Package 'jmastats'

May 8, 2026

Version 0.3.0

Title Download Weather Data from Japan Meteorological Agency Website

Description Provides features that allow users to download weather data published by the Japan Meteorological Agency (JMA) website (<https://www.jma.go.jp/jma/index.html>). The data includes information dating back to 1976 and aligns with the categories available on the website. Additionally, users can process the best track data of typhoons and easily handle earthquake record files.

Depends R (>= 4.1)

License MIT + file LICENSE

BugReports https://github.com/uribo/jmastats/issues

URL https://uribo.github.io/jmastats/, https://github.com/uribo/jmastats

Imports cli (>= 3.4.0), crayon (>= 1.3.4), dplyr (>= 1.1.0), forcats (>= 0.4.0), ggplot2 (>= 2.2.1), lifecycle (>= 1.0.3), lubridate (>= 1.7.4), purrr (>= 1.0.2), rappdirs (>= 0.3.3), readr (>= 1.1.1), rlang (>= 0.2.1), rvest (>= 0.3.2), sf (>= 0.6.3), stringr (>= 1.3.1), tibble (>= 3.0.0), tidyselect (>= 1.1.0), tidyr (>= 1.0.0), units (>= 0.5.1), xml2 (>= 1.2.0)

Suggests knitr (>= 1.20), lwgeom (>= 0.1-4), rmarkdown (>= 1.9), testthat (>= 2.0.0)

Encoding UTF-8

LazyData true

ByteCompile true

RoxygenNote 7.3.2

NeedsCompilation no

Author Shinya Uryu [aut, cph, cre] (ORCID: <https://orcid.org/0000-0002-0493-6186>)

Maintainer Shinya Uryu <suika1127@gmail.com>

Repository CRAN

Date/Publication 2025-01-28 05:00:02 UTC

Contents

`earthquake_station` *Japan Meteorological Agency's earthquake observe stations*

Description

This data corresponds to the July 18, 2024 update.

Usage

```
earthquake_station
```

Format

A simple feature data frame with 671 rows 7 variables

Examples

```
head(earthquake_station)
```

ggplot2-scales *Scales to use for ggplot2*

Description

[Experimental]

Usage

```
scale_color_jma_absolute(type = "precipitation", ...)

scale_color_jma_relative(type = "amedas", ...)

scale_fill_jma_absolute(type = "precipitation", ...)

scale_fill_jma_relative(type = "amedas", ...)
```

Arguments

type	Display item
...	Arguments to pass on to ggplot2::scale_color_gradientn or ggplot2::scale_fill_gradientn

Value

A ScaleContinuous object that can be added to a ggplot object

Examples

```
d <-
data.frame(
  date = as.Date(c(17410, 17411, 17412, 17413, 17414, 17415), origin = "1970-01-01"),
  precipitation_sum = units::set_units(c(3.5, 9.5, 0, 0, 0, 5), "mm"))

library(ggplot2)
library(units)
ggplot(d, aes(date, precipitation_sum,
          color = drop_units(precipitation_sum),
          fill = drop_units(precipitation_sum))) +
  geom_bar(stat = "identity") +
  scale_color_jma_absolute(type = "precipitation") +
  scale_fill_jma_absolute(type = "precipitation")
```

jma_collect *Collect JMA Historical Weather Data*

Description

[Stable]

Refer to the data available in the JMA Historical Weather Data Search. Executed by specifying the target location and date. Currently, not all types of data acquisition are supported.

Usage

```
jma_collect(
  item = NULL,
  block_no,
  year,
  month,
  day,
  cache = TRUE,
  pack = TRUE,
  quiet = FALSE
)
```

Arguments

item	Type of weather data to be acquired. Mainly specifies the interval between records (e.g. daily or hourly). See NOTE for details.
block_no	Block number of the location to be observed. It is assumed that block_no is input as a string consisting of a 4- or 5-digit number. If a numeric value is specified, it is processed as a string.
year	select year
month	select month
day	select date (default NULL)
cache	use cash and save to cache. (TRUE, the default)
pack	Whether to packing common variables or not. (TRUE, the default)
quiet	Whether to output information on variable and row combinations that were treated as missing values for some reason. (TRUE, the default)

Value

a tbl object

Note

The parameter `item` chooses one from these:

- annually: Annual value. Please specify a location by `block_no`.
- monthly: Monthly value. Please specify location and year.
- 3monthly: Value every 3 months. Please specify location and year.
- 10daily: Seasonal value. Please specify location and year.
- mb5daily: Semi-seasonal value. Please specify location and year.
- daily: Daily value. Please specify location, year and month.
- hourly: Hourly value. Please specify location, year, month and day.
- rank: Values of the largest in the history of observations.
- nml_ym: Climatological normals for each year and month.
- nml_3m: Climatological normals for each 3 months.
- nml_10d: Climatological normals for each season (almost 10 days).
- nml_mb5d: Climatological normals for each semi-season (almost 5 days).
- nml_daily: Daily climatological normals for specific month. for each location.

