Package 'rmet'

November 12, 2026

Type Package

Title Download and Read Brazilian Meteorological Data from INMET

Version 0.4.0

Description Automates the download and processing of historical hourly weather data from automatic stations of the Brazilian National Institute of Meteorology (INMET). It provides the official station catalogue with an offline snapshot and nearest-station search, validated and resumable downloads that detect updated archives, and a fast parser for the formatting and encoding inconsistencies of raw CSV files across years. It returns analysis-ready data frames with standardized column names and UTC timestamps. It flags implausible values with transparent quality-control checks, reports data completeness, and builds daily summaries in each station's local time with explicit rules for incomplete days. Data are retrieved from <https://portal.inmet.gov.br/dadoshistoricos> and <https://apitempo.inmet.gov.br/estacoes/T>.

Depends R (>= 4.0.0)

Encoding UTF-8

LazyData true

Language en-US

License GPL (>= 3)

URL https://github.com/rodrigosqrt3/rmet

BugReports https://github.com/rodrigosqrt3/rmet/issues

RoxygenNote 7.3.3

Imports curl (>= 4.0), jsonlite

Suggests spelling, testthat (>= 3.0.0), knitr, rmarkdown, sf, withr

VignetteBuilder knitr

Config/testthat/edition 3

NeedsCompilation yes

Author Rodrigo Fonseca Villa [aut, cre] (ORCID: <https://orcid.org/0009-0005-2938-2270>)

Maintainer Rodrigo Fonseca Villa <rodrigo03.villa@gmail.com>

Repository CRAN

Date/Publication 2026-11-12 15:20:02 UTC

Contents

inmet_anomaly . . . 3
inmet_as_sf . . . 4
inmet_availability . . . 4
inmet_cache_clear . . . 5
inmet_cache_dir . . . 6
inmet_cache_status . . . 7
inmet_completeness . . . 7
inmet_convert_radiation . . . 8
inmet_daily . . . 9
inmet_derive . . . 10
inmet_design_et0 . . . 11
inmet_download . . . 12
inmet_et0 . . . 13
inmet_extract . . . 14
inmet_fill . . . 15
inmet_get . . . 16
inmet_indices . . . 18
inmet_interpolate . . . 19
inmet_monthly . . . 20
inmet_nearest . . . 21
inmet_normals . . . 22
inmet_pet_thornthwaite . . . 23
inmet_qc . . . 24
inmet_read . . . 25
inmet_spi . . . 27
inmet_stations . . . 28
inmet_stations_within . . . 29
inmet_timezone . . . 30
inmet_water_balance . . . 31
rmet-classes . . . 32
rmet_example . . . 33

Index **34**

`inmet_anomaly` *Anomalies relative to the climatological normal*

Description

Subtracts (or divides by) the normal of each calendar month from the values of `inmet_monthly()`.

Usage

```
inmet_anomaly(x, normals = NULL, variables = NULL, relative = FALSE, ...)
```

Arguments

`x`	A data frame as returned by `inmet_monthly()`.
`normals`	A data frame from `inmet_normals()`. By default the normals are computed from `x` itself, which is only appropriate when `x` covers a long reference period.
`variables`	Numeric columns to compare; by default the variables of `normals`.
`relative`	If `TRUE`, the anomaly is the percentage of the normal (useful for precipitation) instead of the difference.
`...`	Further arguments passed to `inmet_normals()` when `normals` is not given, such as `years` or `min_years`.

Value

`x` with one `<variable>_anom` column per variable.

See Also

`inmet_normals()`

Examples

```
monthly <- inmet_monthly(inmet_daily(rmet_example))
anomalies <- inmet_anomaly(monthly, min_years = 1)
anomalies[, c("month", "temp_mean_c", "temp_mean_c_anom")]
```

inmet_as_sf *Convert stations or data to spatial points*

Description

Turns any data frame with `latitude` and `longitude` columns, such as the output of `inmet_stations()`, `inmet_nearest()`, `inmet_read()` or `inmet_daily()`, into an `sf` object for mapping and spatial joins. Requires the **sf** package.

Usage

```
inmet_as_sf(x, remove = FALSE)
```

Arguments

`x`	A data frame with numeric `latitude` and `longitude` columns.
`remove`	Drop the coordinate columns from the attributes.

Value

An `sf` data frame with point geometries in WGS 84 (EPSG:4326). Rows with missing coordinates are dropped with a warning.

Examples

```
if (requireNamespace("sf", quietly = TRUE)) {
  rs <- inmet_stations(state = "RS", source = "bundled")
  points <- inmet_as_sf(rs)
  plot(points["elevation"])
}
```

inmet_availability *Which stations and periods are in the cached archives*

Description

Lists the station files inside the annual archives that were already downloaded, without reading any data. It is a quick way to see which stations have data in a year, and which period each one covers, before calling `inmet_read()`.

Usage

```
inmet_availability(years = NULL, stations = NULL, dest_dir = inmet_cache_dir())
```

Arguments

years	Integer years to list, or NULL (default) for every cached archive. Years without a valid cached archive are skipped with a warning.
stations	Optional station codes, such as "A801".
dest_dir	Directory with the <year>.zip archives, inmet_cache_dir() by default.

Details

Only the file names are used, so the call is fast even for archives of 100 MB. The period reflects what INMET announces in the names; a station can still have gaps inside it (see inmet_completeness()).

Value

A data frame with one row per station and year: station_code, state, year, start and end (first and last day covered by the station files, from their names; NA when a name has no period) and n_files (the current-year archive has several cumulative files per station).

See Also

inmet_download(), inmet_stations(), inmet_cache_status()

Examples

```
# an empty cache gives an empty result
inmet_availability(dest_dir = file.path(tempdir(), "rmet-empty"))
```

---

inmet_cache_clear *Clear the rmet cache*

Description

Clear the rmet cache

Usage

```
inmet_cache_clear(
  years = NULL,
  dest_dir = inmet_cache_dir(),
  ask = interactive(),
  catalog = FALSE
)
```

Arguments

years	Integer years to remove, or NULL for every archive.
dest_dir	Cache directory.
ask	Ask for confirmation before deleting.
catalog	Also remove the cached official station catalogue.

Value

Invisible NULL.

