Package: antanym (via r-universe)

October 28, 2024

Type Package

Title Antarctic Geographic Place Names

Version 0.4.4

Description Antarctic geographic names from the Composite Gazetteer of Antarctica, and functions for working with those place names.

URL https://docs.ropensci.org/antanym, https://github.com/ropensci/antanym

BugReports https://github.com/ropensci/antanym/issues

Depends R (>= 3.3.0)

License MIT + file LICENSE

Encoding UTF-8

LazyData true

Imports assertthat, C50, geosphere, httr, magrittr, rappdirs, raster, readr, sp, stringi

Suggests covr, dplyr, testthat (>= 2.0.0), knitr, leaflet, rgdal, rgeos, rmarkdown, rworldmap

RoxygenNote 7.1.1

VignetteBuilder knitr

X-schema.org-applicationCategory Antarctic/Southern Ocean

X-schema.org-keywords Antarctic, Southern Ocean, place names, gazetteer

X-schema.org-isPartOf https://ropensci.org, https://scar.org

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

RemoteUrl https://github.com/ropensci/antanym

RemoteRef master

RemoteSha 8b6a40ae20d0454e38cdf9e093f598e23099609b

Contents

`antanym` **antanym**

Description

Antarctic geographic place names from the Composite Gazetteer of Antarctica, and functions for working with those place names.

References

http://data.aad.gov.au/aadc/gaz/scar

`an_cache_directory` *The cache directory used by antanym*

Description

The cache directory used by antanym

Usage

```
an_cache_directory(cache)
```

Arguments

cache	string: the gazetteer data can be cached locally, so that it can be used offline later. Valid values are "session", "persistent", or a directory name. Specifying cache="session" will use a temporary directory that persists only for the current session. cache="persistent" will use rappdirs::user_cache_dir() to determine the appropriate directory to use. Otherwise, the input string will be assumed to be the path to the directory to use

Value

directory path

See Also

an_read

Examples

```
## per-session caching
an_cache_directory(cache = "session")

## persistent caching that will keep the data from one R session to the next
an_cache_directory(cache = "persistent")
```

---

an_cga_metadata *Information about the Composite Gazetteer of Antarctica data structure*

Description

The Composite Gazetteer of Antarctica data structure (as returned by an_read):

Usage

```
an_cga_metadata(simplified = TRUE)
```

Arguments

simplified	logical: if TRUE, only describe the simplified set of columns (see the equivalent parameter in an_read)

Value

a data frame with columns "field" and "description"

References

https://data.aad.gov.au/aadc/gaz/scar/, https://www.scar.org/data-products/place-names/

See Also

an_read

Examples

```
an_cga_metadata()
```

---

an_feature_types *List feature types present in gazetteer data*

Description

The gazetteer place names are associated with different feature types (e.g. "Hill", "Mountain", "Water body"). This function lists the feature types that are present in a given data frame.

Usage

```
an_feature_types(gaz)
```

Arguments

gaz	data.frame or SpatialPointsDataFrame: as returned by an_read, an_preferred, or an_filter

Value

character vector of country names

See Also

an_filter for filtering data according to feature type

Examples

```
## Not run:
 g <- an_read(cache = "session")

 ## what feature types do we have in our data?
 an_feature_types(g)

## End(Not run)
```

an_filter *Filter a collection of place names by various criteria*

Description

A data frame of place names can be filtered according to name, geographic location, feature type, or other criteria. All text-related matches are by default treated as regular expressions and are case-insensitive: you can change this behaviour via the `ignore_case` and `as_regex` parameters.

Usage

```
an_filter(
  gaz,
  query,
  feature_ids,
  extent,
  feature_type,
  origin,
  origin_gazetteer,
  ignore_case = TRUE,
  as_regex = TRUE
)
```

Arguments

`gaz`	data.frame or SpatialPointsDataFrame: as returned by `an_read` or `an_preferred`
`query`	character: vector of place name terms to search for. Returned place names will be those that match all entries in `query`
`feature_ids`	numeric: return only place names associated with the features identified by these identifiers. Currently these values can only be `scar_common_id` values
`extent`	vector of c(longitude_min, longitude_max, latitude_min, latitude_max): if provided, search only for names within this bounding box. `extent` can also be passed as a raster Extent object, a Raster object (in which case its extent will be used), a Spatial object (in which case the bounding box of the object will be used as the extent), or a matrix (in which case it will be assumed to be the output of `sp::bbox`)
`feature_type`	string: return only place names corresponding to feature types matching this pattern. For valid feature type names see `an_feature_types`
`origin`	string: return only place names originating from bodies (countries or organisations) matching this pattern. For valid `origin` values see `link{an_origins}`
`origin_gazetteer`	string: return only place names originating from gazetteers matching this pattern. For valid gazetteer names see `an_gazetteers`
`ignore_case`	logical: if `TRUE`, use case-insensitive text matching
`as_regex`	logical: if `TRUE`, treat `query` and other string input parameters as regular expressions. If `FALSE`, they will be treated as fixed strings to match against

