Package: giscoR (via r-universe)

November 2, 2026

Type Package

Title Download 'Eurostat' 'GISCO' Spatial Data

Version 1.3.0.9000

Description Tools to download global and European spatial data from the 'Eurostat' 'GISCO' (Geographic Information System of the Commission) data distribution <https://ec.europa.eu/eurostat/web/gisco>. The package provides helpers for country boundaries, Nomenclature of Territorial Units for Statistics ('NUTS') regions, administrative units, statistical units, transport networks, basic service locations and other 'GISCO' datasets. This package is neither affiliated with nor endorsed by 'Eurostat'.

License GPL-3

URL https://ropengov.github.io/giscoR/, https://github.com/rOpenGov/giscoR

BugReports https://github.com/rOpenGov/giscoR/issues

Depends R (>= 4.1)

Imports cli, countrycode (>= 1.2.0), httr2 (>= 1.2.0), jsonlite, lifecycle, rappdirs (>= 0.3.0), sf (>= 1.0.0), tibble, tools, utils

Suggests dplyr, eurostat, ggplot2 (>= 3.5.0), knitr, quarto, stats, testthat (>= 3.0.0), withr

VignetteBuilder quarto

Config/roxygen2/markdown TRUE

Config/roxygen2/version 8.1.1

Config/testthat/edition 3

Config/testthat/parallel true

Copyright The general copyright notice and license policy from Eurostat applies (see file COPYRIGHTS). Some datasets have additional download and usage provisions. See <https://ec.europa.eu/eurostat/web/gisco/geodata>.

Encoding UTF-8

LazyData true

X-schema.org-applicationCategory cartography

X-schema.org-isPartOf https://ropengov.org/

X-schema.org-keywords ropengov, r, spatial, api-wrapper, rstats, r-package, eurostat, gisco, thematic-maps, eurostat-data, cran, ggplot2, gis, cran-r

Config/pak/sysreqs libabsl-dev cmake libgdal-dev gdal-bin libgeos-dev libssl-dev libproj-dev libsqlite3-dev libudunits2-dev

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

Date/Publication 2026-11-02 19:27:25 UTC

RemoteUrl https://github.com/rOpenGov/giscoR

RemoteRef HEAD

RemoteSha d72cbda2b2e4e16214980b88047a4914440ab88a

Contents

gisco_address_api . . . 3
gisco_attributions . . . 6
gisco_bulk_download . . . 8
gisco_clear_cache . . . 11
gisco_coastal_lines . . . 12
gisco_countries_2024 . . . 13
gisco_countrycode . . . 14
gisco_db . . . 15
gisco_get_airports . . . 16
gisco_get_cached_db . . . 17
gisco_get_census . . . 18
gisco_get_coastal_lines . . . 20
gisco_get_communes . . . 22
gisco_get_countries . . . 24
gisco_get_education . . . 26
gisco_get_grid . . . 29
gisco_get_healthcare . . . 31
gisco_get_lau . . . 34
gisco_get_metadata . . . 36
gisco_get_nuts . . . 37
gisco_get_ports . . . 40
gisco_get_postal_codes . . . 42
gisco_get_unit . . . 44
gisco_get_urban_audit . . . 47
gisco_id_api . . . 50
gisco_nuts_2024 . . . 53
gisco_set_cache_dir . . . 55

Index **58**

`gisco_address_api` *GISCO Address API*

Description

Functions to interact with the GISCO Address API, which supports geocoding and reverse geocoding with a pan-European address database.

Each endpoint supported by **giscoR** has a specific function. See **Details**. Search supports structured queries and freeform queries with `q`. Autocomplete through `/search?suggest=` is not implemented in **giscoR**.

Structured searches support approximate string matching. The API's freeform query parameter `q` does not support approximate string matching.

Usage

```
gisco_address_api_search(
  country = NULL,
  province = NULL,
  city = NULL,
  road = NULL,
  housenumber = NULL,
  postcode = NULL,
  verbose = FALSE,
  q = NULL
)

gisco_address_api_reverse(x, y, country = NULL, verbose = FALSE)

gisco_address_api_bbox(
  country = NULL,
  province = NULL,
  city = NULL,
  road = NULL,
  postcode = NULL,
  verbose = FALSE
)

gisco_address_api_countries(verbose = FALSE)

gisco_address_api_provinces(country = NULL, city = NULL, verbose = FALSE)

gisco_address_api_cities(country = NULL, province = NULL, verbose = FALSE)

gisco_address_api_roads(
  country = NULL,
```

```
  province = NULL,
  city = NULL,
  verbose = FALSE
)

gisco_address_api_housenumbers(
  country = NULL,
  province = NULL,
  city = NULL,
  road = NULL,
  postcode = NULL,
  verbose = FALSE
)

gisco_address_api_postcodes(
  country = NULL,
  province = NULL,
  city = NULL,
  verbose = FALSE
)

gisco_address_api_copyright(verbose = FALSE)

gisco_address_api_most_populated_cell(
  province = NULL,
  city = NULL,
  verbose = FALSE
)
```

Arguments

country	A country code (country = "LU").
province	A province within a country. This is a generic term whose administrative level varies by country. For a list of provinces within a country, use the provinces endpoint (gisco_address_api_provinces(country = "LU")).
city	A city within a province. This is a generic term whose administrative level varies by country. For a list of cities within a province, use the cities endpoint (gisco_address_api_cities(province = "capellen")).
road	A road within a city.
housenumber	The house number or house name within a road or street.
postcode	A postcode to use with the previous arguments.
verbose	A logical value indicating whether to display informational messages.
q	A single non-empty string for a freeform address search, or NULL for a structured search. The string can contain a street, house number, city or postcode. Freeform queries do not support approximate matching.
x, y	Longitude and latitude coordinates to convert into human-readable addresses. Reverse geocoding returns at most five results.

Details

For `gisco_address_api_most_populated_cell()`, supply a non-empty string for `province` or `city`. If both are supplied, the search is restricted to the city within the province.

The following table describes the endpoints supported by **giscoR**, based on the GISCO Address API endpoint documentation:

Endpoint	Description
`/countries`	All country codes compatible with the GISCO Address API. Check the coverage map for available
`/provinces`	All provinces within the specified country. The endpoint can also retrieve the province for a specif
`/cities`	All cities within a specified province or country.
`/roads`	Roads or streets within a specified city, limited to 1,000 results.
`/housenumbers`	House numbers or names within the specified road, limited to 1,000 results. In some countries, an
`/postcodes`	Postcodes within the specified address component, such as country, province or city, limited to 1,0
`/search`	Structured or freeform queries to the address database. Various argument combinations can retriev
`/reverse`	Structured addresses for longitude and latitude coordinates, limited to five results.
`/bbox`	A WKT bounding box for an address component, depending on the specified arguments.
`/most-populated-cell`	The most populated census grid cell for a province or city. If both are supplied, the search is restric
`/copyright`	The copyright text for each available country in the GISCO Address API.

Use `q` in `gisco_address_api_search()` for a freeform search. Unlike structured searches, freeform queries do not support approximate matching. Autocomplete through `/search?suggest=` is not implemented in **giscoR**.

The resulting object may include these variables:

Property name	Description
`LD`	Locator designator, which represents the house number part of the address.
`TF`	Thoroughfare, which represents the street or road part of the address.
`L0`	Level 0 of the API administrative levels. Values are two-character country codes.
`L1`	Level 1 of the API administrative levels. Values are province names. "Province" is a generic term that may v
`L2`	Level 2 of the API administrative levels. Values are town or city names. "City" is a generic term that may va
`I3`	ISO 3166-1 alpha-3 country code.
`PC`	Postal code.
`N0`	NUTS 0.
`N1`	NUTS 1.
`N2`	NUTS 2.
`N3`	NUTS 3.
`X` and `Y`	Numeric coordinates extracted from the API's `XY` pair. Search and reverse geocoding return longitude and la
`OL`	The Open Location Code for the address.

Value

A tibble in most cases, except `gisco_address_api_search()`, `gisco_address_api_reverse()` and `gisco_address_api_bbox()`, which return an `sf` object when geometry is available. Searches with no results return an empty tibble. `gisco_address_api_bbox()` returns `NULL` when no bounding box is found. Failed requests return `NULL`.

`gisco_address_api_most_populated_cell()` returns a one-row tibble for the most populated census grid cell, or `NULL` when no cell is found or the request fails. Its numeric `X` and `Y` columns preserve the API coordinates, whose CRS is not specified in the API documentation.

Source

https://gisco-services.ec.europa.eu/addressapi/docs/screen/home.

See Also

gisco_id_api for GISCO ID service API lookups.

See the GISCO Address API documentation at https://gisco-services.ec.europa.eu/addressapi/docs/screen/home.

GISCO API tools: gisco_id_api

Examples

```
# Cities in a region.

gisco_address_api_cities(country = "PT", province = "LISBOA")

# Geocode and reverse geocode with `sf` objects.
# Structured search.
struct <- gisco_address_api_search(
  country = "LU", city = "Luxembourg",
  road = "Rue Alphonse Weicker"
)

struct

# Freeform search.
gisco_address_api_search(q = "alphonse weicker luxembourg")

# Reverse geocoding.
reverse <- gisco_address_api_reverse(x = struct$X[1], y = struct$Y[1])

reverse

# Most populated census grid cell in Madrid.
gisco_address_api_most_populated_cell(city = "Madrid")
```

---

gisco_attributions *Attribution for administrative and statistical GISCO data*

Description

Get the attribution text to use for administrative and statistical data downloaded from GISCO. See the **Copyright** section.

For other datasets you may follow the Eurostat general copyright notice and license policy.

Usage

```
gisco_attributions(lang = "en", copyright = FALSE)
```

Arguments

lang	A character string with the language (two-letter ISO code). See countrycode::codelist and **Details**.
copyright	A logical value indicating whether to display the copyright notice on the console.

Details

Supported languages are:

- "en": English.
- "da": Danish.
- "de": German.
- "es": Spanish.
- "fi": Finnish.
- "fr": French.
- "no": Norwegian.
- "sv": Swedish.

Please consider contributing if you spot a mistake or want to add a new language.

Value

A string with the attribution to be used.

Copyright

The provisions described in this section apply to administrative and statistical data provided by the following functions:

Administrative units

- gisco_get_communes().
- gisco_get_countries().

Statistical units

- gisco_get_census().
- gisco_get_coastal_lines().
- gisco_get_lau().
- gisco_get_nuts().
- gisco_get_urban_audit().

Postal code data use a separate CC-BY-SA-4.0 attribution. See gisco_get_postal_codes().

Copyright notice:

When data downloaded from GISCO is used in any printed or electronic publication, in addition to any other provisions applicable to the whole Eurostat website, the data source must be acknowledged in the legend of the map and on the introductory page of the publication with the following copyright notice:

- EN: © EuroGeographics for the administrative boundaries.
- FR: © EuroGeographics pour les limites administratives.
- DE: © EuroGeographics bezüglich der Verwaltungsgrenzen.

