, the column should be of class Date or a string in the format YYYY-MM-DD. For hourly frequency, the column should be of class POSIXct or a string in the format YYYY-MM-DD HH:MM:SS.
`prcp_col`	Name of precipitation column (daily data) (string)
`precip_col`	Name of precipitation column (hourly data) (string)
`min_valid_years`	Minimum years with valid data (default: 1)

Value

A data.frame with columns: year, Rx1day

Examples

```
# Daily precipitation data
daily_prcp <- data.frame(
  date = seq(as.Date("2000-01-01"), as.Date("2002-12-31"), by = "day"),
  rainfall = pmax(0, rgamma(1096, shape = 0.5, scale = 10))
)

calculate_Rx1day(
  df = daily_prcp,
  frequency = "daily",
  time_col = "date",
  prcp_col = "rainfall"
)
```

---

calculate_Rx5day *Calculate maximum consecutive 5-day precipitation (Rx5day)*

Description

Calculates the annual maximum precipitation amount accumulated over 5 consecutive days, following ETCCDI definition Rx5day.

Usage

```
calculate_Rx5day(
  df,
  frequency = "daily",
  time_col = NULL,
  prcp_col = NULL,
  precip_col = NULL
)
```

Arguments

df	Data frame with precipitation data
frequency	Temporal frequency: "daily" or "hourly" (string)
time_col	Name of the time column (string). Must be in a format recognizable by lubridate (e.g., Date for daily data, POSIXct for hourly data). Recommended formats: - Daily: YYYY-MM-DD (e.g., "2023-01-15") - Hourly: YYYY-MM-DD HH:MM:SS (e.g., "2023-01-15 14:30:00")
prcp_col	Name of precipitation column (daily data) (string)
precip_col	Name of precipitation column (hourly data) (string)

Value

A data.frame with columns: year, Rx5day

Examples

```
# Daily precipitation data
daily_prcp <- data.frame(
  date = seq(as.Date("2000-01-01"), as.Date("2002-12-31"), by = "day"),
  rainfall = pmax(0, rgamma(1096, shape = 0.5, scale = 10))
)

# Calculate maximum 5-day precipitation
calculate_Rx5day(
  df = daily_prcp,
  frequency = "daily",
  time_col = "date",
  prcp_col = "rainfall"
)

# Hourly precipitation data (converted to daily)
hourly_prcp <- data.frame(
  datetime = seq(
    as.POSIXct("2000-01-01 00:00", tz = "UTC"),
    as.POSIXct("2000-01-31 23:00", tz = "UTC"),
    by = "hour"
  ),
  precip = pmax(0, rgamma(31*24, shape = 0.3, scale = 2))
)

calculate_Rx5day(
  df = hourly_prcp,
  frequency = "hourly",
  time_col = "datetime",
  precip_col = "precip"
)
```

---

calculate_SDII *Calculate Simple Daily Intensity Index (SDII)*

Description

Calculates the mean precipitation amount on wet days ($\geq$ 1 mm), following ETCCDI definition SDII.

Usage

```
calculate_SDII(
  df,
  frequency = "daily",
```

```
  time_col = NULL,
  prcp_col = NULL,
  precip_col = NULL,
  wet_threshold = 1
)
```

Arguments

df	Data frame with precipitation data
frequency	Temporal frequency: "daily" or "hourly" (string)
time_col	Name of the time column (string). Must be in a format recognizable by lubridate (e.g., Date for daily data, POSIXct for hourly data). Recommended formats: - Daily: YYYY-MM-DD (e.g., "2023-01-15") - Hourly: YYYY-MM-DD HH:MM:SS (e.g., "2023-01-15 14:30:00")
prcp_col	Name of precipitation column (daily data) (string)
precip_col	Name of precipitation column (hourly data) (string)
wet_threshold	Threshold for wet day in mm (default: 1)

