Package: fifo (via r-universe)

June 9, 2026

Title Extract Australian Agricultural and Ecological Data from Publicly Available Data Set Using GPS Points

Version 1.1.3

Description Extracts point data for a given GPS coordinate that includes soil data, weather data and GRDC agro-ecological zone information at that point in Australia.

License GPL (>= 3)

URL https://codeberg.org/adamhsparks/fifo, https://adamhsparks.codeberg.page/fifo/, https://adamhsparks.r-universe.dev/fifo

BugReports https://codeberg.org/adamhsparks/fifo/issues

Depends R (>= 4.1.0)

Imports brio, cli, data.table, httr2, lifecycle, methods, nert, purrr, read.abares (>= 2.0.0), rlang, sf, stats, terra, utils, weatherOz

Suggests dplyr, knitr, readr, rmarkdown, roxyglobals, spelling, testthat (>= 3.0.0), vcr (>= 0.6.0)

VignetteBuilder knitr

Remotes https://github.com/AAGI-AUS/nert

Config/Needs/build moodymudskipper/devtag

Config/roxygen2/version 8.0.0

Config/roxyglobals/filename generated-globals.R

Config/roxyglobals/unique FALSE

Config/roxylint list(linters = roxylint::tidy)

Config/testthat/edition 3

Config/testthat/parallel true

Encoding UTF-8

Language en-US

NeedsCompilation no

Roxygen list(markdown = TRUE, roclets = c(``collate", ``namespace", ``rd", ``roxyglobals::global_roclet", ``roxylint::roxylint", ``devtag::dev_roclet"))

X-schema.org-applicationCategory Tools

X-schema.org-keywords agriculture, Australia, agroecology, soils, weather, weather-data, soils-data, agriculture-research

Config/pak/sysreqs libabsl-dev cmake libgdal-dev gdal-bin libgeos-dev make libmagick++-dev gsfonts libicu-dev libuv1-dev libxml2-dev libnetcdf-dev libssl-dev libproj-dev libsqlite3-dev libudunits2-dev

Repository https://adamhsparks.r-universe.dev

Date/Publication 2026-06-09 12:33:40 UTC

RemoteUrl https://codeberg.org/adamhsparks/fifo

RemoteRef HEAD

RemoteSha b3cf7f8b71ef826a764a3b018d58f59410219029

Contents

extract_aagis_regions 2
extract_ae_zone 4
extract_area 5
extract_asc 6
extract_daas_soil_order 8
extract_data_drill 10
extract_patched_point 13
extract_smips 17
extract_topsoil_thickness 19

Index **22**

`extract_aagis_regions` *Extract 'Australian Agricultural and Grazing Industries Survey' (AAGIS) regions and zones using Australian GPS coordinates*

Description

Extracts the Australian Agricultural and Grazing Industries Survey (AAGIS) zones and regions. These regions represent the finest level of geographical aggregation for which the ABARES farm survey is designed to produce reliable estimates.

Usage

```
extract_aagis_regions(xy, locations = NULL, lonlat = NULL)
```

Arguments

xy	A list or data.frame object with names containing pairings of longitude and latitude values expressed as decimal degree values in that order. A data.frame may also be an **sf** points object. See **Details** for more.
locations	The column in which the location names are located as an Integer value. Defaults to 1.
lonlat	A vector of integer values indicating the column index in which the longitude and latitude values are contained. Defaults to 2:3.

Value

A data.table::data.table() with the provided GPS coordinates and the respective region(s)' name(s) and zone(s).

xy

xy as a List Object:

If xy is a list object, each individual vector's items should be named "x" (longitude) and "y" (latitude), respectively. The list item names should be descriptive of the individual vectors and will be included in a "location" column of the output.

xy as a Data Frame Object:

If xy is a data.frame, it should be a three column object with the first column being the location name, the second longitude and the third, latitude both as decimal degrees in numeric format.

xy as an sf Object:

If xy is an **sf** object, it should only be point geometry. If the data are projected, they will be reprojected to "EPSG:4326", Geographic Coordinate System WGS 84 before the extraction.

Note

The coordinate reference system for the AAGIS regions is internally converted to EPSG:4326 (WGS 84) for spatial intersection and ease of combining the returned values with other values from **fifo**.