For publications in languages other than English, French or German, the translation of the copyright notice in the language of the publication shall be used.

If you intend to use the data commercially, please contact EuroGeographics for information about their license agreements.

Examples

```
gisco_attributions()

gisco_attributions(lang = "es", copyright = TRUE)

gisco_attributions(lang = "XXX")

# Get the list of codes from countrycode.
if (requireNamespace("dplyr", quietly = TRUE)) {
  library(dplyr)

  countrycode::codelist |>
    select(country.name.en, iso2c)
}
```

`gisco_bulk_download` *GISCO geodata bulk download*

Description

Download zipped data from GISCO to the `cache_dir` and extract the relevant files.

Usage

```
gisco_bulk_download(
  id = c("countries", "coastal_lines", "communes", "lau", "nuts", "urban_audit",
    "postal_codes"),
  year = 2016,
  cache_dir = NULL,
  update_cache = FALSE,
  verbose = FALSE,
  resolution = 10,
```

```
  ext = c("shp", "geojson", "svg", "json", "gdb"),
  recursive = deprecated(),
  ...
)
```

Arguments

id	A character string or numeric value with the dataset type to download, see **Details**. Supported values are: • "countries". • "coastal_lines". • "communes". • "lau". • "nuts". • "urban_audit". • "postal_codes". This argument replaces the previous (deprecated) argument id_giscoR.
year	A character string or numeric value with the release year of the file, see **Details**.
cache_dir	A character string with a path to a cache directory. See the **Caching strategies** section in gisco_set_cache_dir().
update_cache	A logical value indicating whether to refresh the cached file. Defaults to FALSE. When set to TRUE, it forces a new download.
verbose	A logical value indicating whether to display informational messages.
resolution	A character string or numeric value with the geospatial data resolution. One of: • "60": 1:60 million. • "20": 1:20 million. • "10": 1:10 million. • "03": 1:3 million. • "01": 1:1 million.
ext	The extension of the file or files to download. Available formats are "shp", "geojson", "svg", "json" and "gdb". See **Details**.
recursive	**[Deprecated]** recursive is no longer supported. It will never perform recursive extraction of child .zip files. This is the case for shp.zip inside the top-level .zip, which will not be unzipped.
...	Ignored. The argument id_giscoR (**[Deprecated]**) is captured via ... and redirected to id with a warning.

Details

Some arguments only apply to a specific value of id. For example, resolution is ignored for values "communes", "lau", "urban_audit" and "postal_codes".

See available years in the corresponding functions:

- gisco_get_countries().

- gisco_get_coastal_lines().
- gisco_get_communes().
- gisco_get_lau().
- gisco_get_nuts().
- gisco_get_urban_audit().
- gisco_get_postal_codes().

The usual extensions used across **giscoR** are "gpkg" and "shp", but other formats are already available on GISCO. After a bulk download, you may need to adjust the default ext value in the corresponding function to connect it with the downloaded files (see **Examples**).

Value

An invisible character vector with the full paths of the extracted files. See **Examples**.

Source

https://gisco-services.ec.europa.eu/distribution/v2/.

See Also

gisco_get_metadata() to inspect available datasets, years and file formats before downloading.

Bulk and single-unit downloads: gisco_get_unit

Examples

```
tmp <- file.path(tempdir(), "testexample")

dest_files <- gisco_bulk_download(
  id = "countries", resolution = 60,
  year = 2024, ext = "geojson",
  cache_dir = tmp
)
# Read one file.
library(sf)
read_sf(dest_files[1]) |> head()

# Connect the function with the downloaded data.

invisible(gisco_get_countries(
  resolution = 60,
  year = 2024, ext = "geojson",
  cache_dir = tmp, verbose = TRUE
))

# The message shows that the file is already cached.

# Clean up.
unlink(tmp, force = TRUE)
```

gisco_clear_cache *Clear your* R*hrefhttps://CRAN.R-project.org/package=giscoR***giscoR** *cache directory*

Description

Use this function with caution. It clears your cached data and configuration, specifically:

- Deletes the **giscoR** config directory (`tools::R_user_dir("giscoR", "config")`).
- Deletes the `cache_dir` directory.
- Deletes the value stored in `Sys.getenv("GISCO_CACHE_DIR")`.

Usage

```
gisco_clear_cache(config = FALSE, cached_data = TRUE, verbose = FALSE)
```

Arguments

`config`	If `TRUE`, delete the configuration folder of **giscoR**.
`cached_data`	If `TRUE`, delete your `cache_dir` and all its contents.
`verbose`	A logical value indicating whether to display informational messages.

Details

Fully resets your cache state as if you had never installed or used **giscoR**.

Value

Invisible. Called for its side effects.

See Also

`tools::R_user_dir()`

Cache management utilities: `gisco_set_cache_dir()`

Examples

```
# Do not run this. It modifies your current state.
## Not run:
my_cache <- gisco_detect_cache_dir()

# Set an example cache.
ex <- file.path(tempdir(), "example", "cache")
gisco_set_cache_dir(ex, verbose = FALSE)

# Restore the initial cache.
gisco_clear_cache(verbose = TRUE)
```

```
gisco_set_cache_dir(my_cache)
identical(my_cache, gisco_detect_cache_dir())

## End(Not run)
```

---

gisco_coastal_lines *Coastal lines 2016 sf object*

Description

This object contains the coastal lines of the world.

Format

An sf object with POLYGON geometries, resolution: 1:20 million and EPSG:4326.

Copyright

See the GISCO statistical unit copyright provisions: https://ec.europa.eu/eurostat/web/gisco/geodata/statistical-units.

Note

Check the download and usage provisions in gisco_attributions().

Source

COAS_RG_20M_2016_4326.gpkg file.

See Also

gisco_get_coastal_lines()

Included datasets: gisco_countries_2024, gisco_countrycode, gisco_db, gisco_nuts_2024

Examples

```
library(sf)
data("gisco_coastal_lines")
gisco_coastal_lines
```

gisco_countries_2024 *Countries 2024 sf object*

Description

This object contains world administrative boundaries at the country level.

Format

An sf object with MULTIPOLYGON geometries, resolution: 1:20 million and EPSG:4326. It has 263 rows and 12 variables:

CNTR_ID Country ID from Eurostat.

CNTR_NAME Official country name in the local language.

NAME_ENGL Country name in English.

NAME_FREN Country name in French.

ISO3_CODE ISO 3166-1 alpha-3 code of each country from GISCO.

SVRG_UN Sovereign status according to the United Nations.

CAPT Capital city.

EU_STAT European Union member.

EFTA_STAT EFTA member.

CC_STAT EU candidate member.

NAME_GERM Country name in German.

geometry Geometry field.

Copyright

See the GISCO administrative unit copyright provisions: https://ec.europa.eu/eurostat/web/gisco/geodata/administrative-units.

Note

Check the download and usage provisions in gisco_attributions().

Source

CNTR_RG_20M_2024_4326.gpkg file.

See Also

gisco_get_countries()

Included datasets: gisco_coastal_lines, gisco_countrycode, gisco_db, gisco_nuts_2024

Examples

```
data("gisco_countries_2024")
head(gisco_countries_2024)
```

`gisco_countrycode` *Database with different country code schemes and world regions*

Description

A tibble containing conversions between country code schemes (Eurostat, ISO 3166-1 alpha-2 and ISO 3166-1 alpha-3) and geographic regions from the World Bank and the UN (M49 Standard). This database was extracted from the **countrycode** package.

Format

A data frame with 249 rows and 13 variables:

`ISO3_CODE` ISO 3166-1 alpha-3 code of each country.

`CNTR_CODE` Eurostat country code.

`iso2c` ISO 3166-1 alpha-2 code of each country.

`iso.name.en` ISO English short name.

`cldr.short.en` English short name as provided by the Unicode Common Locale Data Repository.

`continent` Continent from the World Bank.

`un.region.code` UN M49 numeric region code.

`un.region.name` UN M49 region name.

`un.regionintermediate.code` UN M49 numeric intermediate region code.

`un.regionintermediate.name` UN M49 intermediate region name.

`un.regionsub.code` UN M49 numeric subregion code.

`un.regionsub.name` UN M49 subregion name.

`eu` Logical value indicating whether the country belongs to the European Union.

World regions

Regions follow the UN geographic regions (see https://unstats.un.org/unsd/methodology/m49/). Under this scheme Cyprus is assigned to Asia.

Source

countrycode::codelist **v1.6.1**.

See Also

gisco_get_countries(), countrycode::codelist.

See also Unicode Common Locale Data Repository.

Included datasets: gisco_coastal_lines, gisco_countries_2024, gisco_db, gisco_nuts_2024

Examples

```
data("gisco_countrycode")
dplyr::glimpse(gisco_countrycode)
```

---

gisco_db *Cached GISCO database*

Description

Database with the list of files in the GISCO geodata distribution as of 2026-10-29.

Format

A tibble with 11,173 rows.

Details

This database is used to redirect the corresponding functions to the correct API endpoints.

This version of the database is used if an update fails. Please use gisco_get_cached_db() with update_cache = TRUE to update the corresponding API endpoints.

Source

GISCO geodata distribution datasets.json.

See Also

gisco_get_metadata() and gisco_get_cached_db().

Included datasets: gisco_coastal_lines, gisco_countries_2024, gisco_countrycode, gisco_nuts_2024

GISCO database and metadata: gisco_get_cached_db(), gisco_get_metadata()

Examples

```
data("gisco_db")
gisco_db |>
  dplyr::glimpse()
```

gisco_get_airports *Airports dataset*

Description

This function accesses the GISCO airport and heliport datasets. Airports are identified using International Civil Aviation Organization (ICAO) airport codes.

Usage

```
gisco_get_airports(
  year = c(2024, 2013, 2006),
  country = NULL,
  cache_dir = NULL,
  update_cache = FALSE,
  verbose = FALSE
)
```

Arguments

year	A character string or numeric value with the release year of the file. One of 2024, 2013, 2006.
country	A character vector of country codes. It can be either a vector of country names, a vector of ISO 3166-1 alpha-3 country codes or a vector of Eurostat country codes. See also countrycode::countrycode().
cache_dir	A character string with a path to a cache directory. See the **Caching strategies** section in gisco_set_cache_dir().
update_cache	A logical value indicating whether to refresh the cached file. Defaults to FALSE. When set to TRUE, it forces a new download.
verbose	A logical value indicating whether to display informational messages.

Details

The returned object is transformed to EPSG:4326.

Value

An sf object.

Copyright

See the Eurostat general copyright and license provisions: https://ec.europa.eu/eurostat/web/gisco/geodata.

Source

https://ec.europa.eu/eurostat/web/gisco/geodata/transport-networks.