Examples

```
inmet_cache_clear(dest_dir = file.path(tempdir(), "rmet"), ask = FALSE)
```

---

inmet_cache_dir *Default cache directory of rmet*

Description

Returns the directory where rmet stores downloaded archives and the station catalogue. It is the first one that is set among the option `rmet.cache_dir`, the environment variable `RMET_CACHE_DIR`, and the user-level cache directory of R (`tools::R_user_dir("rmet", "cache")`). Every function that takes a `dest_dir` or `cache_dir` argument uses it as the default.

Usage

```
inmet_cache_dir()
```

Value

A single path.

See Also

`inmet_cache_status()`, `inmet_cache_clear()`

Examples

```
inmet_cache_dir()

# use a different directory for the whole session
old <- options(rmet.cache_dir = file.path(tempdir(), "rmet"))
inmet_cache_dir()
options(old)
```

inmet_cache_status *Inspect downloaded INMET archives*

Description

Inspect downloaded INMET archives

Usage

```
inmet_cache_status(dest_dir = inmet_cache_dir())
```

Arguments

dest_dir	Cache directory used by inmet_download(). Defaults to inmet_cache_dir().

Value

A data frame with one row per cached year: archive path, size in MB, whether the ZIP is valid, whether a partial download exists, and when the archive was downloaded (UTC, if recorded). Returned invisibly when the cache is empty.

Examples

```
inmet_cache_status(dest_dir = file.path(tempdir(), "rmet"))
```

inmet_completeness *Data completeness by station and period*

Description

Summarises how complete hourly data are, to choose stations and periods before an analysis.

Usage

```
inmet_completeness(x, by = c("month", "year"), variables = NULL)
```

Arguments

x	A data frame as returned by inmet_read().
by	Period length: "month" (default) or "year". Periods are UTC calendar months or years.
variables	Measurement columns to assess. Defaults to all numeric measurement columns.

Details

The expected number of hours is the full length of the period, so periods only partly covered by `x` (for example, the current month, or a period cut by `start_date` in `inmet_read()`) show low completeness. Global radiation is missing at night at most stations, so its completeness is usually around 50 %.

Value

A data frame with one row per station and period: `station_code`, `period` (`"YYYY-MM"` or `"YYYY"`), `hours_expected` (hours in the period), `hours_present` (hourly records in `x`), and, for each variable, the percentage of the expected hours with a valid value. Values flagged by `inmet_qc()` do not count as valid.

See Also

`inmet_qc()`, `inmet_daily()`

Examples

```
inmet_completeness(rmet_example, variables = c("temp_dry_c", "precip_mm"))
```

---

inmet_convert_radiation

Convert units of solar radiation

Description

Converts daily solar radiation between the units used in agrometeorology.

Usage

```
inmet_convert_radiation(x, from, to)
```

Arguments

`x`	Numeric vector of daily radiation.
`from`, `to`	Units: `"mj_m2"` (MJ per square metre per day), `"kj_m2"`, `"j_cm2"`, `"cal_cm2"`, `"w_m2"` (daily mean power) and `"mm"` (equivalent evaporation, mm per day).

Value

A numeric vector.

See Also

`inmet_daily()`

Examples

```
inmet_convert_radiation(20, from = "mj_m2", to = "w_m2")
```

`inmet_daily` *Daily summaries of hourly INMET data*

Description

Aggregates the hourly output of `inmet_read()` or `inmet_get()` into one row per station and local calendar day, handling the conventions of the source data correctly.

Usage

```
inmet_daily(x, tz = "auto", min_hours = 20L)
```

Arguments

`x`	A data frame as returned by `inmet_read()`, with at least a `datetime` column.
`tz`	Time zone defining the calendar day. `"auto"` (default) uses each station's own local time, from its state and longitude (see `inmet_timezone()`); this needs the `state` column. Otherwise, any name in `OlsonNames()`, for example `"America/Sao_Paulo"`, or `"UTC"` for UTC days.
`min_hours`	Minimum number of non-missing hourly values a variable needs on a given day; otherwise its daily value is `NA`. Local days have 24 hours (23 or 25 on daylight-saving transitions, which Brazil observed until 2019).

Details

INMET stamps every record with the end of the hour it describes, in UTC. A record is therefore assigned to the local day containing the *middle* of its hour (`datetime` minus 30 minutes): the value stamped 03:00 UTC, for example, describes 23:00 to 00:00 Brasilia time and belongs to the earlier day. Grouping with `as.Date(datetime)` instead would use UTC days, which in Brazil end at 21:00 or 22:00 local time.

Radiation is usually missing at night, so `radiation_mj_m2` does not use `min_hours`. It is the sum of the available values on days with at least `min_hours` hourly records, and `NA` otherwise; daytime gaps make it an underestimate.

The function does not itself check data quality, but values flagged by `inmet_qc()` (non-missing `<variable>_flag` columns) are left out of the daily statistics.

Value

A data frame with one row per station and day, sorted by station and date, containing the station identifiers, `date`, `time_zone`, `n_hours` (number of hourly records for that day) and the daily variables that can be computed from the columns present:

Column	Definition
temp_max_c	Maximum of temp_max_prev_c (or of temp_dry_c if absent)
temp_min_c	Minimum of temp_min_prev_c (or of temp_dry_c if absent)
temp_mean_c	Mean of temp_dry_c
temp_dew_mean_c	Mean of temp_dew_c
humid_rel_mean_pct	Mean of humid_rel_pct
humid_rel_min_pct	Minimum of humid_rel_min_pct (or of humid_rel_pct)
precip_mm	Sum of precip_mm
pressure_station_mean_hpa	Mean of pressure_station_hpa
wind_speed_mean_ms	Mean of wind_speed_ms
wind_gust_max_ms	Maximum of wind_gust_ms
radiation_mj_m2	Sum of positive radiation_kjm2, in MJ/m^2

Examples

```
daily <- inmet_daily(rmet_example)
head(daily[, c("date", "n_hours", "temp_max_c", "temp_min_c", "precip_mm")])

# after quality control, flagged hourly values are ignored
daily_qc <- inmet_daily(inmet_qc(rmet_example))
```

---

inmet_derive *Derived hourly variables*

Description

Adds commonly needed variables computed from the hourly measurements: vapour pressure, vapour pressure deficit, wind components and heat index. Values flagged by inmet_qc() are ignored.