Source

https://www.agriculture.gov.au/sites/default/files/documents/aagis_asgs16v1_g5a.shp_.zip

References

https://www.agriculture.gov.au/abares/research-topics/surveys/farm-definitions-methods#regions

Examples

```
locs <- list(
  "Merredin" = c(x = 118.28, y = -31.48),
  "Corrigin" = c(x = 117.87, y = -32.33),
  "Tamworth" = c(x = 150.84, y = -31.07)
)

extract_aagis_regions(xy = locs)
```

`extract_ae_zone` *Extract GRDC agroecological zones using Australian GPS coordinates*

Description

Extract GRDC agroecological zones using Australian GPS coordinates

Usage

```
extract_ae_zone(xy, locations = NULL, lonlat = NULL)
```

Arguments

`xy`	A `list` or `data.frame` object with names containing pairings of longitude and latitude values expressed as decimal degree values in that order. A `data.frame` may also be an **sf** points object. See **Details** for more.
`locations`	The column in which the location names are located as an `Integer` value. Defaults to `1`.
`lonlat`	A `vector` of `integer` values indicating the column index in which the longitude and latitude values are contained. Defaults to `2:3`.

Value

A `data.table::data.table()` with the provided GPS coordinates and the respective GRDC agroecological zone.

xy

xy as a List Object:

If `xy` is a `list` object, each individual vector's items should be named "x" (longitude) and "y" (latitude), respectively. The `list` item names should be descriptive of the individual vectors and will be included in a "location" column of the output.

xy as a Data Frame Object:

If `xy` is a `data.frame`, it should be a three column object with the first column being the location name, the second longitude and the third, latitude both as decimal degrees in `numeric` format.

xy as an sf Object:

If `xy` is an **sf** object, it should only be point geometry. If the data are projected, they will be reprojected to "EPSG:4326", Geographic Coordinate System WGS 84 before the extraction.

Examples

```
locs <- list(
  "Merredin" = c(x = 118.28, y = -31.48),
  "Corrigin" = c(x = 117.87, y = -32.33),
  "Tamworth" = c(x = 150.84, y = -31.07)
)

extract_ae_zone(xy = locs)
```

`extract_area` *Extract an area of interest using Australian GPS coordinates*

Description

A generic function to extract an area of interest for user-provided GPS coordinates.

Usage

```
extract_area(xy, spatial, area, locations = NULL, lonlat = NULL)
```

Arguments

`xy`	A `list` or `data.frame` object with names containing pairings of longitude and latitude values expressed as decimal degree values in that order. A `data.frame` may also be an **sf** points object. See **Details** for more.
`spatial`	a user-supplied `sf::sf()` or `terra::SpatVector()` object that contains information from which to derive location information, `area`.
`area`	the field in `spatial` that should be returned.
`locations`	The column in which the location names are located as an `Integer` value. Defaults to `1`.
`lonlat`	A `vector` of `integer` values indicating the column index in which the longitude and latitude values are contained. Defaults to `2:3`.

Value

A `data.table::data.table()` with the provided GPS coordinates and the respective `area` value from `spatial`.

xy

xy as a List Object:

If `xy` is a `list` object, each individual vector's items should be named "x" (longitude) and "y" (latitude), respectively. The `list` item names should be descriptive of the individual vectors and will be included in a "location" column of the output.

xy as a Data Frame Object:

If `xy` is a `data.frame`, it should be a three column object with the first column being the location name, the second longitude and the third, latitude both as decimal degrees in `numeric` format.

xy as an sf Object:

If `xy` is an **sf** object, it should only be point geometry. If the data are projected, they will be reprojected to "EPSG:4326", Geographic Coordinate System WGS 84 before the extraction.

Examples

```
# load the `aez` data included in the package for use in example only.
# the `extract_ae_zone()` performs this exact task, this is strictly for
# demonstration purposes only

library(sf)

aez <- st_read(system.file(
  "extdata",
  "aez.gpkg",
  package = "fifo",
  mustWork = TRUE
))

locs <- list(
  "Merredin" = c(x = 118.28, y = -31.48),
  "Corrigin" = c(x = 117.87, y = -32.33),
  "Tamworth" = c(x = 150.84, y = -31.07)
)

extract_area(xy = locs, spatial = aez, area = "AEZ")
```

---

extract_asc *Extract Australian Soil Class data from TERN's API using Australian GPS coordinates*

Description

Extracts the Australian Soil Class data at the GPS points provided, assuming that they are land-based coordinates in Australia from the Australian Soil Class (ASC). Values for the Confusion Index, the amount of model certainty around the soil class at that grid cell, are also supported.