Value

data.frame of results

References

https://data.aad.gov.au/aadc/gaz/scar/, https://www.scar.org/data-products/place-names/

See Also

an_read, an_gazetteers, an_origins

Examples

```
## Not run:
 g <- an_read(cache = "session")

 ## simple search for any place name containing the word 'William'
 an_filter(g, query = "William")

 ## which bodies (countries or organisations) provided the names in our data?
 an_origins(g)

 ## find names containing "William" and originating from Australia or the USA
 an_filter(g, query = "William", origin = "Australia|United States of America")

 ## this search will return no matches
 ## because the actual place name is 'William Scoresby Archipelago'
 an_filter(g, query = "William Archipelago")

 ## we can split the search terms so that each is matched separately
 an_filter(g, query = c("William", "Archipelago"))

 ## or use a regular expression
 an_filter(g, query = "William .* Archipelago")

 ## or refine the search using feature type
 an_filter(g, query = "William", feature_type = "Archipelago")

 ## what feature types do we have in our data?
 an_feature_types(g)

 ## for more complex text searching, use regular expressions
 ## e.g. names matching "West" or "East"
 an_filter(g, query = "West|East")

 ## names starting with "West" or "East"
 an_filter(g, query = "^(West|East)")

 ## names with "West" or "East" appearing as complete words in the name
 ## ["\b" matches a word boundary: see help("regex") ]
 an_filter(g, query = "\\b(West|East)\\b")
```

```
## filtering by spatial extent
nms <- an_filter(g, extent = c(100, 120, -70, -65), origin = "Australia")
with(nms, plot(longitude, latitude))
with(nms, text(longitude, latitude, place_name))

## searching within the extent of an sp object
my_sp <- sp::SpatialPoints(cbind(c(100, 120), c(-70, -65)))
an_filter(g, extent = my_sp)

## or equivalently
an_filter(g, extent = bbox(my_sp))

## or using the sp form of the gazetteer data
gsp <- an_read(cache = "session", sp = TRUE)
an_filter(gsp, extent = my_sp)

## using the pipe operator
g %>% an_filter(query = "Ross", feature_type = "Ice shelf|Mountain")

g %>% an_near(loc = c(100, -66), max_distance = 20) %>%
      an_filter(feature_type = "Island")

## find all names for feature 1589 and the naming
##  authority for each name
an_filter(g, feature_ids = 1589)[, c("place_name", "origin")]

## End(Not run)
```

---

an_gazetteers *The place name gazetteers available*

Description

Return a character vector that lists all of the gazetteers present in the gaz data, or (if gaz was not provided) all of the gazetteers available through the antanym package. Currently only one gazetteer is available: the Composite Gazetteer of Antarctica.

Usage

```
an_gazetteers(gaz)
```

Arguments

gaz	data.frame or SpatialPointsDataFrame: (optional) as returned by an_read, an_preferred, or an_filter

Value

character vector. If gaz was provided, this will be a list of all gazetteers present in gaz. Otherwise, it will be a list of all gazetteers available through the antanym package

See Also

an_read, an_filter

Examples

```
an_gazetteers()

## Not run:
 g <- an_read(cache = "session")
 an_gazetteers(g)

## End(Not run)
```

---

an_get_url *Get links to gazetteer entries*

Description

Each entry in the Composite Gazetteer of Antarctica has its own web page. The an_url function will return the URL of the page associated with a given gazetteer entry.

Usage

```
an_get_url(gaz)
```

Arguments

gaz	data.frame or SpatialPointsDataFrame: as returned by an_read, an_preferred, or an_filter

Value

character vector, where each component is a URL to a web page giving more information about the associated gazetteer entry

References

https://data.aad.gov.au/aadc/gaz/scar/, https://www.scar.org/data-products/place-names/

Examples

```
## Not run:
 g <- an_read(cache = "session")
 my_url <- an_get_url(an_filter(g, query = "Ufs Island")[1, ])
 browseURL(my_url)

## End(Not run)
```

an_mapscale *Calculate approximate map scale*

Description

Calculate approximate map scale

Usage

```
an_mapscale(map_dimensions, map_extent)
```

Arguments

map_dimensions	numeric: 2-element numeric giving width and height of the map, in mm
map_extent	vector of c(longitude_min, longitude_max, latitude_min, latitude_max): the geographic extent of the map. map_extent can also be passed as a raster Extent object, a Raster object (in which case its extent will be used), a Spatial object (in which case the bounding box of the object will be used as the extent), or a matrix (in which case it will be assumed to be the output of sp::bbox)