See Also

Transport network datasets: gisco_get_ports()

Examples

```
airp <- gisco_get_airports(year = 2024)
coast <- giscoR::gisco_get_countries(year = 2024)

if (!is.null(airp) && requireNamespace("ggplot2", quietly = TRUE)) {
  library(ggplot2)

  ggplot(coast) +
    geom_sf(fill = "grey10", color = "grey20") +
    geom_sf(
      data = airp, color = "#00F0FF",
      size = 0.2, alpha = 0.25
    ) +
    theme_void() +
    theme(
      plot.background = element_rect(fill = "black"),
      text = element_text(color = "white"),
      panel.grid = element_blank(),
      plot.title = element_text(face = "bold", hjust = 0.5),
      plot.subtitle = element_text(face = "italic", hjust = 0.5)
    ) +
    labs(
      title = "Airports in Europe", subtitle = "Year 2024",
      caption = "Source: Eurostat, Airports 2024 dataset."
    ) +
    # Center on Europe with EPSG 3035.
    coord_sf(
      crs = 3035,
      xlim = c(2377294, 7453440),
      ylim = c(1313597, 5628510)
    )
}
```

---

gisco_get_cached_db *Retrieve and update the GISCO database used by* R*hrefhttps://CRAN.R-project.org/package=giscoR***giscoR**

Description

Returns or optionally updates the cached database with endpoints from the GISCO geodata distribution.

Usage

```
gisco_get_cached_db(update_cache = FALSE)
```

Arguments

update_cache	A logical value. If TRUE, rebuild the cached database with the most recent information from the GISCO geodata distribution.

Details

The cached database is stored in the **giscoR** cache path. See gisco_set_cache_dir() for details. The cached database is used in subsequent R sessions.

On new GISCO data releases, you can access the updated data by refreshing the cached database without waiting for a new version of **giscoR**.

A static database gisco_db is shipped with the package. This database is used if there is any problem during the update.

Value

A tibble.

Source

https://gisco-services.ec.europa.eu/distribution/v2/.

See Also

GISCO database and metadata: gisco_db, gisco_get_metadata()

Examples

```
gisco_get_cached_db() |>
  dplyr::glimpse()
```

---

gisco_get_census *Census dataset*

Description

This dataset provides 2011 census units as region or point geometries.

Usage

```
gisco_get_census(
  year = 2011,
  cache_dir = NULL,
  update_cache = FALSE,
  verbose = FALSE,
  spatialtype = c("RG", "PT")
)
```

Arguments

year	A character string or numeric value with the release year of the file. Currently only "2011" is provided.
cache_dir	A character string with a path to a cache directory. See the **Caching strategies** section in gisco_set_cache_dir().
update_cache	A logical value indicating whether to refresh the cached file. Defaults to FALSE. When set to TRUE, it forces a new download.
verbose	A logical value indicating whether to display informational messages.
spatialtype	The type of geometry to return: • "PT": Points - POINT object. • "RG": Regions - MULTIPOLYGON/POLYGON object.

Value

An sf object.

Copyright

See the GISCO statistical unit copyright provisions: https://ec.europa.eu/eurostat/web/gisco/geodata/statistical-units.

Note

Check the download and usage provisions in gisco_attributions().

Source

https://ec.europa.eu/eurostat/web/gisco/geodata/statistical-units/census.

See Also

See gisco_id_api_census_grid() to download via the GISCO ID service API.

Statistical unit datasets: gisco_get_coastal_lines(), gisco_get_lau(), gisco_get_nuts(), gisco_get_urban_audit()

Examples

```
library(sf)

pts <- gisco_get_census(spatialtype = "PT")

pts
```

`gisco_get_coastal_lines` *Coastal lines dataset*

Description

Download global coastal lines.

Usage

```
gisco_get_coastal_lines(
  year = 2016,
  epsg = 4326,
  cache = TRUE,
  update_cache = FALSE,
  cache_dir = NULL,
  verbose = FALSE,
  resolution = 20,
  ext = "gpkg"
)
```

Arguments

`year`	A character string or numeric value with the release year of the file. One of `"2016"`, `"2013"`, `"2010"`, `"2006"`.
`epsg`	A character string or numeric value with the coordinate reference system as a 4-digit EPSG code. One of: • `"4326"`: WGS84. • `"3035"`: ETRS89 / ETRS-LAEA. • `"3857"`: Pseudo-Mercator.
`cache`	A logical value indicating whether to cache results. Defaults to `TRUE`. See the **Caching strategies** section in `gisco_set_cache_dir()`.
`update_cache`	A logical value indicating whether to refresh the cached file. Defaults to `FALSE`. When set to `TRUE`, it forces a new download.
`cache_dir`	A character string with a path to a cache directory. See the **Caching strategies** section in `gisco_set_cache_dir()`.
`verbose`	A logical value indicating whether to display informational messages.
`resolution`	A character string or numeric value with the geospatial data resolution. One of: • `"60"`: 1:60 million. • `"20"`: 1:20 million. • `"10"`: 1:10 million. • `"03"`: 1:3 million. • `"01"`: 1:1 million.
`ext`	A character string with the extension of the file (default `"gpkg"`). One of `"shp"`, `"gpkg"`, `"geojson"`.

Value

An sf object.

Copyright

See the GISCO statistical unit copyright provisions: https://ec.europa.eu/eurostat/web/gisco/geodata/statistical-units.

Note

Check the download and usage provisions in gisco_attributions().

Source

GISCO coastal lines distribution API: https://gisco-services.ec.europa.eu/distribution/v2/coas/.

See Also

gisco_coastal_lines.

See gisco_bulk_download() to perform a bulk download of datasets.

Statistical unit datasets: gisco_get_census(), gisco_get_lau(), gisco_get_nuts(), gisco_get_urban_audit()

Examples

```
coast <- gisco_get_coastal_lines()

if (requireNamespace("ggplot2", quietly = TRUE)) {
  library(ggplot2)

  ggplot(coast) +
    geom_sf(color = "#1278AB", fill = "#FDFBEA") +
    # Zoom on the Mediterranean Sea.
    coord_sf(
      xlim = c(-4, 35),
      ylim = c(31, 45)
    ) +
    theme_minimal() +
    theme(
      panel.background = element_rect(fill = "#C7E7FB", color = NA),
      panel.border = element_rect(colour = "black", fill = NA)
    )
}
```

gisco_get_communes *Communes dataset*

Description

This dataset provides pan-European administrative boundaries at commune level. Communes may correspond to municipalities represented in the separate Local Administrative Units dataset. See gisco_get_lau().

Usage

```
gisco_get_communes(
  year = 2016,
  epsg = 4326,
  cache = deprecated(),
  update_cache = FALSE,
  cache_dir = NULL,
  verbose = FALSE,
  spatialtype = "RG",
  country = NULL,
  ext = "shp"
)
```

Arguments

year	A character string or numeric value with the release year of the file. One of "2016", "2013", "2010", "2008", "2006", "2004", "2001".
epsg	A character string or numeric value with the coordinate reference system as a 4-digit EPSG code. One of: • "4326": WGS84. • "3035": ETRS89 / ETRS-LAEA. • "3857": Pseudo-Mercator.
cache	**[Deprecated]**. These functions always cache the result because of its size. See the **Caching strategies** section in gisco_set_cache_dir().
update_cache	A logical value indicating whether to refresh the cached file. Defaults to FALSE. When set to TRUE, it forces a new download.
cache_dir	A character string with a path to a cache directory. See the **Caching strategies** section in gisco_set_cache_dir().
verbose	A logical value indicating whether to display informational messages.
spatialtype	A character string with the type of geometry to return. Available values are: • "RG": Regions - MULTIPOLYGON/POLYGON object. • "LB": Labels - POINT object. • "BN": Boundaries - LINESTRING object. Argument country is only applied when spatialtype is "RG" or "LB".

country A character vector of country codes. It can be either a vector of country names, a vector of ISO 3166-1 alpha-3 country codes or a vector of Eurostat country codes. See also countrycode::countrycode().

ext A character string with the extension of the file (default "shp"). One of "shp", "gpkg", "geojson".

Details

The Nomenclature of Territorial Units for Statistics (NUTS) and the LAU nomenclature are hierarchical classifications of statistical regions that together subdivide the EU economic territory into statistical regions and local units, moving from larger to smaller territorial units: NUTS 1, 2 and 3 and LAU.

The dataset is based on EuroBoundaryMap from EuroGeographics. Geographic coverage includes the European Union 28, EFTA countries and candidate countries. The scale of the dataset is 1:100 000.

The LAU classification is not covered by any legislative act.

Value

An sf object.

Copyright

See the GISCO administrative unit copyright provisions: https://ec.europa.eu/eurostat/web/gisco/geodata/administrative-units.

Note

Check the download and usage provisions in gisco_attributions().

Source

GISCO communes distribution API: https://gisco-services.ec.europa.eu/distribution/v2/communes/.

See Also

gisco_get_lau().

See gisco_bulk_download() to perform a bulk download of datasets.

Administrative unit datasets: gisco_get_countries(), gisco_get_postal_codes()

Examples

```
ire_comm <- gisco_get_communes(spatialtype = "LB", country = "Ireland")

if (!is.null(ire_comm) && requireNamespace("ggplot2", quietly = TRUE)) {
  library(ggplot2)

  ggplot(ire_comm) +
```

```
    geom_sf(shape = 21, col = "#009A44", size = 0.5) +
    labs(
      title = "Communes in Ireland",
      subtitle = "Year 2016",
      caption = gisco_attributions()
    ) +
    theme_void() +
    theme(text = element_text(
      colour = "#009A44",
      family = "serif", face = "bold"
    ))
}
```

`gisco_get_countries` *Countries dataset*

Description

This dataset contains world administrative boundaries at the country level. It provides two feature classes, regions and boundaries, at five scale levels: 1M, 3M, 10M, 20M and 60M.

Downloads data from the aggregated GISCO country file. To download single-unit country files, use `gisco_get_unit_country()`.

Usage

```
gisco_get_countries(
  year = 2024,
  epsg = 4326,
  cache = TRUE,
  update_cache = FALSE,
  cache_dir = NULL,
  verbose = FALSE,
  resolution = 20,
  spatialtype = "RG",
  country = NULL,
  region = NULL,
  ext = "gpkg"
)
```

Arguments

`year`	A character string or numeric value with the release year of the file. One of `"2024"`, `"2020"`, `"2016"`, `"2013"`, `"2010"`, `"2006"`, `"2001"`.
`epsg`	A character string or numeric value with the coordinate reference system as a 4-digit EPSG code. One of: • `"4326"`: WGS84.

- "3035": ETRS89 / ETRS-LAEA.
- "3857": Pseudo-Mercator.

cache — A logical value indicating whether to cache results. Defaults to TRUE. See the **Caching strategies** section in gisco_set_cache_dir().

update_cache — A logical value indicating whether to refresh the cached file. Defaults to FALSE. When set to TRUE, it forces a new download.

cache_dir — A character string with a path to a cache directory. See the **Caching strategies** section in gisco_set_cache_dir().

verbose — A logical value indicating whether to display informational messages.

resolution — A character string or numeric value with the geospatial data resolution. One of:

- "60": 1:60 million.
- "20": 1:20 million.
- "10": 1:10 million.
- "03": 1:3 million.
- "01": 1:1 million.

spatialtype — A character string with the type of geometry to return. Available values are:

- "RG": Regions - MULTIPOLYGON/POLYGON object.
- "LB": Labels - POINT object.
- "BN": Boundaries - LINESTRING object.
- "COASTL": Coastal lines - LINESTRING object.
- "INLAND": Inland boundaries - LINESTRING object.