Examples

```
# Annually
jma_collect(item = "annually", "1284", year = 2017, month = 11, cache = FALSE)
# Daily
jma_collect(item = "daily", block_no = "0010", year = 2017, month = 11, cache = FALSE)
jma_collect(item = "daily", "0422", year = 2017, month = 11, cache = FALSE)
# Hourly
jma_collect("hourly", "0010", 2018, 7, 30, cache = FALSE)
# Historical Ranking
jma_collect("rank", block_no = "47646", year = 2020, cache = FALSE)
# Climatological normals
jma_collect("nml_ym", block_no = "47646", cache = FALSE, pack = FALSE)
jma_collect("nml_3m", "47646", cache = FALSE, pack = FALSE, quiet = TRUE)
jma_collect("nml_10d", "0228", cache = FALSE, pack = FALSE, quiet = TRUE)
jma_collect("nml_mb5d", "0228", cache = FALSE, pack = FALSE, quiet = FALSE)
```

---

nearest_station *Find out neighborhood stations*

Description

[Stable]

Return the nearest stations information to the given coordinates.

Usage

```
nearest_station(longitude, latitude, geometry = NULL)

pick_neighbor_stations(
  longitude,
  latitude,
  distance = 1,
  .unit = "m",
  geometry = NULL
)

pick_neighbor_tide_stations(
  year,
  longitude,
  latitude,
  distance = 100,
  .unit = "km",
  geometry = NULL
)
```

Arguments

longitude	Longitude.
latitude	Latitude.
geometry	XY sf::sf object.
distance	Distance from station to station to adjustment.
.unit	Unit used for extraction from the point of interest. Default *m* (meters). This value is passed to units::as_units.
year	For tide level data. Restricted to the observation points in the target year.

Details

- `nearest_station()`: Return single station data.
- `pick_neighbor_stations()`: Pick-up neighbourhood stations.
- `pick_neighbor_tide_stations()`: Pick-up neighbourhood tidal observation stations. Filter by distance from target point.

Value

an object of class `sf`.

Examples

```
nearest_station(142.9313, 43.70417)

pick_neighbor_stations(140.10, 36.08, 300000)
```

```
d <-
  pick_neighbor_stations(140.10, 36.08, 30, "km")
pick_neighbor_stations(geometry = sf::st_point(c(140.1833, 36.23333)),
                       distance = 100)

pick_neighbor_tide_stations(longitude = 133.4375, latitude = 34.45833,
                            year = 2020,
                            distance = 100, .unit = "km")
```

---

parse_unit *Parse data variable units*

Description

[Stable]

Usage

```
parse_unit(data, rename = TRUE)
```

Arguments

data	data
rename	*logical*

Value

a tbl object

Examples

```
# For data retrieved with jma_collect(), here is a minimal example.
d <-
tibble::tibble(date = as.Date(c(17742, 17742, 17742, 17742, 17742, 17742), origin = "1970-01-01"),
               time = c(1, 2, 3, 4, 5, 6),
               `precipitation(mm)` = c(0, 0, 0, 0, 0, 0),
               `temperature(℃)` = c(22.4, 22.1, 21, 20.2, 20.4, 23.5))
d |> parse_unit(rename = TRUE)
```

pivot_tide_level *Convert and split tidal level data*

Description

[Stable]

Usage

```
pivot_tide_level(data)
```

Arguments

data	tidal level data

Value

List to store two datasets containing hourly and tide level data.

See Also

read_tide_level()

Examples

```
read_tide_level(system.file("dummy/tide.txt", package = "jmastats")) |>
  pivot_tide_level()
```

read_eqdb_csv *Read the csv of the earthquake database*

Description

[Stable]

Usage

```
read_eqdb_csv(path, show_metadata = TRUE)
```

Arguments

path	local file path to earthquake record file.
show_metadata	logical. If *FALSE*, returns only the values observed at each location.