Value

numeric

Examples

```
## an A3-sized map of the Southern Ocean (1:20M)
an_mapscale(map_dimensions = c(400, 570), map_extent = c(-180, 180, -90, -40))
```

an_near *Find placenames near a given location*

Description

Find placenames near a given location

Usage

```
an_near(gaz, loc, max_distance)
```

Arguments

gaz	data.frame or SpatialPointsDataFrame: as returned by an_read, an_preferred, or an_filter
loc	numeric: target location (a two-element numeric vector giving longitude and latitude, or a SpatialPoints object)
max_distance	numeric: maximum search distance in kilometres

Value

data.frame of results

References

https://data.aad.gov.au/aadc/gaz/scar/, https://www.scar.org/data-products/place-names/

See Also

an_read

Examples

```
## Not run:
 g <- an_read(cache = "session")

 ## named features within 10km of 110E, 66S
 an_near(g, loc = c(110, -66), max_distance = 10)

 ## using pipe operator
 g %>% an_near(loc = c(100, -66), max_distance = 10)

 ## with sp objects
 gsp <- an_read(cache = "session", sp = TRUE)
 loc <- sp::SpatialPoints(matrix(c(110, -66), nrow = 1),
   proj4string = CRS("+proj=longlat +datum=WGS84 +ellps=WGS84"))
 an_near(gsp, loc = loc, max_distance = 10)

## End(Not run)
```

---

an_origins *List the origins of place names present in gazetteer data*

Description

The Composite Gazetteer of Antarctica is a compilation of place names provided by different countries and organisations. This function lists the originating bodies that provided the names in a given data frame.

Usage

```
an_origins(gaz)
```

Arguments

gaz	data.frame or SpatialPointsDataFrame: as returned by an_read, an_preferred, or an_filter

Value

character vector of origin names (countries or organisations)

See Also

an_filter for filtering data according to origin

Examples

```
## Not run:
 g <- an_read(cache = "session")

 ## which bodies (countries or organisations) provided the names in our data?
 an_origins(g)

## End(Not run)
```

---

an_preferred *Find one name per feature in the Composite Gazetteer*

Description

The Composite Gazetteer of Antarctica is a compilation of place names provided by different countries and organisations. The composite nature of the CGA means that there may be multiple names associated with a single feature. The an_preferred function can be used to resolve a single name per feature. Provide one or more origin entries and the input gaz will be filtered to a single name per feature. For features that have multiple names (e.g. have been named by multiple countries) a single name will be chosen, preferring names from the specified origin bodies where possible.

Usage

```
an_preferred(gaz, origin, unmatched = "random")
```

Arguments

gaz	data.frame or SpatialPointsDataFrame: as returned by an_read or an_filter
origin	character: vector of preferred name origins (countries or organisations), in order of preference. If a given feature has been named by one of these bodies, this place name will be chosen. If the feature in question has not been given a name by any of these bodies, a place name given by another body will be chosen, with preference according to the unmatched parameter. For valid origin values, see an_origins
unmatched	string: how should names be chosen for features that have not been been named by one of the preferred origin bodies? Valid values are "random" (the non-preferred originating bodies will be randomly ordered) or "count" (the non-preferred originating bodies will be ordered by their number of entries, with the largest first)

Value

data.frame of results

References

https://data.aad.gov.au/aadc/gaz/scar/, https://www.scar.org/data-products/place-names/

See Also

an_read, an_origins

Examples

```
## Not run:
 g <- an_read(cache = "session")

 ## get a single name per feature, preferring the
 ##  Polish name where there is one
 pnames <- an_preferred(g, origin = "Poland")

 ## names starting with "Sm", preferring US names then
 ##  Australian ones if available
 g %>% an_filter("^Sm") %>%
       an_preferred(origin = c("United States of America", "Australia"))

## End(Not run)
```

---

an_read *Load Antarctic place name data*

Description

Place name data will be downloaded and optionally cached locally. If you wish to be able to use `antanym` offline, consider using `cache = "persistent"` so that the cached data will persist from one R session to the next. See an_cache_directory to get the path to the cache directory.