Usage

```
extract_asc(
  xy,
  locations = NULL,
  lonlat = NULL,
  confusion_index = FALSE,
  api_key = nert::get_key(),
  max_tries = 3L,
  initial_delay = 1L
)
```

Arguments

xy	A `list` or `data.frame` object with names containing pairings of longitude and latitude values expressed as decimal degree values in that order. A `data.frame` may also be an **sf** points object. See **Details** for more.
locations	The column in which the location names are located as an `Integer` value. Defaults to `1`.
lonlat	A `vector` of `integer` values indicating the column index in which the longitude and latitude values are contained. Defaults to `2:3`.
confusion_index	`Boolean`, extract the values for the Confusion Index with the soil class? Defaults to `FALSE` and returns the soil class data only.
api_key	A `character` string containing your API key, a random string provided to you by TERN, for the request. Defaults to automatically detecting your key from your local .Renviron, .Rprofile or similar. Alternatively, you may directly provide your key as a string here or use functionality like that from **keyring**. If nothing is provided, you will be prompted on how to set up your R session so that it is auto-detected and a browser window will open at the TERN website for you to request a key.
max_tries	An integer `Integer` with the number of times to retry a failed download before emitting an error message. Defaults to `3`.
initial_delay	An `Integer` with the number of seconds to delay before retrying the download. This increases as the tries increment. Defaults to `1`.

Value

A `data.table::data.table()` with the provided GPS coordinates and the respective Australian Soil Class for the coordinates from the ASC data ("asc_longitude" and "asc_latitude").

xy

xy as a List Object:

If `xy` is a `list` object, each individual vector's items should be named "x" (longitude) and "y" (latitude), respectively. The `list` item names should be descriptive of the individual vectors and will be included in a "location" column of the output.

xy as a Data Frame Object:

If `xy` is a `data.frame`, it should be a three column object with the first column being the location name, the second longitude and the third, latitude both as decimal degrees in `numeric` format.

xy as an sf Object:

If `xy` is an **sf** object, it should only be point geometry. If the data are projected, they will be reprojected to "EPSG:4326", Geographic Coordinate System WGS 84 before the extraction.

Source

You must be logged in to view these.

ASC Data https://data.tern.org.au/model-derived/slga/NationalMaps/SoilClassifications/ASC/90m/ASC_EV_C_P_AU_TRN_N.cog.tif

Confusion Index https://data.tern.org.au/model-derived/slga/NationalMaps/SoilClassifications/ASC/90m/ASC_CI_C_P_AU_TRN_N.cog.tif

References

https://portal.tern.org.au/metadata/TERN/15728dba-b49c-4da5-9073-13d8abe67d7c

Examples

```
locs <- list(
  "Merredin" = c(x = 118.28, y = -31.48),
  "Corrigin" = c(x = 117.87, y = -32.33),
  "Tamworth" = c(x = 150.84, y = -31.07)
)

extract_asc(xy = locs, confusion_index = TRUE)
```

---

extract_daas_soil_order

Extract soil order from the 'Digital Atlas of Australian Soils' (DAAS) using Australian GPS coordinates

---

Description

[Superseded]

This function has been superseded by `extract_asc()`, which provides access to a more accurate dataset based on the Australian Soil Classification (ASC).

Extracts the major soil order at the GPS points provided assuming that they are land-based coordinates.

Usage

```
extract_daas_soil_order(xy, locations = NULL, lonlat = NULL)
```

Arguments

xy	A `list` or `data.frame` object with names containing pairings of longitude and latitude values expressed as decimal degree values in that order. A `data.frame` may also be an **sf** points object. See **Details** for more.
locations	The column in which the location names are located as an `Integer` value. Defaults to `1`.
lonlat	A `vector` of `integer` values indicating the column index in which the longitude and latitude values are contained. Defaults to `2:3`.

Value

A `data.table::data.table()` with the provided GPS coordinates and the respective Digital Atlas of Australian Soils (DAAS soil order), "Spatial Data Conversion of the Atlas of Australian Soils to the Australian Soil Classification v01".

xy

xy as a List Object:

If `xy` is a `list` object, each individual vector's items should be named "x" (longitude) and "y" (latitude), respectively. The `list` item names should be descriptive of the individual vectors and will be included in a "location" column of the output.

xy as a Data Frame Object:

If `xy` is a `data.frame`, it should be a three column object with the first column being the location name, the second longitude and the third, latitude both as decimal degrees in `numeric` format.

xy as an sf Object:

If `xy` is an **sf** object, it should only be point geometry. If the data are projected, they will be reprojected to "EPSG:4326", Geographic Coordinate System WGS 84 before the extraction.