Value

A data.frame with columns: year, SDII, wet_days, total_prcp

Examples

```
# Daily precipitation data
daily_prcp <- data.frame(
  date = seq(as.Date("2000-01-01"), as.Date("2002-12-31"), by = "day"),
  rainfall = pmax(0, rgamma(1096, shape = 0.5, scale = 10))
)

# Calculate Simple Daily Intensity Index
calculate_SDII(
  df = daily_prcp,
  frequency = "daily",
  time_col = "date",
  prcp_col = "rainfall"
)

# With custom wet threshold (5mm instead of 1mm)
calculate_SDII(
  df = daily_prcp,
  frequency = "daily",
  time_col = "date",
  prcp_col = "rainfall",
  wet_threshold = 5
)
```

`calculate_TN0` *Calculate number of days with temperature < 0°C*

Description

Counts the number of days per year when daily temperature is less than 0°C.

Usage

```
calculate_TN0(
  df,
  frequency = "daily",
  time_col = NULL,
  tmin_col = NULL,
  temp_col = NULL
)
```

Arguments

`df`	Data frame with climate data
`frequency`	Temporal frequency: "daily" or "hourly" (string)
`time_col`	Name of the time column (string). For daily frequency, the column should be of class Date or a string in the format YYYY-MM-DD. For hourly frequency, the column should be of class POSIXct or a string in the format YYYY-MM-DD HH:MM:SS.
`tmin_col`	Name of minimum temperature column (daily data) (string)
`temp_col`	Name of temperature column (for single temp or hourly) (string)

Value

A tibble with columns: year, TN0

Examples

```
# Daily data with minimum temperature
daily_data <- data.frame(
  date = seq(as.Date("2000-01-01"), as.Date("2002-12-31"), by = "day"),
  tmin = rnorm(1096, mean = 5, sd = 5)
)

calculate_TN0(
  df = daily_data,
  frequency = "daily",
  time_col = "date",
  tmin_col = "tmin"
)
```

`calculate_TN10p` *Calculate 10th percentile of daily temperature (TN10p)*

Description

Calculates the 10th percentile of daily temperature per year, used as threshold for extreme cold nights.

Usage

```
calculate_TN10p(
  df,
  frequency = "daily",
  time_col = NULL,
  tmin_col = NULL,
  temp_col = NULL
)
```

Arguments

`df`	Data frame with climate data
`frequency`	Temporal frequency: "daily" or "hourly" (string)
`time_col`	Name of the time column (string). For daily frequency, the column should be of class Date or a string in the format YYYY-MM-DD. For hourly frequency, the column should be of class POSIXct or a string in the format YYYY-MM-DD HH:MM:SS.
`tmin_col`	Name of minimum temperature column (daily data) (string)
`temp_col`	Name of temperature column (for single temp or hourly) (string)

Value

A tibble with columns: year, TN10p

Examples

```
# Daily data with minimum temperature
daily_data <- data.frame(
  date = seq(as.Date("2000-01-01"), as.Date("2002-12-31"), by = "day"),
  tmin = rnorm(1096, mean = 10, sd = 5)
)

calculate_TN10p(
  df = daily_data,
  frequency = "daily",
  time_col = "date",
  tmin_col = "tmin"
)
```

`calculate_TNn` *Calculate monthly minimum value of daily minimum temperature (TNn)*

Description

Calculates the lowest daily minimum temperature for each month, following ETCCDI definition TNn.