Arguments country and region are only applied when spatialtype is "RG" or "LB".

country — A character vector of country codes. It can be either a vector of country names, a vector of ISO 3166-1 alpha-3 country codes or a vector of Eurostat country codes. See also countrycode::countrycode().

region — An optional character vector of UN M49 region codes or European Union membership. Possible values are "Africa", "Americas", "Asia", "Europe", "Oceania" or "EU" for countries belonging to the European Union as of 2021. See **World regions** and gisco_countrycode.

ext — A character string with the extension of the file (default "gpkg"). One of "shp", "gpkg", "geojson".

Value

An sf object.

World regions

Regions follow the UN geographic regions (see https://unstats.un.org/unsd/methodology/m49/). Under this scheme Cyprus is assigned to Asia.

Copyright

See the GISCO administrative unit copyright provisions: https://ec.europa.eu/eurostat/web/gisco/geodata/administrative-units.

Note

Check the download and usage provisions in gisco_attributions().

Source

GISCO countries distribution API: https://gisco-services.ec.europa.eu/distribution/v2/countries/.

See Also

gisco_countrycode, gisco_countries_2024, gisco_get_metadata(), countrycode::countrycode().

See gisco_bulk_download() to perform a bulk download of datasets.

See gisco_get_unit_country() to download single-unit files.

See gisco_id_api_country() to download via the GISCO ID service API.

Administrative unit datasets: gisco_get_communes(), gisco_get_postal_codes()

Examples

```
cntries <- gisco_get_countries()

library(ggplot2)
ggplot(cntries) +
  geom_sf()

# Get a region.

africa <- gisco_get_countries(region = "Africa")
ggplot(africa) +
  geom_sf(fill = "#078930", col = "white") +
  theme_minimal()
```

---

gisco_get_education *Education services in Europe*

Description

This dataset integrates Member States' official data on the location of education services. Additional information on these services is included when available. See **Details**.

Usage

```
gisco_get_education(
  year = c(2023, 2020),
  cache = TRUE,
  update_cache = FALSE,
  cache_dir = NULL,
```

```
  verbose = FALSE,
  country = NULL
)
```

Arguments

year	A character string or numeric value with the release year of the file. One of 2023, 2020.
cache	A logical value indicating whether to cache results. Defaults to TRUE. See the **Caching strategies** section in gisco_set_cache_dir().
update_cache	A logical value indicating whether to refresh the cached file. Defaults to FALSE. When set to TRUE, it forces a new download.
cache_dir	A character string with a path to a cache directory. See the **Caching strategies** section in gisco_set_cache_dir().
verbose	A logical value indicating whether to display informational messages.
country	A character vector of country codes. It can be either a vector of country names, a vector of ISO 3166-1 alpha-3 country codes or a vector of Eurostat country codes. See also countrycode::countrycode().

Details

Files are distributed in EPSG:4326.

The following table describes the education service attributes:

Attribute	Description
id	The education service identifier, based on national identification codes when available.
name	The name of the educational institution.
site_name	The name of a specific site or branch of the educational institution.
lat	Latitude (WGS 84).
lon	Longitude (WGS 84).
street	Street name.
house_number	House number.
postcode	Postcode.
address	Address information when the source does not separate address components.
city	City name. In some sources, this refers to a region or municipality.
cntr_id	Country code (2 letters, ISO 3166-1 alpha-2).
levels	Education levels represented by a single integer or range, using ISCED 2011.
max_students	Maximum student capacity.
enrollment	Number of enrolled students.
fields	Academic disciplines in which the institution specializes, using ISCED-F 2013.
facility_type	Type of institution by ownership and operation, such as Catholic or international.
public_private	Public or private status of the education service.
tel	Telephone number.
email	Email address.
url	URL for the institution's website.
ref_date	The reference date (DD/MM/YYYY) for the data.
pub_date	The publication date of the dataset by Eurostat (DD/MM/YYYY).

geo_qual	Geolocation quality indicator: 1 = Good, 2 = Medium, 3 = Low, 4 = From source, -1 = Unknown, -2 = No
comments	Additional information on the education service.

Value

An sf object.

Copyright

The general Eurostat copyright and license provisions apply. Detailed metadata also list source-specific licensing conditions by country and data provider. Review the conditions for the selected data before use.

Data quality

The data are extracted from official national registers. They may contain inconsistencies, inaccuracies and gaps due to the heterogeneity of the national input data.

Source

https://ec.europa.eu/eurostat/web/gisco/geodata/basic-services.

See Also

Basic service datasets: gisco_get_healthcare()

Examples

```
edu_austria <- gisco_get_education(country = "Austria", year = 2023)

# Plot if downloaded.
if (!is.null(edu_austria) && requireNamespace("ggplot2", quietly = TRUE)) {
  austria_nuts3 <- gisco_get_nuts(country = "Austria", nuts_level = 3)

  library(ggplot2)
  ggplot(austria_nuts3) +
    geom_sf(fill = "grey10", color = "grey60") +
    geom_sf(
      data = edu_austria, aes(color = rev(public_private)),
      alpha = 0.25
    ) +
    theme_void() +
    theme(
      plot.background = element_rect(fill = "black"),
      text = element_text(color = "white"),
      panel.grid = element_blank(),
      plot.title = element_text(face = "bold", hjust = 0.5),
      plot.subtitle = element_text(face = "italic", hjust = 0.5)
    ) +
    labs(
```

```
    title = "Education", subtitle = "Austria (2023)",
    caption = "Source: Eurostat, Education 2023 dataset.",
    color = "Type"
  ) +
  coord_sf(crs = 3035)
}
```

`gisco_get_grid` *Grid dataset*

Description

These datasets contain grid cells covering the EU and neighboring countries at resolutions from 1 km to 100 km. Population figures are available for selected reference years.

Usage

```
gisco_get_grid(
  resolution = c(100, 50, 20, 10, 5, 2, 1),
  spatialtype = c("REGION", "POINT"),
  cache_dir = NULL,
  update_cache = FALSE,
  verbose = FALSE
)
```

Arguments

`resolution`	The grid cell resolution in km. Available values are `"1"`, `"2"`, `"5"`, `"10"`, `"20"`, `"50"` and `"100"`. See **Details**.
`spatialtype`	A character string selecting `"REGION"` or `"POINT"`.
`cache_dir`	A character string with a path to a cache directory. See the **Caching strategies** section in `gisco_set_cache_dir()`.
`update_cache`	A logical value indicating whether to refresh the cached file. Defaults to `FALSE`. When set to `TRUE`, it forces a new download.
`verbose`	A logical value indicating whether to display informational messages.

Details

`gisco_get_grid()` downloads the GeoPackage representation of the grid as polygon cells (`spatialtype = "REGION"`) or cell-center points (`spatialtype = "POINT"`). The official distribution also provides tabular CSV and Parquet files, which this function does not download.

All grid geometries use `EPSG:3035`. Population columns are named `TOT_P_YYYY`, where `YYYY` is the reference year. To calculate population density, divide a population value by the cell area in square kilometers (`resolution^2`).

The file sizes range from 428 KB (`resolution = 100`) to 1.7 GB (`resolution = 1`).

Value

An sf object.

Copyright

Population variables (TOT_P_*) have year- and country-specific licensing conditions. Other grid elements are covered by the general Eurostat copyright provisions. Review the licensing table and metadata on the official grid page before redistributing or publishing the data.

Source

https://ec.europa.eu/eurostat/web/gisco/geodata/grids.

Examples

```
grid <- gisco_get_grid(resolution = 20)

# Proceed if downloaded correctly.
if (
  !is.null(grid) &&
    requireNamespace("dplyr", quietly = TRUE) &&
    requireNamespace("ggplot2", quietly = TRUE)
) {
  library(dplyr)

  grid <- grid |>
    mutate(popdens = TOT_P_2021 / 20^2)

  breaks <- c(0, 1, 10, 25, 50, 100, 250, 500, 1000, Inf)

  # Cut groups.
  grid <- grid |>
    mutate(popdens_cut = cut(popdens,
      breaks = breaks,
      include.lowest = TRUE
    ))

  cut_labs <- prettyNum(breaks, big.mark = " ")[-1]
  cut_labs[1] <- "0"
  cut_labs[9] <- "> 1000"

  pal <- c(
    "black",
    hcl.colors(length(breaks) - 2, palette = "Spectral", alpha = 0.9)
  )

  library(ggplot2)

  ggplot(grid) +
    geom_sf(aes(fill = popdens_cut), color = NA, linewidth = 0) +
    coord_sf(
      xlim = c(2500000, 7000000),
```

```
    ylim = c(1500000, 5200000)
  ) +
  scale_fill_manual(
    values = pal, na.value = "black",
    name = "",
    labels = cut_labs
  ) +
  theme_void() +
  labs(
    title = "Population density in Europe (2021)",
    subtitle = "Grid: 20 km. People by square km.",
    caption = paste(
      "Source: Eurostat GISCO grid dataset.\n",
      "Review the applicable population-data licence."
    )
  ) +
  theme(
    text = element_text(colour = "white"),
    plot.background = element_rect(fill = "grey2"),
    plot.title = element_text(hjust = 0.5),
    plot.subtitle = element_text(hjust = 0.5, face = "bold"),
    plot.caption = element_text(
      color = "grey60", hjust = 0.5, vjust = 0,
      margin = margin(t = 5, b = 10)
    ),
    legend.key.height = unit(0.5, "lines"),
    legend.key.width = unit(1, "lines")
  )
}
```

---

gisco_get_healthcare *Healthcare services in Europe*

Description

The dataset contains information on healthcare services considered hospitals by Member States. The definition varies slightly from country to country. Broadly, hospitals are licensed establishments primarily engaged in providing medical, diagnostic and treatment services, including physician, nursing and other health services to inpatients and the specialized accommodation services required by inpatients.

Usage

```
gisco_get_healthcare(
  year = c(2023, 2020),
  cache = TRUE,
  update_cache = FALSE,
  cache_dir = NULL,
```

```
  verbose = FALSE,
  country = NULL
)
```

Arguments

year	A character string or numeric value with the release year of the file. One of 2023, 2020.
cache	A logical value indicating whether to cache results. Defaults to TRUE. See the **Caching strategies** section in gisco_set_cache_dir().
update_cache	A logical value indicating whether to refresh the cached file. Defaults to FALSE. When set to TRUE, it forces a new download.
cache_dir	A character string with a path to a cache directory. See the **Caching strategies** section in gisco_set_cache_dir().
verbose	A logical value indicating whether to display informational messages.
country	A character vector of country codes. It can be either a vector of country names, a vector of ISO 3166-1 alpha-3 country codes or a vector of Eurostat country codes. See also countrycode::countrycode().

Details

Files are distributed in EPSG:4326.