Source

https://data.gov.au/data/dataset/5ccb44bf-93f2-4f94-8ae2-4c3f699ea4e7/resource/56ba5f25-2324-43b5-8df8-b9c69ae2ea0b/download/6f804e8b-2de9-4c88-adfa-918ec327c32f.zip

References

https://data.gov.au/data/dataset/2d0809ec-34c8-4e66-8cef-e3de2416c144

Examples

```
locs <- list(
  "Merredin" = c(x = 118.28, y = -31.48),
  "Corrigin" = c(x = 117.87, y = -32.33),
  "Tamworth" = c(x = 150.84, y = -31.07)
)

extract_daas_soil_order(xy = locs)
```

`extract_data_drill` *Extract Weather From SILO Data Drill Using Australian GPS Coordinates*

Description

A modified wrapper version of weatherOz::get_data_drill that allows for fetching many geophysical points or a single geophysical point. Extracts interpolated weather data from the SILO API from the gridded data, PatchedPointData, data set.

Usage

```
extract_data_drill(
  xy,
  locations = NULL,
  lonlat = NULL,
  start_date,
  end_date,
  values = "all",
  api_key = weatherOz::get_key(service = "SILO")
)
```

Arguments

`xy`	A `list` or `data.frame` object with names containing pairings of longitude and latitude values expressed as decimal degree values in that order. A `data.frame` may also be an **sf** points object. See **Details** for more.
`locations`	The column in which the location names are located as an `Integer` value. Defaults to `1`.
`lonlat`	A `vector` of `integer` values indicating the column index in which the longitude and latitude values are contained. Defaults to `2:3`.
`start_date`	A `character` string or `Date` object representing the beginning of the range to query in the format "yyyy-mm-dd" (ISO8601). Data returned is inclusive of this date.
`end_date`	A `character` string or `Date` object representing the end of the range query in the format "yyyy-mm-dd" (ISO8601). Data returned is inclusive of this date. Defaults to the current system date.
`values`	A `character` string with the type of weather data to return. See **Available Values** for a full list of valid values. Defaults to `all` with all available values being returned.
`api_key`	A `character` string specifying a valid email address to use for the request. The query will return an error if a valid email address is not provided.

Value

A `data.table::data.table()` with the weather data queried with the weather variables in alphabetical order. The first eight columns will always be:

- `longitude`,
- `latitude`,
- `elev_m` (elevation in metres),
- `date` (ISO8601 format, YYYYMMDD),
- `year`,
- `month`,
- `day`,
- `extracted` (the date on which the query was made)

Available Values

all Which will return all of the following values
rain (mm) Rainfall
max_temp (degrees C) Maximum temperature
min_temp (degrees C) Minimum temperature
vp (hPa) Vapour pressure
vp_deficit (hPa) Vapour pressure deficit
evap_pan (mm) Class A pan evaporation
evap_syn (mm) Synthetic estimate[1]
evap_comb (mm) Combination (synthetic estimate pre-1970, class A pan 1970 onwards)
evap_morton_lake (mm) Morton's shallow lake evaporation
radiation (Mj/m^2) Solar exposure, consisting of both direct and diffuse components
rh_tmax (%) Relative humidity at the time of maximum temperature
rh_tmin (%) Relative humidity at the time of minimum temperature
et_short_crop (mm) FAO56[4] short crop
et_tall_crop (mm) ASCE[5] tall crop[6]
et_morton_actual (mm) Morton's areal actual evapotranspiration
et_morton_potential (mm) Morton's point potential evapotranspiration
et_morton_wet (mm) Morton's wet-environment areal potential evapotranspiration over land
mslp (hPa) Mean sea level pressure

Value information

Solar radiation: total incoming downward shortwave radiation on a horizontal surface, derived from estimates of cloud oktas and sunshine duration[3].

Relative humidity: calculated using the vapour pressure measured at 9am, and the saturation vapour pressure computed using either the maximum or minimum temperature[6].

Evaporation and evapotranspiration: an overview of the variables provided by SILO is available here, https://data.longpaddock.qld.gov.au/static/publications/Evapotranspiration_overview.pdf.