Usage

```
inmet_derive(
  x,
  variables = c("vapor_pressure_kpa", "vpd_kpa", "wind_u_ms", "wind_v_ms",
    "heat_index_c")
)
```

Arguments

x	A data frame as returned by inmet_read().
variables	Which variables to add; any of "vapor_pressure_kpa", "vpd_kpa", "wind_u_ms", "wind_v_ms" and "heat_index_c". Variables that cannot be computed from the columns of x are skipped with a message.

Value

x with the new columns.

Variables

Column	Definition
vapor_pressure_kpa	Actual vapour pressure: saturation pressure at the dew point, or at the air temperature times relative
vpd_kpa	Vapour pressure deficit: saturation pressure at the air temperature minus vapor_pressure_kpa (at l
wind_u_ms, wind_v_ms	Eastward and northward wind components, with the meteorological convention that the direction is
heat_index_c	Apparent temperature from the US National Weather Service formula (Rothfusz regression with the

Saturation vapour pressure uses the FAO-56 formula `0.6108 * exp(17.27 * T / (T + 237.3))`.

See Also

inmet_et0(), inmet_daily()

Examples

```
hourly <- inmet_derive(rmet_example)
summary(hourly$vpd_kpa)
```

---

inmet_design_et0 *Design reference evapotranspiration*

Description

Computes the reference evapotranspiration that is exceeded only on a given fraction of days, a common design value for irrigation systems.

Usage

```
inmet_design_et0(x, percentile = 0.8, by = NULL)
```

Arguments

x	A data frame with et0_mm, such as the output of inmet_et0().
percentile	Percentile of the daily values, between 0 and 1. The default 0.8 is the value not exceeded on 80 % of the days.
by	Optional names of columns that define groups, for example "station_code". The default is a single group.

Value

A data frame with one row per group, the grouping columns, n_days (days with a value) and et0_design_mm.

See Also

inmet_et0()

Examples

```
daily <- inmet_et0(inmet_daily(rmet_example))
inmet_design_et0(daily, by = "station_code")
```

`inmet_download` *Download INMET historical data archives*

Description

Downloads annual ZIP archives (one per year, containing every automatic station) from INMET's historical data portal into a local cache.

Usage

```
inmet_download(
  years = .current_year(),
  dest_dir = inmet_cache_dir(),
  max_tries = 15L,
  quiet = FALSE,
  force = FALSE,
  check_updates = TRUE
)
```

Arguments

`years`	Integer vector of years to download. Archives start in 2000. Defaults to the current year.
`dest_dir`	Cache directory, `inmet_cache_dir()` by default. It is created when necessary.
`max_tries`	Maximum attempts per year.
`quiet`	Suppress progress messages.
`force`	Download again even when a valid archive is cached.
`check_updates`	For cached archives of the current and previous year, ask the server whether a newer version exists and download it if so. INMET keeps appending data to these archives. Older years are never re-checked. Requires a (very small) request per recent year.

Details

Archives are written to `<year>.zip.part` and only renamed to `<year>.zip` after their size has been checked against the size reported by the server and the ZIP structure has been validated, so a partial file is never mistaken for a complete one. Interrupted downloads resume from where they stopped on the next call. A small `<year>.zip.meta` file records the server's version information.

The INMET server sends one archive at a time per client, so years are downloaded sequentially. Typical archives for recent years are 90 to 110 MB.

Value

Invisibly, a named character vector with the paths of the archives available after the call.

See Also

`inmet_get()`, `inmet_read()`, `inmet_cache_status()`

Examples

```
paths <- inmet_download(2023, dest_dir = file.path(tempdir(), "rmet"))
```

---

inmet_et0 *Reference evapotranspiration*

Description

Adds the daily grass reference evapotranspiration `et0_mm` to the output of `inmet_daily()`. The default is the FAO Irrigation and Drainage Paper 56 Penman-Monteith equation; the Hargreaves-Samani equation needs only temperature.

Usage

```
inmet_et0(
  x,
  method = c("penman-monteith", "hargreaves"),
  estimate_radiation = FALSE,
  coastal = FALSE
)
```

Arguments

`x`	A data frame as returned by `inmet_daily()`. All methods need `date`, `latitude`, `temp_max_c` and `temp_min_c`. Penman-Monteith also needs `elevation`, `wind_speed_mean_ms`, either `temp_dew_mean_c` or `humid_rel_mean_pct`, and `radiation_mj_m2` (unless `estimate_radiation = TRUE`).
`method`	`"penman-monteith"` (default) or `"hargreaves"`.
`estimate_radiation`	For Penman-Monteith: use the radiation estimated from the daily temperature range (Hargreaves) on days when the measured `radiation_mj_m2` is missing, or in every day when the column is absent.
`coastal`	Use the coastal coefficient (0.19) instead of the interior one (0.16) when estimating radiation.

Details

Penman-Monteith follows FAO-56: mean temperature as the average of the daily extremes, actual vapour pressure from the mean dew point (or, if it is not available, from the mean relative humidity), net radiation from the measured global radiation with the clear-sky radiation of the station's latitude and elevation, no soil heat flux, and pressure from the elevation. INMET wind sensors are at 10 m, so wind speed is converted to 2 m with the logarithmic profile.

Hargreaves-Samani is `0.0023 * (Tmean + 17.8) * sqrt(Tmax - Tmin) * Ra` with the extraterrestrial radiation `Ra` in mm/day. It is less accurate but does not depend on radiation, wind or humidity, which are often missing.

Daily radiation is the sum of the hourly values, and so is an underestimate on days with missing daytime hours; use the `min_hours` rule of `inmet_daily()` and check the result before using it in applications such as irrigation scheduling.

Value

`x` with the new column `et0_mm` (mm/day); `NA` on days when any input is missing.

See Also

`inmet_daily()`, `inmet_monthly()`, `inmet_design_et0()`

Examples

```
daily <- inmet_daily(rmet_example)
daily <- inmet_et0(daily)
summary(daily$et0_mm)

# temperature-only alternative
hs <- inmet_et0(daily, method = "hargreaves")
```

---

inmet_extract *Extract INMET archives*

Description

Copies the original station CSV files out of the annual archives, for use with other software.