Usage

```
calculate_TNn(
  df,
  frequency = "daily",
  time_col = NULL,
  tmin_col = NULL,
  temp_col = NULL,
  min_days = 20
)
```

Arguments

`df`	Data frame with climate data
`frequency`	Temporal frequency: "daily" or "hourly"
`time_col`	Name of the time column. For daily frequency, the column should be of class Date or a string in the format YYYY-MM-DD. For hourly frequency, the column should be of class POSIXct or a string in the format YYYY-MM-DD HH:MM:SS.
`tmin_col`	Name of minimum temperature column (daily data)
`temp_col`	Name of temperature column (for single temp or hourly)
`min_days`	Minimum days per month for valid calculation (default: 20)

Value

A tibble with columns: year, month, TNn

Examples

```
# Daily data with minimum temperature
daily_data <- data.frame(
  date = seq(as.Date("2000-01-01"), as.Date("2002-12-31"), by = "day"),
  tmin = rnorm(1096, mean = 10, sd = 5)
)

calculate_TNn(
  df = daily_data,
```

```
  frequency = "daily",
  time_col = "date",
  tmin_col = "tmin"
)

# With custom minimum days per month
calculate_TNn(
  df = daily_data,
  frequency = "daily",
  time_col = "date",
  tmin_col = "tmin",
  min_days = 25
)
```

`calculate_TR20` *Calculate number of tropical nights (TN > 20ˆ∘C)*

Description

Counts the number of days per year when daily minimum temperature exceeds 20°C, following ETCCDI definition TR20.

Usage

```
calculate_TR20(
  df,
  frequency = "daily",
  time_col = NULL,
  tmin_col = NULL,
  temp_col = NULL,
  threshold = 20
)
```

Arguments

`df`	Data frame with climate data
`frequency`	Temporal frequency: "daily" or "hourly"
`time_col`	Name of the time column. For daily frequency, the column should be of class Date or a string in the format YYYY-MM-DD. For hourly frequency, the column should be of class POSIXct or a string in the format YYYY-MM-DD HH:MM:SS.
`tmin_col`	Name of minimum temperature column (daily data)
`temp_col`	Name of temperature column (for single temp or hourly)
`threshold`	Temperature threshold in °C (default: 20)

Value

A tibble with columns: year, TR20

Examples

```
# Daily data with separate min temperature
daily_data <- data.frame(
  date = seq(as.Date("2000-01-01"), as.Date("2002-12-31"), by = "day"),
  tmin = rnorm(1096, mean = 18, sd = 5)
)

calculate_TR20(
  df = daily_data,
  frequency = "daily",
  time_col = "date",
  tmin_col = "tmin"
)

# Hourly data (will be aggregated to daily min temperature)
hourly_data <- data.frame(
  datetime = seq(
    as.POSIXct("2000-01-01 00:00", tz = "UTC"),
    as.POSIXct("2000-01-31 23:00", tz = "UTC"),
    by = "hour"
  ),
  temperature = rnorm(31*24, mean = 16, sd = 3)
)

calculate_TR20(
  df = hourly_data,
  frequency = "hourly",
  time_col = "datetime",
  temp_col = "temperature",
  threshold = 20
)
```

---

calculate_TX25 *Calculate number of summer days (TX > 25ˆ∘C)*

Description

Counts the number of days per year when daily maximum temperature exceeds 25°C, following ETCCDI definition SU25.

Usage

```
calculate_TX25(
  df,
  frequency = "daily",
  time_col = NULL,
  tmax_col = NULL,
  temp_col = NULL,
  threshold = 25
)
```

Arguments

df	Data frame with climate data
frequency	Temporal frequency: "daily" or "hourly"
time_col	Name of the time column. For daily frequency, the column should be of class Date or a string in the format YYYY-MM-DD. For hourly frequency, the column should be of class POSIXct or a string in the format YYYY-MM-DD HH:MM:SS.
tmax_col	Name of maximum temperature column (daily data)
temp_col	Name of temperature column (for single temp or hourly)
threshold	Temperature threshold in °C (default: 25)