Data codes

Data codes Where possible (depending on the file format), the data are supplied with codes indicating how each datum was obtained.

0 Official observation as supplied by the Bureau of Meteorology

15 Deaccumulated rainfall (original observation was recorded over a period exceeding the standard 24 hour observation period)

25 Interpolated from daily observations for that date

26 Synthetic Class A pan evaporation, calculated from temperatures, radiation and vapour pressure

35 Interpolated from daily observations using an anomaly interpolation method

75 Interpolated from the long term averages of daily observations for that day of year

xy

xy as a List Object:

If `xy` is a `list` object, each individual vector's items should be named "x" (longitude) and "y" (latitude), respectively. The `list` item names should be descriptive of the individual vectors and will be included in a "location" column of the output.

xy as a Data Frame Object:

If `xy` is a `data.frame`, it should be a three column object with the first column being the location name, the second longitude and the third, latitude both as decimal degrees in `numeric` format.

xy as an sf Object:

If `xy` is an **sf** object, it should only be point geometry. If the data are projected, they will be reprojected to "EPSG:4326", Geographic Coordinate System WGS 84 before the extraction.

Author(s)

Adam H. Sparks, <adamhsparks@gmail.com>

References

1. Rayner, D. (2005). Australian synthetic daily Class A pan evaporation. Technical Report December 2005, Queensland Department of Natural Resources and Mines, Indooroopilly, Qld., Australia, 40 pp.
2. Morton, F. I. (1983). Operational estimates of areal evapotranspiration and their significance to the science and practice of hydrology, *Journal of Hydrology*, Volume 66, 1-76.
3. Zajaczkowski, J., Wong, K., & Carter, J. (2013). Improved historical solar radiation gridded data for Australia, *Environmental Modelling & Software*, Volume 49, 64–77. DOI: doi:10.1016/j.envsoft.2013.06.013.
4. Food and Agriculture Organization of the United Nations, Irrigation and drainage paper 56: Crop evapotranspiration - Guidelines for computing crop water requirements, 1998.
5. ASCE's Standardized Reference Evapotranspiration Equation, proceedings of the National Irrigation Symposium, Phoenix, Arizona, 2000.

6. For further details refer to Jeffrey, S.J., Carter, J.O., Moodie, K.B. and Beswick, A.R. (2001). Using spatial interpolation to construct a comprehensive archive of Australian climate data, *Environmental Modelling and Software*, Volume 16/4, 309-330. DOI: doi:10.1016/S1364-8152(01)000081.

See Also

Other weather data: `extract_patched_point()`

Other SILO: `extract_patched_point()`

Examples

```
## Not run:
# Source data from a list of latitude and longitude coordinates in NSW
# and WA
locs <- list(
  "Merredin" = c(x = 118.28, y = -31.48),
  "Corrigin" = c(x = 117.87, y = -32.33),
  "Tamworth" = c(x = 150.84, y = -31.07)
)

wd <- extract_data_drill(
  xy = locs,
  start_date = "20211001",
  end_date = "20211201",
  values = "all",
  api_key = "your_api_key"
)

## End(Not run)
```

---

extract_patched_point *Extract weather from the nearest Patched Point data set BoM weather station in the SILO network using Australian GPS coordinates*

Description

Extracts station-based weather data for the provided dates from the nearest weather station to the provided longitude and latitude values in `x`. Weather data for the single closest station(s) to the provided longitude and latitude will always be returned.

Usage

```
extract_patched_point(
  xy,
  locations = NULL,
  lonlat = NULL,
  start_date,
```

```
  end_date,
  n_stations = 1L,
  values = "all",
  api_key = weatherOz::get_key(service = "SILO")
)
```

Arguments

xy	A `list` or `data.frame` object with names containing pairings of longitude and latitude values expressed as decimal degree values in that order. A `data.frame` may also be an **sf** points object. See **Details** for more.
locations	The column in which the location names are located as an `Integer` value. Defaults to `1`.
lonlat	A `vector` of `integer` values indicating the column index in which the longitude and latitude values are contained. Defaults to `2:3`.
start_date	A `character` string or `Date` object representing the beginning of the range to query in the format "yyyy-mm-dd" (ISO8601). Data returned is inclusive of this date.
end_date	A `character` string or `Date` object representing the end of the range query in the format "yyyy-mm-dd" (ISO8601). Data returned is inclusive of this date. Defaults to the current system date.
n_stations	An `integer` value that indicates how many stations per location should be returned. Defaults to one (1) station with only the nearest station being returned. Values greater than 1 will result in the next nearest stations being returned from 1 to n.
values	A `character` string with the type of weather data to return. See **Available Values** for a full list of valid values. Defaults to `all` with all available values being returned.
api_key	A `character` string specifying a valid email address to use for the request. The query will return an error if a valid email address is not provided.