Usage

```
inmet_extract(
  years,
  dest_dir = inmet_cache_dir(),
  out_dir = file.path(dest_dir, "csv"),
  stations = NULL,
  overwrite = FALSE,
  quiet = FALSE
)
```

Arguments

years	Integer years to extract.
dest_dir	Directory containing annual ZIP files.
out_dir	Destination directory.
stations	Optional station codes; only their files are extracted.
overwrite	Overwrite existing files.
quiet	Suppress progress messages.

Value

Invisibly, the paths of the extracted files.

Examples

```
cache <- file.path(tempdir(), "rmet")
inmet_download(2023, dest_dir = cache)
inmet_extract(2023, dest_dir = cache, out_dir = file.path(tempdir(), "csv"),
              stations = "A801")
```

---

inmet_fill *Fill short gaps in hourly or daily data*

Description

Fills missing values by linear interpolation across short gaps and, for what remains, with the typical value for that day of the year (or hour of the day). Every filled value is marked in a `<variable>_filled` column, so the result can always be traced back to measurements.

Usage

```
inmet_fill(
  x,
  variables = NULL,
  max_gap = 3L,
  method = c("both", "linear", "climatology"),
  min_years = 3L
)
```

Arguments

x	A data frame with a `datetime` column (hourly data from `inmet_read()`) or a `date` column (daily data from `inmet_daily()`).
variables	Measurement columns to fill. By default every numeric measurement column except precipitation, wind direction and radiation. Precipitation and wind direction cannot be filled: interpolating rain invents events, and directions are circular.

max_gap	Longest gap, in hours (hourly data) or days (daily data), that is filled by interpolation. A gap is the number of consecutive missing steps, including absent rows, between two valid values.
method	"both" (default) interpolates and then uses the typical value; "linear" and "climatology" use only one of them.
min_years	For the typical value: minimum number of valid values of the station for that day of the year (or hour of the day) that are needed to compute the mean.

Details

Each station is filled on its own. The typical value is the mean of the valid values of the same station on the same month and day (daily data) or month, day and UTC hour (hourly data), across all years in x; it needs several years of data to be meaningful, and it smooths out extreme days. Check how much was filled before an analysis: table(filled$temp_dry_c_filled, useNA = "ifany").

Value

x with the missing values filled and one <variable>_filled column per filled variable, holding "linear", "climatology" or NA. Rows that are absent from x are not created.

See Also

inmet_completeness(), inmet_qc()

Examples

```
hourly <- rmet_example
hourly$temp_dry_c[100:102] <- NA
filled <- inmet_fill(hourly, "temp_dry_c", method = "linear")
table(filled$temp_dry_c_filled, useNA = "ifany")
```

---

inmet_get *Download and read INMET data in one call*

Description

The recommended high-level interface: downloads missing annual archives with inmet_download(), reuses valid cached files, and returns the result of inmet_read().

Usage

```
inmet_get(
  years,
  stations = NULL,
  dest_dir = inmet_cache_dir(),
  tz = "UTC",
  variables = NULL,
  start_date = NULL,
```

```
  end_date = NULL,
  drop_empty = TRUE,
  max_tries = 15L,
  force = FALSE,
  check_updates = TRUE,
  quiet = FALSE
)
```

Arguments

years	Integer vector of years.
stations	Optional station codes, such as "A801". Matching is case-insensitive. Use inmet_stations() or inmet_nearest() to find codes.
dest_dir	Directory containing <year>.zip archives.
tz	Output time zone. Defaults to "UTC", the time standard used in the source files. Any name in OlsonNames() is accepted; only the display of the timestamps changes, not the instants they represent.
variables	Optional measurement columns to retain (see *Columns*). Identifier columns are always kept.
start_date, end_date	Optional inclusive date limits in YYYY-MM-DD format, interpreted in tz.
drop_empty	Drop hours in which every measurement is missing (default). INMET files contain such rows for periods when a station was not reporting. Use FALSE to keep them, for example to obtain a regular hourly grid.
max_tries	Maximum attempts per year.
force	Download again even when a valid archive is cached.
check_updates	For cached archives of the current and previous year, ask the server whether a newer version exists and download it if so. INMET keeps appending data to these archives. Older years are never re-checked. Requires a (very small) request per recent year.
quiet	Suppress progress messages.

Value

A data frame as documented in inmet_read().

See Also

inmet_daily(), inmet_stations(), inmet_nearest()

Examples

```
weather <- inmet_get(
  years = 2023,
  stations = "A801",
  variables = c("temp_dry_c", "precip_mm"),
  dest_dir = file.path(tempdir(), "rmet")
```

```
)
daily <- inmet_daily(weather)
```

`inmet_indices` *Climate indices from daily data*

Description

Computes yearly or monthly climate indices from the output of `inmet_daily()`, in the spirit of the indices of the Expert Team on Climate Change Detection and Indices (ETCCDI), plus growing degree days.

Usage

```
inmet_indices(x, by = c("year", "month"), min_valid = 0.8, base_temp = 10)
```

Arguments

`x`	A data frame as returned by `inmet_daily()`, with `date` and at least `precip_mm` or `temp_max_c` and `temp_min_c`.
`by`	Period of each index: `"year"` (default) or `"month"`, calendar years or months in local time.
`min_valid`	Minimum fraction of the days of the period that must have a value for an index to be computed; otherwise it is `NA`. Days absent from `x` count as missing.
`base_temp`	Base temperature (degC) of the growing degree days.

Details

Runs (`cdd`, `cwd`) are broken by missing days. `rx5day` uses five consecutive calendar days without missing values.

Value

A data frame with one row per station and period: the station identifiers, `period` (first day of the period), `n_days` (rows present) and the indices that can be computed:

Column	Definition
`txx`, `tnn`	Highest daily maximum and lowest daily minimum temperature (degC)
`dtr`	Mean daily temperature range (degC)
`summer_days`	Days with maximum temperature above 25 degC
`tropical_nights`	Days with minimum temperature above 20 degC
`frost_days`	Days with minimum temperature below 0 degC
`gdd`	Growing degree days: sum of mean temperature above `base_temp` (degC day)
`prcptot`	Total precipitation on wet days (>= 1 mm)
`rain_days`, `r10mm`, `r20mm`	Days with at least 1, 10 and 20 mm
`sdii`	Mean precipitation on wet days (mm/day)

rx1day, rx5day	Highest 1-day and 5-day precipitation
cdd, cwd	Longest run of dry (< 1 mm) and wet (>= 1 mm) days

See Also

inmet_daily(), inmet_spi()

Examples

```
inmet_indices(inmet_daily(rmet_example))
```

---

inmet_interpolate *Interpolate station values to other locations*

Description

Estimates a variable at any location (for example, the centroids of municipalities or the points of a grid) from the values observed at the stations, by inverse-distance weighting of the nearest stations.