Usage

```
an_read(
  gazetteers = "all",
  sp = FALSE,
  cache,
  refresh_cache = FALSE,
  simplified = TRUE,
  verbose = FALSE
)
```

Arguments

gazetteers	character: vector of gazetteers to load. For the list of available gazetteers, see an_gazetteers. Use gazetteers = "all" to load all available gazetteers. Currently only one gazetteer is available: the Composite Gazetteer of Antarctica
sp	logical: if FALSE return a data.frame; if TRUE return a SpatialPointsDataFrame
cache	string: the gazetteer data can be cached locally, so that it can be used offline later. Valid values are "session", "persistent", or a directory name. Specifying cache = "session" will use a temporary directory that persists only for the current session. cache = "persistent" will use rappdirs::user_cache_dir() to determine the appropriate directory to use. Otherwise, if a string is provided it will be assumed to be the path to the directory to use. In this case, an attempt will be made to create the cache directory if it does not exist. A warning will be given if a cached copy of the data exists and is more than 30 days old
refresh_cache	logical: if TRUE, and a data file already exists in the cache, it will be refreshed. If FALSE, the cached copy will be used
simplified	logical: if TRUE, only return a simplified set of columns (see details in "Value", below)
verbose	logical: show progress messages?

Value

a data.frame or SpatialPointsDataFrame, with the following columns (note that not all information is populated for all place names):

- gaz_id - the unique identifier of each gazetteer entry. Note that the same feature (e.g. "Browns Glacier") might have multiple gazetteer entries, each with their own gaz_id, because the feature has been named multiple times by different naming authorities. The scar_common_id for these entries will be identical, because scar_common_id identifies the feature itself
- scar_common_id - the unique identifier (in the Composite Gazetteer of Antarctica) of the feature. A single feature may have multiple names, given by different naming authorities
- place_name - the name of the feature
- place_name_transliterated - the name of the feature transliterated to simple ASCII characters (e.g. with diacritical marks removed)
- longitude and latitude - the longitude and latitude of the feature (negative values indicate degrees west or south). Note that many features are not point features (e.g. mountains, lakes), in which case the longitude and latitude values are indicative only, generally of the centroid of the feature
- altitude - the altitude of the feature, in metres relative to sea level. Negative values indicate features below sea level
- feature_type_name - the feature type (e.g. "Archipelago", "Channel", "Mountain")
- date_named - the date on which the feature was named
- narrative - a text description of the feature; may include a synopsis of the history of its name
- named_for - the person after whom the feature was named, or other reason for its naming. For historical reasons the distinction between "narrative" and "named for" is not always obvious

- origin - the naming authority that provided the name. This is a country name, or organisation name for names that did not come from a national source
- relic - if TRUE, this name is associated with a feature that no longer exists (e.g. an ice shelf feature that has disappeared)
- gazetteer - the gazetteer from which this information came (currently only "CGA")

If simplified is FALSE, these additional columns will also be included:

- meeting_date - the date on which the name was formally approved by the associated naming authority. This is not available for many names: see the date_named column
- meeting_paper - references to papers or documents associated with the naming of the feature
- remote_sensor_info - text describing the remote sensing information (e.g. satellite platform name and image details) used to define the feature, if applicable
- coordinate_accuracy - an indicator of the accuracy of the coordinates, in metres
- altitude_accuracy - an indicator of the accuracy of the altitude value, in metres
- cga_source_gazetteer - for the Composite Gazetteer, this entry gives the source gazetteer from which this entry was taken. This is currently either a three-letter country code (e.g. "ESP", "USA") or "GEBCO" (for the GEBCO gazetteer of undersea features)
- country_name - the full name of the country where cga_source_gazetteer is a country
- source_name - the cartographic/GIS/remote sensing source from which the coordinates were derived
- source_publisher - where coordinates were derived from a map, the publisher of that map
- source_scale - the scale of the map from which the coordinates were derived
- source_institution - the institution from which the coordinate information came
- source_person - the contact person at the source institution, if applicable
- source_country_code - the country from which the coordinate information came
- source_identifier - where a coordinate or elevation was derived from a map, the identifier of that map
- comments - comments about the name or naming process

References

https://data.aad.gov.au/aadc/gaz/scar/, https://www.scar.org/data-products/place-names/

See Also

an_cache_directory, an_gazetteers, an_cga_metadata

Examples

```
## Not run:
 ## download without caching
 g <- an_read()

 ## download to session cache, in sp format
```

```
g <- an_read(cache = "session", sp = TRUE)

## download and cache to a persistent directory for later, offline use
g <- an_read(cache = "persistent")

## refresh the cached copy
g <- an_read(cache = "persistent", refresh_cache = TRUE)

## download and cache to a persistent directory of our choice
g <- an_read(cache = "c:/my/cache/directory")

## End(Not run)
```

---

an_suggest *Suggest names for a map (experimental)*

Description

Features are given a suitability score based on maps prepared by expert cartographers. Data were tabulated from a collection of such maps, indicating for each feature whether it was named on a given map, along with details (such as scale) of the map. These data are used as the basis of a recommendation algorithm, which suggests the best features to name on a map given its properties (extent and scale). This is an experimental function and currently only implemented for `map_scale` values of 10 million or larger.