Value

A `data.table::data.table()` with the weather data queried with the weather variables in alphabetical order. The first eight columns will always be:

- `location`,
- `x` (longitude of `location`),
- `y` (latitude of `location`),
- `station_code`,
- `station_name`,
- `year`,
- `day`,
- `date` (ISO8601 format, "YYYYMMDD").

Column Name Details

Column names are converted from the default returns of the API to be snake_case formatted and where appropriate, the names of the values that are analogous between SILO and DPIRD data are named using the same name for ease of interoperability, *e.g.*, using `rbind()` to create a `data.table` that contains data from both APIs.

The SILO documentation provides the following information for the PatchedPoint data.

These data are a continuous daily time series of data at either recording stations or grid points across Australia:

- *Data at station locations consists of observational records which have been supplemented by interpolated estimates when observed data are missing. Datasets are available at approximately 8,000 Bureau of Meteorology recording stations around Australia.*
- *Data at grid points consists entirely of interpolated estimates. The data are taken from the SILO gridded datasets and are available at any pixel on a 0.05° × 0.05° grid over the land area of Australia (including some islands).*

Available Values

all Which will return all of the following values

rain (mm) Rainfall

max_temp (degrees C) Maximum temperature

min_temp (degrees C) Minimum temperature

vp (hPa) Vapour pressure

vp_deficit (hPa) Vapour pressure deficit

evap_pan (mm) Class A pan evaporation

evap_syn (mm) Synthetic estimate[1]

evap_comb (mm) Combination (synthetic estimate pre-1970, class A pan 1970 onwards)

evap_morton_lake (mm) Morton's shallow lake evaporation

radiation (Mj/m^2) Solar exposure, consisting of both direct and diffuse components

rh_tmax (%) Relative humidity at the time of maximum temperature

rh_tmin (%) Relative humidity at the time of minimum temperature

et_short_crop (mm) FAO56[4] short crop

et_tall_crop (mm) ASCE[5] tall crop[6]

et_morton_actual (mm) Morton's areal actual evapotranspiration

et_morton_potential (mm) Morton's point potential evapotranspiration

et_morton_wet (mm) Morton's wet-environment areal potential evapotranspiration over land

mslp (hPa) Mean sea level pressure

Duplicated values

The resulting data.table::data.table may contain duplicated weather values depending on the distance from each location to the nearest station, *i.e.* one station may be the closest station to more than one queried location. Therefore, the final data.frame will have the same weather data for multiple locations as provided in `xy`.

Value information

Solar radiation: total incoming downward shortwave radiation on a horizontal surface, derived from estimates of cloud oktas and sunshine duration[3].

Relative humidity: calculated using the vapour pressure measured at 9am, and the saturation vapour pressure computed using either the maximum or minimum temperature[6].

Evaporation and evapotranspiration: an overview of the variables provided by SILO is available here, https://data.longpaddock.qld.gov.au/static/publications/Evapotranspiration_overview.pdf.

Data codes

The data are supplied with codes indicating how each datum was obtained.

0 Official observation as supplied by the Bureau of Meteorology

15 Deaccumulated rainfall (original observation was recorded over a period exceeding the standard 24 hour observation period).

25 Interpolated from daily observations for that date.

26 Synthetic Class A pan evaporation, calculated from temperatures, radiation and vapour pressure.

35 Interpolated from daily observations using an anomaly interpolation method.

75 Interpolated from the long term averages of daily observations for that day of year.

xy

xy as a List Object:

If `xy` is a `list` object, each individual vector's items should be named "x" (longitude) and "y" (latitude), respectively. The `list` item names should be descriptive of the individual vectors and will be included in a "location" column of the output.

xy as a Data Frame Object:

If `xy` is a `data.frame`, it should be a three column object with the first column being the location name, the second longitude and the third, latitude both as decimal degrees in `numeric` format.

xy as an sf Object:

If `xy` is an **sf** object, it should only be point geometry. If the data are projected, they will be reprojected to "EPSG:4326", Geographic Coordinate System WGS 84 before the extraction.