Usage

```
inmet_interpolate(
  x,
  targets,
  variable,
  power = 2,
  k = 5L,
  max_distance_km = Inf
)
```

Arguments

x	A data frame with one row per station: latitude, longitude and the column named in variable, such as monthly or daily values of a single period. Stations with a missing value are ignored.
targets	A data frame with latitude and longitude columns.
variable	Name of the numeric column of x to interpolate.
power	Exponent of the distance in the weights (2 by default).
k	Number of nearest stations used for each target.
max_distance_km	Stations farther than this from a target are not used; targets without any station in range get NA.

Details

A target that coincides with a station gets that station's value. Inverse distance weighting is simple and makes no assumption about the spatial structure, but it cannot extrapolate beyond the observed values, ignores elevation, and is poor for rainfall in sparse networks; use it for a first estimate or for filling gaps, not for precise surfaces.

Value

`targets` with the interpolated column (named as `variable`), `n_stations` (stations used) and `nearest_km` (distance to the closest one).

See Also

`inmet_nearest()`, `inmet_monthly()`

Examples

```
stations <- inmet_stations(state = "RS", source = "bundled")
stations$elevation_obs <- stations$elevation
targets <- data.frame(latitude = c(-30, -29), longitude = c(-51, -53))
inmet_interpolate(stations, targets, "elevation_obs", k = 3)
```

---

inmet_monthly *Monthly summaries of daily INMET data*

Description

Aggregates the output of `inmet_daily()` into one row per station and calendar month, with an explicit rule for incomplete months.

Usage

```
inmet_monthly(x, min_days = 20L)
```

Arguments

`x`	A data frame as returned by `inmet_daily()` (with a `date` column), optionally with `et0_mm` from `inmet_et0()`.
`min_days`	Minimum number of days with a valid value that a variable needs in a month; otherwise its monthly value is `NA`. Missing daily values (for example, days left out by the `min_hours` rule of `inmet_daily()`) do not count.

Details

Months are the calendar months of the `date` column, which `inmet_daily()` builds in each station's local time. Sums (`precip_mm`, `rain_days`, `et0_mm`) are not scaled up for missing days, so they are only comparable between months when `min_days` is close to the month length. Use `inmet_completeness()` to see how complete the hourly data are.

Value

A data frame with one row per station and month, sorted by station and month, with the station identifiers, `month` (the first day of the month, as a `Date`), `n_days` (days present in `x`) and the monthly variables that can be computed from the columns present:

Column	Definition
`temp_max_mean_c`, `temp_min_mean_c`	Mean of the daily maximum and minimum temperature
`temp_mean_c`	Mean of the daily mean temperature
`temp_max_abs_c`, `temp_min_abs_c`	Highest daily maximum and lowest daily minimum
`humid_rel_mean_pct`	Mean of the daily mean relative humidity
`precip_mm`	Sum of the daily precipitation
`rain_days`	Number of days with at least 1 mm of precipitation
`wind_speed_mean_ms`	Mean of the daily mean wind speed
`wind_gust_max_ms`	Highest daily gust
`radiation_mean_mj_m2`	Mean daily global radiation
`et0_mm`	Sum of the daily reference evapotranspiration

See Also

`inmet_daily()`, `inmet_et0()`

Examples

```
daily <- inmet_daily(rmet_example)
monthly <- inmet_monthly(daily)
monthly[, c("month", "n_days", "temp_mean_c", "precip_mm", "rain_days")]
```

`inmet_nearest` *Find the stations closest to locations*

Description

Ranks automatic stations by great-circle distance from one or many points, for example the centroids of municipalities.

Usage

```
inmet_nearest(latitude, longitude, n = 5L, ...)
```

Arguments

`latitude`, `longitude`	Coordinates in decimal degrees (south and west are negative): single numbers, or vectors of the same length for many locations.
`n`	Number of stations to return for each location.
`...`	Further arguments passed to `inmet_stations()`, for example `status = "Operante"` or `source = "bundled"`.

Value

The rows of `inmet_stations()` for the `n` nearest stations, with an added `distance_km` column, closest first. With several locations the rows of all of them are stacked, with an added first column `location` (the position of the point in `latitude`) and `rank`.

See Also

`inmet_stations_within()`, `inmet_interpolate()`

Examples

```
# Stations near Porto Alegre, from the bundled catalogue
inmet_nearest(-30.03, -51.23, n = 3, source = "bundled")

# several locations at once
inmet_nearest(c(-30.03, -23.55), c(-51.23, -46.63), n = 2, source = "bundled")
```

---

inmet_normals *Climatological normals*

Description

Computes the mean of each calendar month over the years available for each station, from the output of `inmet_monthly()`.

Usage

```
inmet_normals(x, variables = NULL, years = NULL, min_years = 10L)
```

Arguments

`x`	A data frame as returned by `inmet_monthly()`.
`variables`	Numeric columns to average. Default: all numeric measurement columns (everything except identifiers and counts).
`years`	Optional integer vector of years to use as the reference period, such as `1991:2020`. Default: all years in `x`.
`min_years`	Minimum number of years with a value that a station needs for a calendar month; otherwise the normal is `NA`. WMO uses 30 years for official normals; the default is lower so that shorter series give something, but check `n_years`.

Value

A data frame with one row per station and calendar month: `station_code`, `month_of_year` (1 to 12), `n_years` (the largest number of years behind any variable) and the mean of each variable.

See Also

`inmet_anomaly()`, `inmet_monthly()`

Examples

```
normals <- inmet_normals(inmet_monthly(inmet_daily(rmet_example)), min_years = 1)
normals[, c("month_of_year", "temp_mean_c", "precip_mm")]
```

`inmet_pet_thornthwaite` *Potential evapotranspiration by Thornthwaite*

Description

Adds the monthly potential evapotranspiration `pet_mm` to the output of `inmet_monthly()`, from air temperature alone, with the method of Thornthwaite (1948) corrected for the day length and the number of days of each month.