The following table describes the healthcare service attributes:

Attribute	Description
id	The healthcare service identifier, based on national identification codes when available.
hospital_name	The name of the healthcare institution.
site_name	The name of a specific site or branch of the healthcare institution.
lat	Latitude (WGS 84).
lon	Longitude (WGS 84).
street	Street name.
house_number	House number.
postcode	Postcode.
address	Address information when the source does not separate address components.
city	City name. In some sources, this refers to a region or municipality.
cntr_id	Country code (2 letters, ISO 3166-1 alpha-2).
emergency	yes or no, indicating whether the healthcare site provides emergency medical services.
cap_beds	Number of beds, the most common capacity measure.
cap_prac	Number of practitioners.
cap_rooms	Number of rooms.
facility_type	Type of healthcare service, such as a psychiatric hospital, based on national classification.
public_private	Public or private status of the healthcare service.
list_specs	List of specialties recognized in the EU and EEA according to the 2005 EU Directive (Annex V).
tel	Telephone number.
email	Email address.
url	URL for the institution's website.
ref_date	The reference date (DD/MM/YYYY) for the data.

pub_date	The publication date of the dataset by Eurostat (DD/MM/YYYY).
geo_qual	Geolocation quality indicator: 1 = Good, 2 = Medium, 3 = Low, 4 = From source, -1 = Unknown, -2 = No
comments	Additional information on the healthcare service.

Value

An sf object.

Copyright

The general Eurostat copyright and license provisions apply. Detailed metadata also list source-specific licensing conditions by country and data provider. Review the conditions for the selected data before use.

Data quality

The data are extracted from official national registers. They may contain inconsistencies, inaccuracies and gaps due to the heterogeneity of the national input data.

Source

https://ec.europa.eu/eurostat/web/gisco/geodata/basic-services.

See Also

Basic service datasets: gisco_get_education()

Examples

```
health_benelux <- gisco_get_healthcare(
  country = c("BE", "NL", "LU"),
  year = 2023
)

# Plot if downloaded.
if (
  !is.null(health_benelux) &&
    requireNamespace("ggplot2", quietly = TRUE)
) {
  benelux <- gisco_get_countries(country = c("BE", "NL", "LU"))

  library(ggplot2)
  ggplot(benelux) +
    geom_sf(fill = "grey10", color = "grey20") +
    geom_sf(
      data = health_benelux, color = "red",
      size = 0.2, alpha = 0.25
    ) +
    theme_void() +
    theme(
```

```
    plot.background = element_rect(fill = "black"),
    text = element_text(color = "white"),
    panel.grid = element_blank(),
    plot.title = element_text(face = "bold", hjust = 0.5),
    plot.subtitle = element_text(face = "italic", hjust = 0.5)
  ) +
  labs(
    title = "Healthcare services", subtitle = "Benelux (2023)",
    caption = "Source: Eurostat, Healthcare 2023 dataset."
  ) +
  coord_sf(crs = 3035)
}
```

`gisco_get_lau` *Local Administrative Units (LAU) dataset*

Description

Local Administrative Units are the building blocks of NUTS and other statistical regions. They comprise municipalities and communes in the European Statistical System. See `gisco_get_communes()` for the separate commune dataset.

Usage

```
gisco_get_lau(
  year = 2024,
  epsg = 4326,
  cache = deprecated(),
  update_cache = FALSE,
  cache_dir = NULL,
  verbose = FALSE,
  country = NULL,
  gisco_id = NULL,
  ext = "gpkg"
)
```

Arguments

`year`	A character string or numeric value with the release year of the file. One of `"2024"`, `"2023"`, `"2022"`, `"2021"`, `"2020"`, `"2019"`, `"2018"`, `"2017"`, `"2016"`, `"2015"`, `"2014"`, `"2013"`, `"2012"`, `"2011"`.
`epsg`	A character string or numeric value with the coordinate reference system as a 4-digit EPSG code. One of: • `"4326"`: WGS84. • `"3035"`: ETRS89 / ETRS-LAEA. • `"3857"`: Pseudo-Mercator.

cache	**[Deprecated]**. Always caches the result due to its size. See the **Caching strategies** section in gisco_set_cache_dir().
update_cache	A logical value indicating whether to refresh the cached file. Defaults to FALSE. When set to TRUE, it forces a new download.
cache_dir	A character string with a path to a cache directory. See the **Caching strategies** section in gisco_set_cache_dir().
verbose	A logical value indicating whether to display informational messages.
country	A character vector of country codes. It can be either a vector of country names, a vector of ISO 3166-1 alpha-3 country codes or a vector of Eurostat country codes. See also countrycode::countrycode().
gisco_id	An optional character vector of GISCO_ID LAU values.
ext	A character string with the extension of the file (default "gpkg"). One of "shp", "gpkg", "geojson".

Details

The Nomenclature of Territorial Units for Statistics (NUTS) and the LAU nomenclature are hierarchical classifications of statistical regions that together subdivide the EU economic territory into statistical regions and local units, moving from larger to smaller territorial units: NUTS 1, 2 and 3 and LAU.

The LAU classification is not covered by any legislative act. Geographical extent covers the European Union, EFTA countries and candidate countries. The scale of the dataset is 1:100 000.

The data contain the National Statistical Agency LAU code, which can be joined to LAU lists. They also contain a GISCO_ID field, which is a unique identifier consisting of the country code and LAU code.

Total resident population figures (31 December) have also been added in some versions based on the associated LAU lists.

Value

An sf object.

Copyright

See the GISCO statistical unit copyright provisions: https://ec.europa.eu/eurostat/web/gisco/geodata/statistical-units.

Note

Check the download and usage provisions in gisco_attributions().

Source

GISCO Local Administrative Units distribution API: https://gisco-services.ec.europa.eu/distribution/v2/lau/.

See Also

gisco_get_communes().

See gisco_bulk_download() to perform a bulk download of datasets.

See gisco_id_api_lau() to download via the GISCO ID service API.

Statistical unit datasets: gisco_get_census(), gisco_get_coastal_lines(), gisco_get_nuts(), gisco_get_urban_audit()

Examples

```
## Not run:

lu_lau <- gisco_get_lau(year = 2024, country = "Luxembourg")

if (!is.null(lu_lau) && requireNamespace("ggplot2", quietly = TRUE)) {
  library(ggplot2)

  ggplot(lu_lau) +
    geom_sf(aes(fill = POP_DENS_2024)) +
    labs(
      title = "Population density in Luxembourg",
      subtitle = "Year 2024",
      caption = gisco_attributions()
    ) +
    scale_fill_viridis_b(
      option = "cividis",
      label = \(x) prettyNum(x, big.mark = ",")
    ) +
    theme_void() +
    labs(fill = "pop/km2")
}

## End(Not run)
```

---

gisco_get_metadata *Get metadata*

Description

Get a table with names and IDs of administrative and statistical units.

Usage

```
gisco_get_metadata(
  id = c("nuts", "countries", "urban_audit"),
  year = 2024,
  verbose = FALSE
)
```

Arguments

id	A character string with the unit type to download. Accepted values are "nuts", "countries" or "urban_audit".
year	A character string or numeric value with the release year of the metadata.
verbose	A logical value indicating whether to display informational messages.

Value

A tibble.

Source

https://gisco-services.ec.europa.eu/distribution/v2/.

See Also

gisco_bulk_download() for bulk downloads and gisco_get_unit for single-unit downloads.

GISCO database and metadata: gisco_db, gisco_get_cached_db()

Examples

```
cities <- gisco_get_metadata(id = "urban_audit", year = 2020)

cities
```

---

gisco_get_nuts *NUTS statistical units dataset*

Description

The GISCO statistical unit dataset represents the Nomenclature of Territorial Units for Statistics (NUTS) and statistical regions using multipart polygon, polyline and point topology. The NUTS geographical information is completed by attribute tables and a set of cartographic help lines to better visualize multipart polygonal regions.

NUTS is a hierarchical system divided into three levels:

- NUTS 1: major socio-economic regions.
- NUTS 2: basic regions for the application of regional policies.
- NUTS 3: small regions for specific diagnoses.

There is also a NUTS 0 level, which usually corresponds to national boundaries.

Downloads data from the aggregated GISCO NUTS file, which contains data for all countries at the requested NUTS level or levels. To download single-unit NUTS files, use gisco_get_unit_nuts().

Usage

```
gisco_get_nuts(
  year = 2024,
  epsg = 4326,
  cache = TRUE,
  update_cache = FALSE,
  cache_dir = NULL,
  verbose = FALSE,
  resolution = 20,
  spatialtype = "RG",
  country = NULL,
  nuts_id = NULL,
  nuts_level = c("all", "0", "1", "2", "3"),
  ext = "gpkg"
)
```

Arguments

year	A character string or numeric value with the release year of the file. One of "2024", "2021", "2016", "2013", "2010", "2006", "2003".
epsg	A character string or numeric value with the coordinate reference system as a 4-digit EPSG code. One of: • "4326": WGS84. • "3035": ETRS89 / ETRS-LAEA. • "3857": Pseudo-Mercator.
cache	A logical value indicating whether to cache results. Defaults to TRUE. See the **Caching strategies** section in gisco_set_cache_dir().
update_cache	A logical value indicating whether to refresh the cached file. Defaults to FALSE. When set to TRUE, it forces a new download.
cache_dir	A character string with a path to a cache directory. See the **Caching strategies** section in gisco_set_cache_dir().
verbose	A logical value indicating whether to display informational messages.
resolution	A character string or numeric value with the geospatial data resolution. One of: • "60": 1:60 million. • "20": 1:20 million. • "10": 1:10 million. • "03": 1:3 million. • "01": 1:1 million.
spatialtype	A character string with the type of geometry to return. Available values are: • "RG": Regions - MULTIPOLYGON/POLYGON object. • "LB": Labels - POINT object. • "BN": Boundaries - LINESTRING object. Arguments country, nuts_level and nuts_id are only applied when spatialtype is "RG" or "LB".

country	A character vector of country codes. It can be either a vector of country names, a vector of ISO 3166-1 alpha-3 country codes or a vector of Eurostat country codes. See also countrycode::countrycode().
nuts_id	An optional character vector of NUTS IDs.
nuts_level	A character string with the NUTS level. One of 0, 1, 2, 3 or all for all levels.
ext	A character string with the extension of the file (default "gpkg"). One of "shp", "gpkg", "geojson".

Details

The NUTS nomenclature is a hierarchical classification of statistical regions and subdivides the EU economic territory into regions of three different levels (NUTS 1, 2 and 3, moving respectively from larger to smaller territorial units). NUTS 1 is the most aggregated level. Additional country-level NUTS 0 data are also available for countries where the statistical national level does not coincide with the administrative boundaries.

NUTS legislation is amended periodically, so GISCO provides multiple classification vintages. An unofficial NUTS-like classification is defined for EFTA countries, candidate countries and potential candidates based on bilateral agreements between Eurostat and the respective statistical agencies.

An introduction to the NUTS classification is available here: https://ec.europa.eu/eurostat/web/nuts/overview.