Author(s)

Adam H. Sparks, <adamhsparks@gmail.com>

References

1. Rayner, D. (2005). Australian synthetic daily Class A pan evaporation. Technical Report December 2005, Queensland Department of Natural Resources and Mines, Indooroopilly, Qld., Australia, 40 pp.
2. Morton, F. I. (1983). Operational estimates of areal evapotranspiration and their significance to the science and practice of hydrology, *Journal of Hydrology*, Volume 66, 1-76.

3. Zajaczkowski, J., Wong, K., & Carter, J. (2013). Improved historical solar radiation gridded data for Australia, *Environmental Modelling & Software*, Volume 49, 64–77. DOI: doi:10.1016/j.envsoft.2013.06.013.
4. Food and Agriculture Organization of the United Nations, Irrigation and drainage paper 56: Crop evapotranspiration - Guidelines for computing crop water requirements, 1998.
5. ASCE's Standardized Reference Evapotranspiration Equation, proceedings of the National Irrigation Symposium, Phoenix, Arizona, 2000.
6. For further details refer to Jeffrey, S.J., Carter, J.O., Moodie, K.B. and Beswick, A.R. (2001). Using spatial interpolation to construct a comprehensive archive of Australian climate data, *Environmental Modelling and Software*, Volume 16/4, 309-330. DOI: doi:10.1016/S1364-8152(01)000081.

See Also

Other weather data: `extract_data_drill()`

Other SILO: `extract_data_drill()`

Examples

```
## Not run:

# Source data from a list of latitude and longitude coordinates in NSW & WA

locs <- list(
  "Merredin" = c(x = 118.28, y = -31.48),
  "Corrigin" = c(x = 117.87, y = -32.33),
  "Tamworth" = c(x = 150.84, y = -31.07)
)

# Replace "your_api_key" with a valid email address.

w <- extract_patched_point(
  xy = locs,
  start_date = "20210101",
  end_date = "20210131",
  api_key = "your_api_key"
)

## End(Not run)
```

`extract_smips` *Extract soil moisture data from TERN's SMIPS using Australian GPS coordinates*

Description

Extracts the soil moisture at the GPS points provided assuming that they are land-based coordinates in Australia from the Soil Moisture Integration and Prediction System SMIPS data set.

Usage

```
extract_smips(
  xy,
  day,
  locations = NULL,
  lonlat = NULL,
  collection = "totalbucket",
  api_key = nert::get_key(),
  max_tries = 3L,
  initial_delay = 1L
)
```

Arguments

xy	A `list` or `data.frame` object with names containing pairings of longitude and latitude values expressed as decimal degree values in that order. A `data.frame` may also be an **sf** points object. See **Details** for more.
day	A vector of date(s) to query, *e.g.*, `day = "2017-12-31"` or `day = seq.Date(as.Date("2017-12-01"), as.Date("2017-12-31"), "days")`, both `Character` and `Date` classes are accepted.
locations	The column in which the location names are located as an `Integer` value. Defaults to `1`.
lonlat	A `vector` of `integer` values indicating the column index in which the longitude and latitude values are contained. Defaults to `2:3`.
collection	A character vector of the data collection to be queried, currently only "smips" is supported with the following collections: • SMindex • bucket1 • bucket2 • deepD • runoff • totalbucket. Defaults to "totalbucket". Multiple collections are supported, *e.g.*, `collection = c("SMindex", "totalbucket")`.
api_key	A `character` string containing your API key, a random string provided to you by TERN, for the request. Defaults to automatically detecting your key from your local .Renviron, .Rprofile or similar. Alternatively, you may directly provide your key as a string here or use functionality like that from **keyring**. If nothing is provided, you will be prompted on how to set up your R session so that it is auto-detected and a browser window will open at the TERN website for you to request a key.
max_tries	An integer `Integer` with the number of times to retry a failed download before emitting an error message. Defaults to `3`.
initial_delay	An `Integer` with the number of seconds to delay before retrying the download. This increases as the tries increment. Defaults to `1`.