Value

a tbl object

See Also

https://www.data.jma.go.jp/svd/eqdb/data/shindo/index.html

Examples

```
read_eqdb_csv(system.file("dummy/eqdb.csv", package = "jmastats"))
```

`read_jma_weather` *Read the past weather*

Description

[Experimental]

Read the past weather data files downloaded from JMA.

Usage

```
read_jma_weather(path)
```

Arguments

`path`	The path to the downloaded file.

Value

a `tbl` object

See Also

https://www.data.jma.go.jp/gmd/risk/obsdl/index.php, https://www.data.jma.go.jp/gmd/risk/obsdl/top/help3.html

Examples

```
read_jma_weather(system.file("dummy/dl_data.csv", package = "jmastats"))
```

`read_kishou_feed` *Read Kishou Disaster Prevention Information Feed*

Description

[Experimental]

Usage

```
read_kishou_feed(frequency, type)
```

Arguments

`frequency`	Release frequency. Select either high frequency ("high") or long term ("low")
`type`	Feed type. Specify the item to be retrieved as a string. See details for the items.

Details

The following items can be specified in the type argument.

- regular: It will be announced on time.
- extra: It will be announced at any time.
- eqvol: Earthquakes and Volcanoes.
- other: Other informations.

Value

a `tbl` object

See Also

https://xml.kishou.go.jp

Examples

```
read_kishou_feed("high", type = "regular")
read_kishou_feed("low", "other")
```

read_rsmc_besttrack *Read RSMC Tokyo-Typhoon Center's best track data*

Description

[Stable]

Tidy formatting best track data and combine each point to line.

Usage

```
read_rsmc_besttrack(path)

track_combine(
  data,
  group_vars = c("international_number", "storm_name"),
  keep_vars = NULL,
  geometry = geometry
)
```

Arguments

path	path to best track data (.txt). Give the path as a directory in the user's computer or the URL.
data	Import data using read_rsmc_besttrack
group_vars	To combine track variables.
keep_vars	Keep variables.
geometry	geometry column name (default geometry).

Details

- read_rsmc_besttrack(): Read single best track data into sf contains observation record as point.
- track_combine(): Combine track data to line by id (such as international_number and storm_name).

Value

a tbl object

See Also

https://www.jma.go.jp/jma/jma-eng/jma-center/rsmc-hp-pub-eg/RSMC_HP.htm

Examples

```
read_rsmc_besttrack(path = system.file("dummy/bst.txt", package = "jmastats"))

read_rsmc_besttrack(path = system.file("dummy/bst.txt", package = "jmastats")) |>
  track_combine()
```

`read_tide_level` *Read and parse tide level text data*

Description

[Stable]

Usage

```
read_tide_level(path = NULL, .year, .month, .stn, raw = FALSE)
```

Arguments

`path`	URL or local file path to sea tide level file
`.year`	A.D. 1997 to present year.
`.month`	Month number. 1997 only, valid after March.
`.stn`	Station identification name in uppercase two-digit letters.
`raw`	If *TRUE*, return raw format data

Value

a `tbl` object

See Also

https://www.data.jma.go.jp/gmd/kaiyou/db/tide/suisan/readme.html

Examples

```
# Read a local storage file (dummy data)
read_tide_level(system.file("dummy/tide.txt", package = "jmastats"))

# Request from URL
read_tide_level("https://www.data.jma.go.jp/gmd/kaiyou/data/db/tide/suisan/txt/2020/TK.txt")
# Request from parameters
read_tide_level(.year = 2020, .month = 2, .stn = "TK")
```

reset_cache *Remove all cache files*

Description

[Experimental]

Remove all package cache files.

Usage

```
reset_cache()
```

Value

None

Examples

```
if (interactive())
  reset_cache()
```

stations *Japan Meteorological Agency's Stations list*

Description

This data corresponds to the April 1, 2024 update.

Usage

```
stations
```

Format

A data frame with 1323 rows 14 variables:

- area
- station_no
- station_type
- station_name
- address
- elevation
- observation_begin
- note1

- note1
- note2
- katakana
- prec_no
- block_no
- pref_code
- geometry

Examples

```
head(stations)
dim(stations)
```

`tide_station` *Tidal observation stations of Japan Meteorological Agency*

Description

Observation stations from 1997 to 2024. This data corresponds to the January 1, 2024 update.

Usage

```
tide_station
```

Format

A data frame with 1949 rows 7 variables

Examples

```
head(tide_station)
```

Index