Value

An sf object.

Copyright

See the GISCO statistical unit copyright provisions: https://ec.europa.eu/eurostat/web/gisco/geodata/statistical-units.

Note

Check the download and usage provisions in gisco_attributions().

Source

GISCO NUTS distribution API: https://gisco-services.ec.europa.eu/distribution/v2/nuts/.

See Also

gisco_nuts_2024, eurostat::get_eurostat_geospatial().

See gisco_bulk_download() to perform a bulk download of datasets.

See gisco_get_unit_nuts() to download single-unit files.

See gisco_id_api_nuts() to download via the GISCO ID service API.

Statistical unit datasets: gisco_get_census(), gisco_get_coastal_lines(), gisco_get_lau(), gisco_get_urban_audit()

Examples

```
nuts2 <- gisco_get_nuts(nuts_level = 2)

library(ggplot2)

ggplot(nuts2) +
  geom_sf() +
  # ETRS89 / ETRS-LAEA
  coord_sf(
    crs = 3035, xlim = c(2377294, 7453440),
    ylim = c(1313597, 5628510)
  ) +
  labs(title = "NUTS 2 levels")
# NUTS 3 for Germany.
germany_nuts3 <- gisco_get_nuts(nuts_level = 3, country = "Germany")

ggplot(germany_nuts3) +
  geom_sf() +
  labs(
    title = "NUTS 3 levels",
    subtitle = "Germany",
    caption = gisco_attributions()
  )

# Select specific regions
select_nuts <- gisco_get_nuts(nuts_id = c("ES2", "FRJ", "FRL", "ITC"))

ggplot(select_nuts) +
  geom_sf(aes(fill = CNTR_CODE)) +
  scale_fill_viridis_d()
```

---

gisco_get_ports *Ports dataset*

Description

This dataset includes port locations worldwide. The ports are identified following the UN LOCODE list.

Usage

```
gisco_get_ports(
  year = c(2013, 2009),
  country = NULL,
  cache_dir = NULL,
  update_cache = FALSE,
  verbose = FALSE
)
```

Arguments

year	A character string or numeric value with the release year of the file. One of 2013, 2009.
country	A character vector of country codes. It can be either a vector of country names, a vector of ISO 3166-1 alpha-3 country codes or a vector of Eurostat country codes. See also countrycode::countrycode().
cache_dir	A character string with a path to a cache directory. See the **Caching strategies** section in gisco_set_cache_dir().
update_cache	A logical value indicating whether to refresh the cached file. Defaults to FALSE. When set to TRUE, it forces a new download.
verbose	A logical value indicating whether to display informational messages.

Details

The returned object is transformed to EPSG:4326.

gisco_get_ports() adds a new field, CNTR_ISO2, to identify the country of the port.

Value

An sf object.

Copyright

See the Eurostat general copyright and license provisions: https://ec.europa.eu/eurostat/web/gisco/geodata.

Source

https://ec.europa.eu/eurostat/web/gisco/geodata/transport-networks.

See Also

Transport network datasets: gisco_get_airports()

Examples

```
library(sf)

ports <- gisco_get_ports(2013)
coast <- giscoR::gisco_coastal_lines

if (!is.null(ports) && requireNamespace("ggplot2", quietly = TRUE)) {
  library(ggplot2)

  ggplot(coast) +
    geom_sf(fill = "grey10", color = "grey20") +
    geom_sf(
      data = ports, color = "#6bb857",
      size = 0.2, alpha = 0.25
```

```
  ) +
  theme_void() +
  theme(
    plot.background = element_rect(fill = "black"),
    text = element_text(color = "white"),
    panel.grid = element_blank(),
    plot.title = element_text(face = "bold", hjust = 0.5),
    plot.subtitle = element_text(face = "italic", hjust = 0.5)
  ) +
  labs(
    title = "Ports worldwide", subtitle = "Year 2013",
    caption = "Source: Eurostat, Ports 2013 dataset."
  ) +
  coord_sf(crs = "ESRI:54030")
}
```

`gisco_get_postal_codes` *Postal codes dataset*

Description

The postal code point dataset shows the location of postal codes, NUTS codes and the degree of urbanization classification across the EU, EFTA and candidate countries. Its primary purpose is to create correspondence tables for the NUTS classification established by Regulation (EC) No 1059/2003 as part of the Tercet Regulation (EU) 2017/2391.

Usage

```
gisco_get_postal_codes(
  year = 2025,
  epsg = 4326,
  country = NULL,
  cache_dir = NULL,
  update_cache = FALSE,
  verbose = FALSE,
  ext = "gpkg"
)
```

Arguments

`year`	A character string or numeric value with the release year of the file. One of `"2025"`, `"2024"`, `"2020"`.
`epsg`	A character string or numeric value with the coordinate reference system as a 4-digit EPSG code. One of: • `"4326"`: WGS84. • `"3035"`: ETRS89 / ETRS-LAEA.

- "3857": Pseudo-Mercator.

country — A character vector of country codes. It can be either a vector of country names, a vector of ISO 3166-1 alpha-3 country codes or a vector of Eurostat country codes. See also countrycode::countrycode().

cache_dir — A character string with a path to a cache directory. See the **Caching strategies** section in gisco_set_cache_dir().

update_cache — A logical value indicating whether to refresh the cached file. Defaults to FALSE. When set to TRUE, it forces a new download.

verbose — A logical value indicating whether to display informational messages.

ext — A character string with the extension of the file (default "gpkg"). One of "shp", "gpkg", "geojson".

Details

The GISCO distribution API provides postal code releases for 2025, 2024 and 2020. The 2025 release has a reference date of 1 January 2025.

Value

An sf object.

Copyright

The dataset is released under the CC-BY-SA-4.0 license. Although the distribution API provides a 2025 release, the official GISCO licensing page currently requires the following attribution: © European Union - GISCO, 2024, postal code point dataset, Licence CC-BY-SA 4.0.

Note

This dataset is not covered by gisco_attributions(). Use the attribution specified above until GISCO publishes revised licensing text.

Non-geographical postal codes, such as post boxes and codes used by large organizations, are not included. The dataset may omit or incorrectly locate postal codes because the source data vary considerably among countries.

Source

GISCO administrative units: https://ec.europa.eu/eurostat/web/gisco/geodata/administrative-units.

GISCO postal code distribution API: https://gisco-services.ec.europa.eu/distribution/v2/pcode/.

See Also

See gisco_bulk_download() to perform a bulk download of datasets.

Administrative unit datasets: gisco_get_communes(), gisco_get_countries()

Examples

```
# Large download.
## Not run:

pc_bel <- gisco_get_postal_codes(year = 2025, country = "BE")

if (!is.null(pc_bel) && requireNamespace("ggplot2", quietly = TRUE)) {
  library(ggplot2)

  ggplot(pc_bel) +
    geom_sf(color = "gold") +
    theme_bw() +
    labs(
      title = "Postcodes of Belgium",
      subtitle = "2025",
      caption = paste("\u00a9 European Union - GISCO, 2024,",
        "postal code point dataset",
        "Licence CC-BY-SA 4.0",
        sep = "\n"
      )
    )
}

## End(Not run)
```

---

gisco_get_unit *GISCO geodata single-unit download*

Description

Download datasets of single spatial units from GISCO to the cache_dir.

Unlike gisco_get_countries(), gisco_get_nuts() or gisco_get_urban_audit() (which download full datasets and apply filters), these functions download a single-unit file, reducing the time needed to download and read data into your R session.

Usage

```
gisco_get_unit_country(
  unit = "ES",
  year = 2024,
  epsg = c(4326, 3857, 3035),
  cache = TRUE,
  update_cache = FALSE,
  cache_dir = NULL,
  verbose = FALSE,
  resolution = c(1, 3, 10, 20, 60),
```

```
  spatialtype = c("RG", "LB")
)

gisco_get_unit_nuts(
  unit = "ES416",
  year = 2024,
  epsg = c(4326, 3857, 3035),
  cache = TRUE,
  update_cache = FALSE,
  cache_dir = NULL,
  verbose = FALSE,
  resolution = c(1, 3, 10, 20, 60),
  spatialtype = c("RG", "LB")
)

gisco_get_unit_urban_audit(
  unit = "ES001F",
  year = 2024,
  epsg = c(4326, 3857, 3035),
  cache = TRUE,
  update_cache = FALSE,
  cache_dir = NULL,
  verbose = FALSE,
  spatialtype = c("RG", "LB")
)
```

Arguments

unit	A character vector of unit IDs to download. See **Details**.
year	A character string or numeric value with the release year of the file.
epsg	A character string or numeric value with the coordinate reference system as a 4-digit EPSG code. One of: • "4326": WGS84. • "3035": ETRS89 / ETRS-LAEA. • "3857": Pseudo-Mercator.
cache	A logical value indicating whether to cache results. Defaults to TRUE. See the **Caching strategies** section in gisco_set_cache_dir().
update_cache	A logical value indicating whether to refresh the cached file. Defaults to FALSE. When set to TRUE, it forces a new download.
cache_dir	A character string with a path to a cache directory. See the **Caching strategies** section in gisco_set_cache_dir().
verbose	A logical value indicating whether to display informational messages.
resolution	A character string or numeric value with the geospatial data resolution. One of: • "60": 1:60 million. • "20": 1:20 million.

- "10": 1:10 million.
- "03": 1:3 million.
- "01": 1:1 million.

spatialtype A character string with the type of geometry to return. Available values are:

- "RG": Regions - MULTIPOLYGON/POLYGON object.
- "LB": Labels - POINT object.

Details

Check the available unit IDs for the required argument combination with gisco_get_metadata().

Value

An sf object.

Copyright

See the GISCO copyright provisions for administrative and statistical units:

- Administrative units: https://ec.europa.eu/eurostat/web/gisco/geodata/administrative-units.
- Statistical units: https://ec.europa.eu/eurostat/web/gisco/geodata/statistical-units.

Note

Check the download and usage provisions in gisco_attributions().

Source

GISCO countries distribution API: https://gisco-services.ec.europa.eu/distribution/v2/countries/.

GISCO NUTS distribution API: https://gisco-services.ec.europa.eu/distribution/v2/nuts/.

GISCO Urban Audit distribution API: https://gisco-services.ec.europa.eu/distribution/v2/urau/.

All source files are .geojson files.

See Also

gisco_get_metadata(), gisco_get_countries(), gisco_get_nuts(), gisco_get_urban_audit().

See gisco_id_api to download via the GISCO ID service API.