Value

A `data.table::data.table()` with the provided GPS coordinates and the respective soil moisture data from SMIPS and coordinates from the SMIPS ("smips_longitude" and "smips_latitude").

xy

xy as a List Object:
If `xy` is a `list` object, each individual vector's items should be named "x" (longitude) and "y" (latitude), respectively. The `list` item names should be descriptive of the individual vectors and will be included in a "location" column of the output.

xy as a Data Frame Object:
If `xy` is a `data.frame`, it should be a three column object with the first column being the location name, the second longitude and the third, latitude both as decimal degrees in `numeric` format.

xy as an sf Object:
If `xy` is an **sf** object, it should only be point geometry. If the data are projected, they will be reprojected to "EPSG:4326", Geographic Coordinate System WGS 84 before the extraction.

Source

You must be logged in to view this. https://data.tern.org.au/model-derived/smips/v1_0/

References

https://portal.tern.org.au/metadata/TERN/d1995ee8-53f0-4a7d-91c2-ad5e4a23e5e0

Examples

```
locs <- list(
  "Merredin" = c(x = 118.28, y = -31.48),
  "Corrigin" = c(x = 117.87, y = -32.33),
  "Tamworth" = c(x = 150.84, y = -31.07)
)

extract_smips(xy = locs, day = "2023-09-23")
```

---

extract_topsoil_thickness

Extract topsoil thickness from ABARES's 'Soil Thickness for Australian Areas of Intensive Agriculture of Layer 1' data

---

Description

Extracts the soil thickness for Australian areas of intensive Agriculture of Layer 1 (A Horizon - top-soil) (derived from soil mapping) at the GPS points provided, assuming that they are land-based coordinates in Australia from the Soil Thickness for Australian Areas of Intensive Agriculture of Layer 1 data set on a 0.01 X 0.01 arc-degree quadrat.

Usage

```
extract_topsoil_thickness(xy, locations = NULL, lonlat = NULL)
```

Arguments

xy	A `list` or `data.frame` object with names containing pairings of longitude and latitude values expressed as decimal degree values in that order. A `data.frame` may also be an **sf** points object. See **Details** for more.
locations	The column in which the location names are located as an `Integer` value. Defaults to `1`.
lonlat	A `vector` of `integer` values indicating the column index in which the longitude and latitude values are contained. Defaults to `2:3`.

Value

A data.table::data.table with the provided GPS coordinates and the respective soil thickness data (mm) as an `Integer` value of the soil layer 1 from Soil Thickness for Australian Areas of Intensive Agriculture of Layer 1 and coordinates from the Soil Thickness ("thpk_1_longitude" and "thpk_1_latitude").

xy

xy as a List Object:

If `xy` is a `list` object, each individual vector's items should be named "x" (longitude) and "y" (latitude), respectively. The `list` item names should be descriptive of the individual vectors and will be included in a "location" column of the output.

xy as a Data Frame Object:

If `xy` is a `data.frame`, it should be a three column object with the first column being the location name, the second longitude and the third, latitude both as decimal degrees in `numeric` format.

xy as an sf Object:

If `xy` is an **sf** object, it should only be point geometry. If the data are projected, they will be reprojected to "EPSG:4326", Geographic Coordinate System WGS 84 before the extraction.

Source

https://anrdl-integration-web-catalog-saxfirxkxt.s3-ap-southeast-2.amazonaws.com/warehouse/staiar9cl__059/staiar9cl__05911a01eg_geo___.zip

References

https://data.agriculture.gov.au/geonetwork/srv/eng/catalog.search#/metadata/faa9f157-8e17-4b23-b6a7

Examples

```
locs <- list(
  "Merredin" = c(x = 118.28, y = -31.48),
  "Corrigin" = c(x = 117.87, y = -32.33),
  "Tamworth" = c(x = 150.84, y = -31.07)
)

extract_topsoil_thickness(xy = locs)
```

Index

∗ **SILO**
    extract_data_drill, 10
    extract_patched_point, 13
∗ **weather data**
    extract_data_drill, 10
    extract_patched_point, 13

data.table::data.table, *15*, *20*
data.table::data.table(), *3–5*, *7*, *9*, *11*, *14*, *19*

extract_aagis_regions, 2
extract_ae_zone, 4
extract_area, 5
extract_asc, 6
extract_daas_soil_order, 8
extract_data_drill, 10
extract_data_drill(), *17*
extract_patched_point, 13
extract_patched_point(), *13*
extract_smips, 17
extract_topsoil_thickness, 19

rbind(), *15*

sf::sf(), *5*

terra::SpatVector(), *5*

weatherOz::get_data_drill, *10*