Value

A tibble with columns: year, TX25

Examples

```
# Daily data with separate max/min
daily_data <- data.frame(
  date = seq(as.Date("2000-01-01"), as.Date("2002-12-31"), by = "day"),
  tmax = rnorm(1096, mean = 25, sd = 6)
)

calculate_TX25(
  df = daily_data,
  frequency = "daily",
  time_col = "date",
  tmax_col = "tmax"
)

# Hourly data
hourly_data <- data.frame(
  datetime = seq(
    as.POSIXct("2000-01-01 00:00", tz = "UTC"),
    as.POSIXct("2000-01-31 23:00", tz = "UTC"),
    by = "hour"
  ),
  temperature = rnorm(31*24, mean = 22, sd = 4)
)

calculate_TX25(
  df = hourly_data,
  frequency = "hourly",
  time_col = "datetime",
  temp_col = "temperature",
  threshold = 25
)
```

`calculate_TX30` *Calculate number of days with temperature $\geq 30^{\circ}C$*

Description

Counts the number of days per year when daily temperature is greater than or equal to 30°C.

Usage

```
calculate_TX30(
  df,
  frequency = "daily",
  time_col = NULL,
  tmax_col = NULL,
  temp_col = NULL
)
```

Arguments

`df`	Data frame with climate data
`frequency`	Temporal frequency: "daily" or "hourly"
`time_col`	Name of the time column. For daily frequency, the column should be of class Date or a string in the format YYYY-MM-DD. For hourly frequency, the column should be of class POSIXct or a string in the format YYYY-MM-DD HH:MM:SS.
`tmax_col`	Name of maximum temperature column (daily data)
`temp_col`	Name of temperature column (for single temp or hourly)

Value

A tibble with columns: year, TX30

Examples

```
# Daily data with maximum temperature
daily_data <- data.frame(
  date = seq(as.Date("2000-01-01"), as.Date("2002-12-31"), by = "day"),
  tmax = rnorm(1096, mean = 25, sd = 6)
)

calculate_TX30(
  df = daily_data,
  frequency = "daily",
  time_col = "date",
  tmax_col = "tmax"
)
```

`calculate_TX35` *Calculate number of days with temperature $\geq 35^{\circ}C$*

Description

Counts the number of days per year when daily temperature is greater than or equal to 35°C.

Usage

```
calculate_TX35(
  df,
  frequency = "daily",
  time_col = NULL,
  tmax_col = NULL,
  temp_col = NULL
)
```

Arguments

`df`	Data frame with climate data
`frequency`	Temporal frequency: "daily" or "hourly" (string)
`time_col`	Name of the time column (string). For daily frequency, the column should be of class Date or a string in the format YYYY-MM-DD. For hourly frequency, the column should be of class POSIXct or a string in the format YYYY-MM-DD HH:MM:SS.
`tmax_col`	Name of maximum temperature column (daily data) (string)
`temp_col`	Name of temperature column (for single temp or hourly) (string)

Value

A tibble with columns: year, TX35

Examples

```
# Daily data with maximum temperature
daily_data <- data.frame(
  date = seq(as.Date("2000-01-01"), as.Date("2002-12-31"), by = "day"),
  tmax = rnorm(1096, mean = 25, sd = 6)
)

calculate_TX35(
  df = daily_data,
  frequency = "daily",
  time_col = "date",
  tmax_col = "tmax"
)
```

`calculate_TX90p` *Calculate 90th percentile of daily temperature (TX90p)*

Description

Calculates the 90th percentile of daily temperature per year, used as threshold for extreme warm days.

Usage

```
calculate_TX90p(
  df,
  frequency = "daily",
  time_col = NULL,
  tmax_col = NULL,
  temp_col = NULL
)
```

Arguments

`df`	Data frame with climate data
`frequency`	Temporal frequency: "daily" or "hourly" (string)
`time_col`	Name of the time column (string). For daily frequency, the column should be of class Date or a string in the format YYYY-MM-DD. For hourly frequency, the column should be of class POSIXct or a string in the format YYYY-MM-DD HH:MM:SS.
`tmax_col`	Name of maximum temperature column (daily data) (string)
`temp_col`	Name of temperature column (for single temp or hourly) (string)