Usage

```
inmet_pet_thornthwaite(x)
```

Arguments

`x`	A data frame as returned by `inmet_monthly()`, with `month`, `temp_mean_c` and `latitude`.

Details

The annual heat index of each station is computed from its mean temperature of each calendar month over all the years in `x`, so `x` must have all 12 calendar months for the station; otherwise `pet_mm` is `NA` for it. Stations with several years give a stable index; with a single year the index is that of the year. Thornthwaite's method underestimates in humid tropical regions and overestimates in dry ones, and is best for monthly or longer periods; for daily needs see `inmet_et0()`.

Value

`x` with the new column `pet_mm` (mm per month).

See Also

`inmet_water_balance()`, `inmet_et0()`

Examples

```
monthly <- inmet_monthly(inmet_daily(rmet_example))
monthly <- inmet_pet_thornthwaite(monthly)
monthly[, c("month", "temp_mean_c", "pet_mm")]
```

`inmet_qc` *Quality control of hourly INMET data*

Description

Flags implausible hourly values with simple, transparent checks. INMET publishes automatic-station data without consistency validation; these checks catch the most common errors (impossible values, sensor spikes, stuck sensors and internally inconsistent records) but are no substitute for checks tailored to an analysis.

Usage

```
inmet_qc(
  x,
  action = c("flag", "remove"),
  checks = c("range", "consistency", "spike", "persistence"),
  persistence_hours = 12L,
  quiet = FALSE
)
```

Arguments

`x`	A data frame as returned by `inmet_read()`.
`action`	`"flag"` (default) adds a `<variable>_flag` column for each checked variable, holding the names of the failed checks (separated by commas) or `NA`. `"remove"` sets flagged values to `NA` instead.
`checks`	Checks to run; any of `"range"`, `"consistency"`, `"spike"`, and `"persistence"` (see *Checks*).
`persistence_hours`	Minimum number of consecutive hours with an identical value for the persistence check.
`quiet`	Suppress the summary message.

Value

`x` with flag columns added (or flagged values removed). The attribute `qc_summary` is a data frame with the number and percentage of flagged values per variable and check.

Checks

- **range**: outside physical limits for Brazil, e.g. air temperature outside -15 to 48 degC, relative humidity outside 1 to 100 %, hourly precipitation outside 0 to 150 mm, station pressure outside 650 to 1060 hPa, or wind direction outside 0 to 360 degrees.
- **consistency**: hourly maximum below minimum (temperature, humidity, pressure; both flagged); air temperature outside the extremes of the hour it ends; dew point above air temperature; gust below mean wind speed. A tolerance of 0.5 units allows for rounding.

- **spike**: a value that departs from *both* adjacent hours in the same direction by more than 8 degC (air and dew-point temperature), 5 hPa (pressure) or 45 % (relative humidity). Changes that persist, such as a cold front, are not flagged.
- **persistence**: the same value in at least `persistence_hours` consecutive hours, for temperatures, humidity, pressure, wind and precipitation. Zero wind speed, zero precipitation, humidity of 90 % or more (fog, persistent rain, sensors that saturate below 100 %), and wind direction during calm hours (speed below 0.5 m/s) are not checked.

Neighbouring hours are only compared within a station and when exactly one hour apart.

See Also

`inmet_daily()`, `inmet_completeness()`

Examples

```
checked <- inmet_qc(rmet_example)
attr(checked, "qc_summary")
table(checked$temp_dry_c_flag, useNA = "ifany")

# daily values ignore flagged hours
daily <- inmet_daily(checked)
```

---

inmet_read *Read cached INMET historical data*

Description

Parses the station CSV files inside annual INMET archives (downloaded with `inmet_download()`) into one data frame. Files are read directly from the ZIP archives; nothing is extracted to disk.

Usage

```
inmet_read(
  years,
  stations = NULL,
  dest_dir = inmet_cache_dir(),
  tz = "UTC",
  variables = NULL,
  start_date = NULL,
  end_date = NULL,
  drop_empty = TRUE,
  quiet = FALSE
)
```

Arguments

years	Integer vector of years.
stations	Optional station codes, such as "A801". Matching is case-insensitive. Use inmet_stations() or inmet_nearest() to find codes.
dest_dir	Directory containing <year>.zip archives.
tz	Output time zone. Defaults to "UTC", the time standard used in the source files. Any name in OlsonNames() is accepted; only the display of the timestamps changes, not the instants they represent.
variables	Optional measurement columns to retain (see *Columns*). Identifier columns are always kept.
start_date, end_date	Optional inclusive date limits in YYYY-MM-DD format, interpreted in tz.
drop_empty	Drop hours in which every measurement is missing (default). INMET files contain such rows for periods when a station was not reporting. Use FALSE to keep them, for example to obtain a regular hourly grid.
quiet	Suppress progress messages.

Value

A data frame with one row per station and hour, sorted by station code and time. Identifier columns come first: station_code, station_name, region, state, latitude, longitude, elevation (from each file's header) and datetime (POSIXct). Measurement columns are numeric; missing values are NA.

Repeated files

The archive of the current year contains several cumulative versions of each station file (for example, January to August and January to September). Each station and hour is returned once, taken from the most recent version that contains it.

Timestamps

Each row is stamped with the *end* of the hour it describes, in UTC. Values labelled "previous hour" (such as temp_max_prev_c and precip_mm) summarise the hour ending at datetime. Use inmet_daily() to build daily summaries in local time.

Columns

Column	Description	Unit
precip_mm	Precipitation in the previous hour	mm
pressure_station_hpa	Station-level pressure	hPa
pressure_max_hpa, pressure_min_hpa	Pressure extremes, previous hour	hPa
radiation_kjm2	Global radiation, previous hour	kJ/m^2
temp_dry_c	Air (dry-bulb) temperature	degC
temp_dew_c	Dew-point temperature	degC
temp_max_prev_c, temp_min_prev_c	Air temperature extremes, previous hour	degC

temp_dew_max_c, temp_dew_min_c	Dew-point extremes, previous hour	degC
humid_rel_pct	Relative humidity	%
humid_rel_max_pct, humid_rel_min_pct	Humidity extremes, previous hour	%
wind_dir_deg	Wind direction	degrees
wind_gust_ms	Maximum gust	m/s
wind_speed_ms	Wind speed	m/s

Unrecognised source columns are kept under a snake_case version of their original title.