Bulk and single-unit downloads: gisco_bulk_download()

Examples

```
# Get metadata.
cities <- gisco_get_metadata("urban_audit", year = 2024)

# Valencia, Spain.
valencia <- cities[grep("Valencia", cities$URAU_NAME, fixed = TRUE), ]
valencia
if (
  requireNamespace("dplyr", quietly = TRUE) &&
    requireNamespace("ggplot2", quietly = TRUE)
) {
  library(dplyr)
  # Get `sf` objects and order by `AREA_SQM`.
  valencia_sf <- gisco_get_unit_urban_audit(
    unit = valencia$URAU_CODE,
    year = 2024
  ) |>
    arrange(desc(AREA_SQM))
  # Plot.
  library(ggplot2)

  ggplot(valencia_sf) +
    geom_sf(aes(fill = URAU_CATG)) +
    scale_fill_viridis_d() +
    labs(
      title = "Valencia",
      subtitle = "Urban Audit 2020",
      fill = "Category"
    )
}
```

---

gisco_get_urban_audit *Urban Audit dataset*

Description

This dataset contains the boundaries of cities ("CITIES"), greater cities ("GREATER_CITIES") and functional urban areas ("FUA") defined according to the EC-OECD city definition. It is used for the Eurostat Urban Audit data collection.

Downloads data from the aggregated GISCO Urban Audit file. To download single-unit Urban Audit files, use gisco_get_unit_urban_audit().

Usage

```
gisco_get_urban_audit(
  year = 2024,
  epsg = 4326,
  cache = TRUE,
```

```
  update_cache = FALSE,
  cache_dir = NULL,
  verbose = FALSE,
  spatialtype = c("RG", "LB"),
  country = NULL,
  level = c("all", "CITIES", "FUA", "GREATER_CITIES", "CITY", "KERN", "LUZ"),
  ext = "gpkg"
)
```

Arguments

year	A character string or numeric value with the release year of the file. One of "2024", "2021", "2020", "2018", "2014", "2004", "2001".
epsg	A character string or numeric value with the coordinate reference system as a 4-digit EPSG code. One of: • "4326": WGS84. • "3035": ETRS89 / ETRS-LAEA. • "3857": Pseudo-Mercator.
cache	A logical value indicating whether to cache results. Defaults to TRUE. See the **Caching strategies** section in gisco_set_cache_dir().
update_cache	A logical value indicating whether to refresh the cached file. Defaults to FALSE. When set to TRUE, it forces a new download.
cache_dir	A character string with a path to a cache directory. See the **Caching strategies** section in gisco_set_cache_dir().
verbose	A logical value indicating whether to display informational messages.
spatialtype	A character string with the type of geometry to return. Available values are: • "RG": Regions - MULTIPOLYGON/POLYGON object. • "LB": Labels - POINT object.
country	A character vector of country codes. It can be either a vector of country names, a vector of ISO 3166-1 alpha-3 country codes or a vector of Eurostat country codes. See also countrycode::countrycode().
level	A character string with the Urban Audit level. Possible values are "all" (the default), which downloads the full dataset, "CITIES", "FUA" and, for versions prior to year = 2020, "GREATER_CITIES", "CITY", "KERN" or "LUZ".
ext	A character string with the extension of the file (default "gpkg"). One of "shp", "gpkg", "geojson".

Details

For more information, see: Eurostat - Statistics Explained.

Cities are defined at several conceptual levels:

- The core city ("CITIES"), using an administrative definition.

- The Functional Urban Area/Large Urban Zone ("FUA"), approximating the functional urban region. Coverage includes the EU, Iceland, Norway and Switzerland. The dataset includes polygon features, point features and a related attribute table which can be joined on the URAU code field.

The "URAU_CATG" field defines the Urban Audit category:

- "C" = City.
- "F" = Functional urban area service type.

Value

An sf object.

Copyright

See the GISCO statistical unit copyright provisions: https://ec.europa.eu/eurostat/web/gisco/geodata/statistical-units.

Note

Check the download and usage provisions in gisco_attributions().

Source

GISCO Urban Audit distribution API: https://gisco-services.ec.europa.eu/distribution/v2/urau/.

See Also

See gisco_bulk_download() to perform a bulk download of datasets.

See gisco_get_unit_urban_audit() to download single-unit files.

Statistical unit datasets: gisco_get_census(), gisco_get_coastal_lines(), gisco_get_lau(), gisco_get_nuts()

Examples

```
cities <- gisco_get_urban_audit(year = 2024, level = "CITIES")

if (!is.null(cities) && requireNamespace("ggplot2", quietly = TRUE)) {
  bcn <- cities[cities$URAU_NAME == "Barcelona", ]

  library(ggplot2)
  ggplot(bcn) +
    geom_sf()
}
```

gisco_id_api *GISCO ID service API*

Description

Functions to interact with the GISCO ID service API, which returns attributes and, optionally, geometry for different datasets at specified coordinates in the selected coordinate reference system.

Each endpoint has a corresponding function in **giscoR**. See **Details** for the lookup modes supported by these functions.

Usage

```
gisco_id_api_geonames(
  x = NULL,
  y = NULL,
  xmin = NULL,
  ymin = NULL,
  xmax = NULL,
  ymax = NULL,
  verbose = FALSE
)

gisco_id_api_nuts(
  x = NULL,
  y = NULL,
  year = 2024,
  epsg = c(4326, 4258, 3035),
  verbose = FALSE,
  nuts_id = NULL,
  nuts_level = NULL,
  geometry = TRUE
)

gisco_id_api_lau(
  x,
  y,
  year = 2024,
  epsg = c(4326, 4258, 3035),
  verbose = FALSE,
  geometry = TRUE
)

gisco_id_api_country(
  x,
  y,
  year = 2024,
```

```
  epsg = c(4326, 4258, 3035),
  verbose = FALSE,
  geometry = TRUE
)

gisco_id_api_river_basin(
  x,
  y,
  year = 2019,
  epsg = c(4326, 4258, 3035),
  verbose = FALSE,
  geometry = TRUE
)

gisco_id_api_biogeo_region(
  x,
  y,
  year = 2016,
  epsg = c(4326, 4258, 3035),
  verbose = FALSE,
  geometry = TRUE
)

gisco_id_api_census_grid(
  x,
  y,
  year = 2021,
  epsg = c(4326, 4258, 3035),
  verbose = FALSE,
  geometry = TRUE
)
```

Arguments

x, y	A character string or numeric value with the coordinates to identify, expressed in the CRS selected by epsg. For Geonames, use longitude and latitude in EPSG:4326.
xmin, ymin, xmax, ymax	A character string or numeric value with bounding box coordinates in EPSG:4326 to identify geonames within the box. The extent must not exceed five degrees in either dimension.
verbose	A logical value indicating whether to display informational messages.
year	A character string or numeric value with the dataset year, see GISCO ID service API for details.
epsg	A character string or numeric value with the EPSG code for the coordinate reference system of the input coordinates and returned geometry. One of 4326, 4258 or 3035. This argument is sent as the API's proj parameter.
nuts_id	A character string with the NUTS ID code.

`nuts_level` A character string or numeric value with the NUTS level. One of `0`, `1`, `2` or `3`. This argument is sent as the API's `level` parameter. If `NULL`, the service uses level `3` for coordinate lookups.

`geometry` A logical value indicating whether to return geometry. If `TRUE`, an `sf` object is returned. If `FALSE`, a tibble is returned.

Details

The service also supports identifier lookups for other datasets and Geonames name searches. These lookup modes are not exposed by the corresponding functions in **giscoR**. Only the NUTS function supports lookup by ID through `nuts_id`.

The available endpoints are:

- `gisco_id_api_geonames()`: Get geographic placenames from longitude and latitude coordinates or a bounding box.
- `gisco_id_api_nuts()`: Return NUTS regions from coordinates or an ID.
- `gisco_id_api_lau()`: Return the ID and, optionally, geometry for Local Administrative Units (LAU) at specified coordinates.
- `gisco_id_api_country()`: Return the ID and, optionally, geometry for countries at specified coordinates.
- `gisco_id_api_river_basin()`: Return the ID and, optionally, geometry for river basins at specified coordinates, based on the Water Framework Directive (WFD) reference spatial datasets.
- `gisco_id_api_biogeo_region()`: Return the ID and, optionally, geometry for biogeographical regions at specified coordinates. The biogeographical regions dataset contains the official delineations used in the Habitats Directive (92/43/EEC) and for the EMERALD Network.
- `gisco_id_api_census_grid()`: Return the ID and, optionally, geometry for census grid cells at specified coordinates.

Value

A tibble when `geometry = FALSE`, or an `sf` object when `geometry = TRUE`. Geonames always returns an `sf` object. Failed requests return `NULL`.

Source

https://gisco-services.ec.europa.eu/id/api-docs/.

See Also

`gisco_get_nuts()`, `gisco_get_lau()`, `gisco_get_countries()`, `gisco_get_census()` and gisco_address_api.

GISCO API tools: `gisco_address_api`

Examples

```
gisco_id_api_geonames(x = -2.5, y = 43.06)

lau <- gisco_id_api_lau(x = -2.5, y = 43.06)
nuts3 <- gisco_id_api_nuts(x = -2.5, y = 43.06, nuts_level = 3)

if (
  all(!is.null(lau), !is.null(nuts3)) &&
    requireNamespace("ggplot2", quietly = TRUE)
) {
  library(ggplot2)

  ggplot(nuts3) +
    geom_sf(fill = "lightblue", color = "black") +
    geom_sf(data = lau, fill = "orange", color = "red") +
    labs(
      title = "NUTS 3 and LAU boundaries",
      subtitle = "Arrasate, Basque Country, Spain",
      caption = "Source: GISCO ID service API"
    )
}
```

---

gisco_nuts_2024 *NUTS 2024 sf object*

Description

This dataset represents the regions for levels 0, 1, 2 and 3 of the Nomenclature of Territorial Units for Statistics (NUTS) for 2024.

Format

An sf object with MULTIPOLYGON geometries, resolution: 1:20 million and EPSG:4326. It has 1798 rows and 19 variables:

NUTS_ID NUTS identifier.

LEVL_CODE NUTS level code (0,1,2,3).

CNTR_CODE Eurostat country code.

NAME_LATN NUTS name in Latin characters.

NUTS_NAME NUTS name in the local alphabet.

MOUNT_TYPE Mountain type, see **Details**.

URBN_TYPE Urban type, see **Details**.

COAST_TYPE Coastal type, see **Details**.

NAME_ENGL Country name in English.

NAME_FREN Country name in French.

ISO3_CODE ISO 3166-1 alpha-3 code of each country from GISCO.

SVRG_UN Sovereign status according to the United Nations.

CAPT Capital city.

EU_STAT European Union member.

EFTA_STAT EFTA member.

CC_STAT EU candidate member.

NAME_GERM Country name in German.

geo Same value as NUTS_ID, provided for compatibility with **eurostat**.

geometry Geometry field.

Details

MOUNT_TYPE: Mountain typology:

- 1: More than 50% of the surface is covered by topographic mountain areas.
- 2: More than 50% of the regional population lives in topographic mountain areas.
- 3: More than 50% of the surface is covered by topographic mountain areas and more than 50% of the regional population lives in these mountain areas.
- 4: Non-mountain region / other regions.
- 0: No classification provided.

URBN_TYPE: Urban-rural typology:

- 1: Predominantly urban region.
- 2: Intermediate region.
- 3: Predominantly rural region.
- 0: No classification provided.