Value

A tibble with columns: year, TX90p

Examples

```
# Daily data with maximum temperature
daily_data <- data.frame(
  date = seq(as.Date("2000-01-01"), as.Date("2002-12-31"), by = "day"),
  tmax = rnorm(1096, mean = 25, sd = 6)
)

calculate_TX90p(
  df = daily_data,
  frequency = "daily",
  time_col = "date",
  tmax_col = "tmax"
)
```

`calculate_TXx` *Calculate monthly maximum value of daily maximum temperature (TXx)*

Description

Calculates the highest daily maximum temperature for each month, following ETCCDI definition TXx.

Usage

```
calculate_TXx(
  df,
  frequency = "daily",
  time_col = NULL,
  tmax_col = NULL,
  temp_col = NULL,
  min_days = 20
)
```

Arguments

`df`	Data frame with climate data
`frequency`	Temporal frequency: "daily" or "hourly"
`time_col`	Name of the time column. For daily frequency, the column should be of class Date or a string in the format YYYY-MM-DD. For hourly frequency, the column should be of class POSIXct or a string in the format YYYY-MM-DD HH:MM:SS.
`tmax_col`	Name of maximum temperature column (daily data)
`temp_col`	Name of temperature column (for single temp or hourly)
`min_days`	Minimum days per month for valid calculation (default: 20)

Value

A tibble with columns: year, month, TXx

Examples

```
# Daily data with maximum temperature
daily_data <- data.frame(
  date = seq(as.Date("2000-01-01"), as.Date("2002-12-31"), by = "day"),
  tmax = rnorm(1096, mean = 25, sd = 6)
)

calculate_TXx(
  df = daily_data,
```

```
  frequency = "daily",
  time_col = "date",
  tmax_col = "tmax"
)

# With custom minimum days per month
calculate_TXx(
  df = daily_data,
  frequency = "daily",
  time_col = "date",
  tmax_col = "tmax",
  min_days = 25
)
```

`calculate_WSDI` *Calculate Warm Spell Duration Index (WSDI)*

Description

Calculates the number of days with at least 6 consecutive days where temperature exceeds the 90th percentile, following ETCCDI definition WSDI.

Usage

```
calculate_WSDI(
  df,
  frequency = "daily",
  time_col = NULL,
  tmax_col = NULL,
  temp_col = NULL,
  window_days = 30,
  min_consecutive = 6
)
```

Arguments

`df`	Data frame with climate data
`frequency`	Temporal frequency: "daily" or "hourly" (string)
`time_col`	Name of the time column (string). For daily frequency, the column should be of class Date or a string in the format YYYY-MM-DD. For hourly frequency, the column should be of class POSIXct or a string in the format YYYY-MM-DD HH:MM:SS.
`tmax_col`	Name of maximum temperature column (daily data) (string)
`temp_col`	Name of temperature column (for single temp or hourly) (string)
`window_days`	Window size for percentile calculation (default: 30)
`min_consecutive`	Minimum consecutive days for warm spell (default: 6)

Value

A tibble with columns: year, WSDI, n_spells, mean_spell_length

Examples

```
# Daily data with maximum temperature
set.seed(123)
daily_data <- data.frame(
  date = seq(as.Date("2000-01-01"), as.Date("2002-12-31"), by = "day"),
  tmax = 25 + 10 * sin(seq(0, 4*pi, length.out = 1096)) + rnorm(1096, 0, 5)
)

calculate_WSDI(
  df = daily_data,
  frequency = "daily",
  time_col = "date",
  tmax_col = "tmax"
)

# With custom window and consecutive days
calculate_WSDI(
  df = daily_data,
  frequency = "daily",
  time_col = "date",
  tmax_col = "tmax",
  window_days = 15,
  min_consecutive = 5
)
```

---

list_indices *List available climate indices*

Description

List available climate indices

Usage

```
list_indices()
```

Value

A data frame with available indices and descriptions

Index