See Also

inmet_get(), inmet_download(), inmet_daily(), inmet_stations()

Examples

```
cache <- file.path(tempdir(), "rmet")
inmet_download(2023, dest_dir = cache)
poa <- inmet_read(2023, stations = "A801", dest_dir = cache)
```

---

inmet_spi *Standardized Precipitation Index*

Description

Computes the Standardized Precipitation Index (SPI) of McKee et al. (1993), a measure of how wet or dry an accumulation of months is compared with the station's own history, from the output of inmet_monthly().

Usage

```
inmet_spi(x, scale = 3L, min_years = 20L)
```

Arguments

x	A data frame as returned by inmet_monthly(), with month and precip_mm.
scale	Number of months accumulated (1, 3, 6, 12, ...).
min_years	Minimum number of accumulations of the same calendar month that a station needs; otherwise its SPI is NA. WMO recommends 30 years.

Details

For each station and calendar month, a gamma distribution is fitted to the accumulated precipitation by maximum likelihood (Thom's approximation), with the probability of zero accumulation treated separately, and the cumulative probability is transformed to a standard normal variable. Accumulations with any missing month are NA; months absent from x count as missing.

Value

x with the new column spi (and the attribute spi_scale). Values near 0 are normal, below -1 are dry and above 1 are wet.

See Also

inmet_monthly(), inmet_indices()

Examples

```
monthly <- inmet_monthly(inmet_daily(rmet_example))
# one year is too short for a meaningful SPI: use min_years = 1 only to see the output
inmet_spi(monthly, scale = 3, min_years = 1)[, c("month", "precip_mm", "spi")]
```

---

inmet_stations *Search the INMET automatic-station catalogue*

Description

Returns automatic weather stations published by INMET. By default, the function uses a cached copy of the official API response and refreshes stale caches periodically. If the service is unavailable, it falls back to an older cache or to a snapshot of the official catalogue bundled with the package (its date is stored in the snapshot_date attribute).

Usage

```
inmet_stations(
  region = NULL,
  state = NULL,
  search = NULL,
  status = NULL,
  entity = NULL,
  source = c("auto", "remote", "bundled"),
  refresh = FALSE,
  max_age = 30,
  cache_dir = inmet_cache_dir(),
  quiet = FALSE
)
```

Arguments

region, state	Optional region or state abbreviations.
search	Optional literal text contained in the station name or code.
status	Optional operational status, such as "Operante" or "Pane".
entity	Optional station owner/provider, such as "INMET".

source	One of "auto", "remote", or "bundled". "auto" uses the cache, then the official API, then the bundled fallback.
refresh	Force a fresh request to the official API.
max_age	Maximum cache age in days before an automatic refresh.
cache_dir	Directory used for the catalogue cache.
quiet	Suppress informational messages.

Value

A data frame with station code, name, region, state, coordinates, altitude, operational dates (UTC calendar dates) and status, provider metadata, and the local time_zone (see inmet_timezone()). The legacy start_year column is retained for compatibility. The attribute catalog_source records where the data came from ("cache", "INMET API", "stale cache", or "bundled fallback").

See Also

inmet_nearest(), inmet_get(), inmet_read()

Examples

```
# Offline, reproducible example using the bundled fallback
inmet_stations(state = "RS", source = "bundled")


# Current official catalogue (downloaded once and then cached)
inmet_stations(state = "RS", refresh = TRUE, cache_dir = tempdir())
```

---

inmet_stations_within *Stations inside an area*

Description

Selects the stations whose coordinates fall inside a bounding box or a polygon.

Usage

```
inmet_stations_within(area, ...)
```

Arguments

area	Either a bounding box, a numeric vector c(west, south, east, north) in decimal degrees, or an sf/sfc polygon object (needs the suggested package sf). Polygons in another coordinate system are transformed to WGS 84.
...	Further arguments passed to inmet_stations(), for example status = "Operante" or source = "bundled".

Value

The rows of `inmet_stations()` inside `area`.

See Also

`inmet_nearest()`, `inmet_as_sf()`

Examples

```
# stations in a box around Porto Alegre
inmet_stations_within(c(-52, -31, -50, -29), source = "bundled")
```

`inmet_timezone` *Local time zone of Brazilian stations*

Description

Returns the Olson time-zone name for each station from its state and, where a state spans more than one historical zone, its longitude. The zones carry Brazil's full history of offsets and daylight saving time, so they give correct local times for any year since 2000.

Usage

```
inmet_timezone(state, longitude = NA_real_)
```

Arguments

`state`	Two-letter state abbreviations, such as `"RS"`.
`longitude`	Station longitudes in decimal degrees. Only used for Amazonas (west of 67.2 degrees W is UTC-5), Para (west of 52.5 degrees W was UTC-4 until 2008), and Pernambuco (Fernando de Noronha).

Details

Zone boundaries inside Amazonas and Para follow municipal borders; the longitude limits reproduce them for INMET's stations but are an approximation elsewhere.

Value

A character vector of time-zone names, `NA` for unknown states.

Examples

```
inmet_timezone(c("RS", "MT", "AC", "AM", "AM"),
               c(-51.2, -56.1, -67.8, -60.0, -69.9))
```

`inmet_water_balance` *Climatological water balance*

Description

Sequential soil water balance of Thornthwaite and Mather (1955) for each station, from precipitation and potential evapotranspiration.

Usage

```
inmet_water_balance(x, awc = 100, pet = "pet_mm", initial = awc)
```

Arguments

`x`	A data frame in time order per station, with `precip_mm` and the column named in `pet`; typically the output of `inmet_monthly()` with `inmet_pet_thornthwaite()` or `inmet_et0()` already applied.
`awc`	Available water capacity of the soil, in mm.
`pet`	Name of the column with potential or reference evapotranspiration for the same period as `precip_mm`.
`initial`	Soil water storage at the start of each station's series, in mm. Default: `awc` (soil at capacity).