COAST_TYPE: Coastal typology:

- 1: Coastal (on coast).
- 2: Coastal (less than 50% of the population lives within 50 km of the coastline).
- 3: Non-coastal region.
- 0: No classification provided.

Copyright

See the GISCO statistical unit copyright provisions: https://ec.europa.eu/eurostat/web/gisco/geodata/statistical-units.

Note

Check the download and usage provisions in gisco_attributions().

Source

NUTS_RG_20M_2024_4326.gpkg file.

See Also

gisco_get_nuts()

Included datasets: gisco_coastal_lines, gisco_countries_2024, gisco_countrycode, gisco_db

Examples

```
data("gisco_nuts_2024")
head(gisco_nuts_2024)
```

---

gisco_set_cache_dir *Set your* R*hrefhttps://CRAN.R-project.org/package=giscoR***giscoR** *cache directory*

Description

Stores your cache_dir path on your local machine and loads it in future sessions. Use Sys.getenv("GISCO_CACHE_DIR") or gisco_detect_cache_dir() to find your cached path.

Usage

```
gisco_set_cache_dir(
  cache_dir = NULL,
  overwrite = FALSE,
  install = FALSE,
  verbose = TRUE
)

gisco_detect_cache_dir()
```

Arguments

cache_dir	A path to a cache directory. If NULL, the function stores cached files in a temporary directory (see base::tempdir()).
overwrite	If TRUE, overwrite an existing GISCO_CACHE_DIR that you already have on your local machine.
install	If TRUE, install the cache path on your local machine for use in future sessions. Defaults to FALSE. If cache_dir is FALSE, install is automatically set to FALSE.
verbose	A logical value indicating whether to display informational messages.

Details

By default, when no cache_dir is set, the package uses a folder inside base::tempdir() (so files are temporary and are removed when the R session ends). To persist a cache across R sessions, use gisco_set_cache_dir(cache_dir, install = TRUE), which writes the chosen path to a small configuration file under tools::R_user_dir("giscoR", "config").

Value

gisco_set_cache_dir() invisibly returns a character string with the path to your cache_dir, but it is mainly called for its side effect.

gisco_detect_cache_dir() returns the path to the cache_dir used in this session.

Caching strategies

Some files can be read from their online source without caching using the option cache = FALSE. Otherwise the source file is downloaded to your computer. **giscoR** implements the following caching options:

- For occasional use, rely on the default tempdir()-based cache (no install).
- Modify the cache for a single session by setting gisco_set_cache_dir(cache_dir = "a/path/here").
- For reproducible workflows, install a persistent cache with gisco_set_cache_dir(cache_dir = "a/path/here", install = TRUE), which keeps the path across R sessions.
- For caching specific files, use the cache_dir argument in the corresponding function. See the example in gisco_get_nuts().

Sometimes cached files may be corrupt. In that case, try downloading the data by setting update_cache = TRUE in the corresponding function.

If you experience a download problem, try to download the corresponding file by another method and save it in your cache_dir. Use verbose = TRUE to debug the API query and gisco_detect_cache_dir() to identify your cached path.

Note

In **giscoR** >= 1.0.0 the location of the configuration file has moved from rappdirs::user_config_dir("giscoR", "R") to tools::R_user_dir("giscoR", "config"). We have implemented a function that migrates previous configuration files from one location to another with a message. The message appears only once to inform you of the migration.

See Also

tools::R_user_dir()

Cache management utilities: gisco_clear_cache()

Examples

```
# Do not run this. It modifies your current state.
## Not run:
my_cache <- gisco_detect_cache_dir()

# Set an example cache.
ex <- file.path(tempdir(), "example", "cachenew")
gisco_set_cache_dir(ex)

gisco_detect_cache_dir()

# Restore the initial cache.
gisco_set_cache_dir(my_cache)
identical(my_cache, gisco_detect_cache_dir())

## End(Not run)


gisco_detect_cache_dir()
```

Index

∗ **admin**
  gisco_get_communes, 22
  gisco_get_countries, 24
  gisco_get_postal_codes, 42
∗ **api**
  gisco_address_api, 3
  gisco_id_api, 50
∗ **attribution**
  gisco_attributions, 6
∗ **cache_utils**
  gisco_clear_cache, 11
  gisco_set_cache_dir, 55
∗ **database**
  gisco_db, 15
  gisco_get_cached_db, 17
  gisco_get_metadata, 36
∗ **datasets**
  gisco_coastal_lines, 12
  gisco_countries_2024, 13
  gisco_countrycode, 14
  gisco_db, 15
  gisco_nuts_2024, 53
∗ **downloads**
  gisco_bulk_download, 8
  gisco_get_unit, 44
∗ **grids**
  gisco_get_grid, 29
∗ **services**
  gisco_get_education, 26
  gisco_get_healthcare, 31
∗ **stats**
  gisco_get_census, 18
  gisco_get_coastal_lines, 20
  gisco_get_lau, 34
  gisco_get_nuts, 37
  gisco_get_urban_audit, 47
∗ **transport**
  gisco_get_airports, 16
  gisco_get_ports, 40

base::tempdir(), *55, 56*

cache_dir, *8*, *44*
countrycode::codelist, *7*, *14, 15*
countrycode::countrycode(), *16*, *23*, *25–27*, *32*, *35*, *39*, *41*, *43*, *48*

eurostat::get_eurostat_geospatial(), *39*

gisco_address_api, 3, *52*
gisco_address_api_bbox *(*gisco_address_api*)*, 3
gisco_address_api_cities *(*gisco_address_api*)*, 3
gisco_address_api_copyright *(*gisco_address_api*)*, 3
gisco_address_api_countries *(*gisco_address_api*)*, 3
gisco_address_api_housenumbers *(*gisco_address_api*)*, 3
gisco_address_api_most_populated_cell *(*gisco_address_api*)*, 3
gisco_address_api_postcodes *(*gisco_address_api*)*, 3
gisco_address_api_provinces *(*gisco_address_api*)*, 3
gisco_address_api_reverse *(*gisco_address_api*)*, 3
gisco_address_api_roads *(*gisco_address_api*)*, 3
gisco_address_api_search *(*gisco_address_api*)*, 3
gisco_addressapi *(*gisco_address_api*)*, 3
gisco_addressapi_bbox *(*gisco_address_api*)*, 3
gisco_addressapi_cities *(*gisco_address_api*)*, 3
gisco_addressapi_copyright *(*gisco_address_api*)*, 3

gisco_addressapi_countries
(gisco_address_api), 3
gisco_addressapi_housenumbers
(gisco_address_api), 3
gisco_addressapi_postcodes
(gisco_address_api), 3
gisco_addressapi_provinces
(gisco_address_api), 3
gisco_addressapi_reverse
(gisco_address_api), 3
gisco_addressapi_roads
(gisco_address_api), 3
gisco_addressapi_search
(gisco_address_api), 3
gisco_attributions, 6
gisco_attributions(), *12, 13*, *19*, *21*, *23*, *26*, *35*, *39*, *43*, *46*, *49*, *54*
gisco_bulk_download, 8
gisco_bulk_download(), *21*, *23*, *26*, *36, 37*, *39*, *43*, *46*, *49*
gisco_clear_cache, 11
gisco_clear_cache(), *56*
gisco_coastal_lines, 12, *13*, *15*, *21*, *55*
gisco_countries_2024, *12*, 13, *15*, *26*, *55*
gisco_countrycode, *12, 13*, 14, *15*, *25, 26*, *55*
gisco_db, *12, 13*, 15, *15*, *18*, *37*, *55*
gisco_detect_cache_dir
(gisco_set_cache_dir), 55
gisco_detect_cache_dir(), *55, 56*
gisco_get (gisco_get_countries), 24
gisco_get_airports, 16
gisco_get_airports(), *41*
gisco_get_cached_db, 17
gisco_get_cached_db(), *15*, *37*
gisco_get_census, 18
gisco_get_census(), *7*, *21*, *36*, *39*, *49*, *52*
gisco_get_coastal_lines, 20
gisco_get_coastal_lines(), *7*, *10*, *12*, *19*, *36*, *39*, *49*
gisco_get_coastallines
(gisco_get_coastal_lines), 20
gisco_get_communes, 22
gisco_get_communes(), *7*, *10*, *26*, *34*, *36*, *43*
gisco_get_countries, 24
gisco_get_countries(), *7*, *9*, *13*, *15*, *23*, *43, 44*, *46*, *52*
gisco_get_education, 26
gisco_get_education(), *33*
gisco_get_grid, 29
gisco_get_grid(), *29*
gisco_get_healthcare, 31
gisco_get_healthcare(), *28*
gisco_get_lau, 34
gisco_get_lau(), *7*, *10*, *19*, *21–23*, *39*, *49*, *52*
gisco_get_metadata, 36
gisco_get_metadata(), *10*, *15*, *18*, *26*, *46*
gisco_get_nuts, 37
gisco_get_nuts(), *7*, *10*, *19*, *21*, *36*, *44*, *46*, *49*, *52*, *55, 56*
gisco_get_ports, 40
gisco_get_ports(), *17*, *41*
gisco_get_postal_codes, 42
gisco_get_postal_codes(), *7*, *10*, *23*, *26*
gisco_get_postalcodes
(gisco_get_postal_codes), 42
gisco_get_unit, *10*, *37*, 44
gisco_get_unit_country
(gisco_get_unit), 44
gisco_get_unit_country(), *24*, *26*
gisco_get_unit_nuts (gisco_get_unit), 44
gisco_get_unit_nuts(), *37*, *39*
gisco_get_unit_urban_audit
(gisco_get_unit), 44
gisco_get_unit_urban_audit(), *47*, *49*
gisco_get_urban_audit, 47
gisco_get_urban_audit(), *7*, *10*, *19*, *21*, *36*, *39*, *44*, *46*
gisco_id_api, *6*, *46*, 50
gisco_id_api_biogeo_region
(gisco_id_api), 50
gisco_id_api_census_grid
(gisco_id_api), 50
gisco_id_api_census_grid(), *19*
gisco_id_api_country (gisco_id_api), 50
gisco_id_api_country(), *26*
gisco_id_api_geonames (gisco_id_api), 50
gisco_id_api_lau (gisco_id_api), 50
gisco_id_api_lau(), *36*
gisco_id_api_nuts (gisco_id_api), 50
gisco_id_api_nuts(), *39*
gisco_id_api_river_basin
(gisco_id_api), 50
gisco_nuts_2024, *12, 13*, *15*, *39*, 53
gisco_set_cache_dir, 55
gisco_set_cache_dir(), *9*, *11*, *16*, *18–20*, *22*, *25*, *27*, *29*, *32*, *35*, *38*, *41*, *43*, *45*,

48

sf, *5*, *12, 13*, *16*, *19*, *21*, *23*, *25*, *28*, *30*, *33*, *35*, *39*, *41*, *43*, *46*, *49*, *52, 53*

tempdir(), *56*
tibble, *5, 6*, *14, 15*, *18*, *37*, *52*
tools::R_user_dir(), *11*, *56*

warning, *9*