Usage

```
an_suggest(gaz, map_scale, map_extent, map_dimensions)
```

Arguments

`gaz`	data.frame or SpatialPointsDataFrame: as returned by `an_read`, `an_preferred`, or `an_filter`
`map_scale`	numeric: the scale of the map (e.g. 20e6 for a 1:20M map). If `map_scale` is not provided, it will be estimated from `map_extent` and `map_dimensions`
`map_extent`	vector of c(longitude_min, longitude_max, latitude_min, latitude_max): the extent of the area for which name suggestions are sought. This is required if `map_scale` is not provided, and optional if `map_scale` is provided (if `map_extent` is provided in this situation then the `gaz` data frame will be filtered to this extent before the suggestion algorithm is applied; otherwise all names in `gaz` will be considered). `map_extent` can also be passed as a raster Extent object, a Raster object (in which case its extent will be used), a Spatial object (in which case the bounding box of the object will be used as the extent), or a matrix (in which case it will be assumed to be the output of `sp::bbox`)
`map_dimensions`	numeric: 2-element numeric giving width and height of the map, in mm. Not required if `map_scale` is provided

Value

data.frame of names with a "score" column added. Score values range from 0 to 1. The data frame will be sorted in descending score order. Names with higher scores are those that are suggested as the most suitable for display.

See Also

an_read an_thin

Examples

```
## Not run:
 g <- an_read(cache = "session")

 ## get a single name per feature, preferring the
 ##  Australian name where there is one
 g <- an_preferred(g, origin = "Australia")

 ## suggested names for a 100x100 mm map covering 60-90E, 70-60S
 ##  (this is about a 1:12M scale map)
 suggested <- an_suggest(g, map_extent = c(60, 90, -70, -60), map_dimensions = c(100, 100))
 head(suggested, 20) ## top 20 names

 ## an equivalent result can be achieved by supplying map scale and extent
 suggested <- an_suggest(g, map_scale = 12e6, map_extent = c(60, 90, -70, -60))

## End(Not run)
```

---

an_thin *Thin names to give approximately uniform spatial coverage*

Description

The provided data.frame of names will be thinned down to a smaller number of names. The thinning process attempts to select a subset of names that are uniformly spatially distributed, while simultaneously choosing the most important names (according to their relative score in the score_col column.

Usage

```
an_thin(gaz, n, score_col = "score", score_weighting = 5, row_limit = 2000)
```

Arguments

gaz	data.frame or SpatialPointsDataFrame: typically as returned by an_suggest
n	numeric: number of names to return

score_col — string: the name of the column that gives the relative score of each name (e.g. as returned by an_suggest). Names with higher scores will be preferred by the thinning process. If the specified score_col column is not present in gaz, or if all values within that column are equal, then the thinning will be based entirely on the spatial distribution of the features

score_weighting — numeric: weighting of scores relative to spatial distribution. A lower score_weighting value will tend to choose lower-scored names in order to achieve better spatial uniformity. A higher score_weighting value will trade spatial uniformity in favour of selecting higher-scored names

row_limit — integer: the maximum number of rows allowed in gaz; see Details. Data frames larger than this will not be processed (with an error).

Details

Note that the algorithm calculates all pairwise distances between the rows of gaz. This is memory-intensive, and so if gaz has many rows the algorithm will fail or on some platforms might crash. Input gaz data.frames with more than row_limit rows will not be processed for this reason. You can try increasing row_limit from its default value if necessary.

Value

data.frame

See Also

an_read, an_suggest

Examples

```
## Not run:
 g <- an_read(cache = "session")

 ## get a single name per feature, preferring the
 ##  Japanese name where there is one
 g <- an_preferred(g, origin = "Japan")

 ## suggested names for a 100x100 mm map covering 60-90E, 70-60S
 ##  (this is about a 1:12M scale map)
 suggested <- an_suggest(g, map_extent = c(60, 90, -70, -60), map_dimensions = c(100, 100))

 ## find the top 20 names by score
 head(suggested, 20)

 ## find the top 20 names chosen for spatial coverage and score
 an_thin(suggested, 20)

## End(Not run)
```

Index