Details

When precipitation is below the evapotranspiration, the soil dries exponentially: `storage = previous * exp((P - PET) / awc)`; otherwise it recharges up to `awc` and the rest is excess. The first periods depend on `initial`, so use a series of at least a year before the period of interest. The result does not depend on the time step, but the usual choice is monthly.

Value

`x` with the columns `storage_mm` (soil water at the end of the period), `etr_mm` (actual evapotranspiration), `deficit_mm` (potential minus actual evapotranspiration) and `excess_mm` (surplus that percolates or runs off). Periods with missing precipitation or evapotranspiration have `NA` in the four columns and leave the storage unchanged.

See Also

`inmet_pet_thornthwaite()`, `inmet_et0()`

Examples

```
monthly <- inmet_pet_thornthwaite(inmet_monthly(inmet_daily(rmet_example)))
balance <- inmet_water_balance(monthly, awc = 100)
balance[, c("month", "precip_mm", "pet_mm", "storage_mm", "deficit_mm", "excess_mm")]
```

`rmet-classes` *Classes of rmet data*

Description

The data frames returned by `inmet_read()`, `inmet_get()`, `inmet_daily()` and `inmet_monthly()` have the classes `rmet_hourly`, `rmet_daily` and `rmet_monthly`, which all inherit from `rmet_data` and `data.frame`. They behave as ordinary data frames in every respect (indexing, `merge()`, `dplyr`, `as.data.frame()`), and add a compact `print()` and a `summary()`.

Usage

```
## S3 method for class 'rmet_data'
print(x, ..., n = 10L)

## S3 method for class 'rmet_data'
summary(object, ...)

## S3 method for class 'summary.rmet_data'
print(x, ...)

## S3 method for class 'rmet_data'
as.data.frame(x, row.names = NULL, optional = FALSE, ...)

## S3 method for class 'rmet_data'
plot(
  x,
  y = NULL,
  variable = NULL,
  station = NULL,
  type = "l",
  xlab = "",
  ylab = NULL,
  ...
)
```

Arguments

`x`, `object`	An object returned by one of the functions above.
`...`	Further arguments, ignored.
`n`	Number of rows to print.
`row.names`, `optional`	Passed on to `as.data.frame()`.
`y`, `type`, `xlab`, `ylab`	For `plot()`: ignored `y`, and graphical arguments passed on; `...` goes to `graphics::plot()`.

variable	For plot(): name of the numeric column to draw. Default: the first of air temperature, mean temperature, precipitation, or the first measurement column.
station	For plot(): station codes to draw. Default: all (at most eight are drawn when this is not given).

Value

print() returns x invisibly. summary() returns a data frame with one row per station (class summary.rmet_data): number of rows, first and last time or date, and the percentage of non-missing values in the measurement columns. as.data.frame() removes the rmet classes.

Examples

```
class(rmet_example)
rmet_example
summary(rmet_example)
```

---

rmet_example *Hourly weather data from INMET station A801 (Porto Alegre), 2023*

Description

One year of hourly observations from the automatic station A801 (Porto Alegre - Jardim Botanico, RS), exactly as returned by inmet_get(2023, stations = "A801"). Timestamps are in UTC. Hours with no measurements at all are not included.

Usage

```
rmet_example
```

Format

An rmet_hourly data frame (see rmet-classes) with 8,663 rows and 25 columns. See inmet_read() for the description of each column.

Source

INMET (Instituto Nacional de Meteorologia), https://portal.inmet.gov.br/dadoshistoricos. Generated by data-raw/rmet_example.R.

Examples

```
range(rmet_example$datetime)
head(inmet_daily(rmet_example))
```

Index

∗ **datasets**
rmet_example, 33

as.data.frame(), *32*
as.data.frame.rmet_data *(*rmet-classes*)*, 32

graphics::plot(), *32*

inmet_anomaly, 3
inmet_anomaly(), *22*
inmet_as_sf, 4
inmet_as_sf(), *30*
inmet_availability, 4
inmet_cache_clear, 5
inmet_cache_clear(), *6*
inmet_cache_dir, 6
inmet_cache_dir(), *5*, *7*, *12*
inmet_cache_status, 7
inmet_cache_status(), *5*, *6*, *13*
inmet_completeness, 7
inmet_completeness(), *5*, *16*, *20*, *25*
inmet_convert_radiation, 8
inmet_daily, 9
inmet_daily(), *4*, *8*, *11*, *13–15*, *17–21*, *25–27*, *32*
inmet_derive, 10
inmet_design_et0, 11
inmet_design_et0(), *14*
inmet_download, 12
inmet_download(), *5*, *7*, *16*, *25*, *27*
inmet_et0, 13
inmet_et0(), *11*, *20*, *21*, *23*, *31*
inmet_extract, 14
inmet_fill, 15
inmet_get, 16
inmet_get(), *9*, *13*, *27*, *29*, *32*
inmet_indices, 18
inmet_indices(), *28*
inmet_interpolate, 19
inmet_interpolate(), *22*
inmet_monthly, 20
inmet_monthly(), *3*, *14*, *20*, *22*, *23*, *27*, *28*, *31*, *32*
inmet_nearest, 21
inmet_nearest(), *4*, *17*, *20*, *26*, *29*, *30*
inmet_normals, 22
inmet_normals(), *3*
inmet_pet_thornthwaite, 23
inmet_pet_thornthwaite(), *31*
inmet_qc, 24
inmet_qc(), *8–10*, *16*
inmet_read, 25
inmet_read(), *4*, *7–10*, *13*, *15–17*, *24*, *29*, *32*, *33*
inmet_spi, 27
inmet_spi(), *19*
inmet_stations, 28
inmet_stations(), *4*, *5*, *17*, *21*, *22*, *26*, *27*, *29*, *30*
inmet_stations_within, 29
inmet_stations_within(), *22*
inmet_timezone, 30
inmet_timezone(), *9*, *29*
inmet_water_balance, 31
inmet_water_balance(), *23*

OlsonNames(), *9*, *17*, *26*

plot.rmet_data *(*rmet-classes*)*, 32
print.rmet_data *(*rmet-classes*)*, 32
print.summary.rmet_data *(*rmet-classes*)*, 32

rmet-classes, 32, *33*
rmet_example, 33

summary.rmet_data *(*rmet-classes*)*, 32