# Package 'giscoR'

October 5, 2020

**Type** Package

**Title** Download Geospatial Data from GISCO API - Eurostat

**Version** 0.1.0-9007

**Description** Tools to download data from the GISCO
(Geographic Information System of the COmmission) Eurostat database
<https://ec.europa.eu/eurostat/web/gisco>.
This package is in no way officially related to or endorsed by Eurostat.

**License** GPL-3

**Encoding** UTF-8

**LazyData** true

**RoxygenNote** 7.1.1

**BugReports** https://github.com/dieghernan/giscoR/issues

**URL** https://dieghernan.github.io/giscoR/, https://github.com/dieghernan/giscoR

**Depends** R (>= 3.3.0)

**Imports** sf (>= 0.9),
countrycode (>= 1.2.0),
geojsonsf (>= 2.0)

**Suggests** cartography (>= 2.0.0),
colorspace,
eurostat,
lwgeom,
tinytest,
knitr,
rmarkdown

**Date** 2020-09-21

**VignetteBuilder** knitr

## R topics documented:

---

`giscoR-package` *Download geospatial data from GISCO API - Eurostat*

---

## Description

giscoR is a API package that helps to retrieve data from Eurostat - GISCO (the Geographic Information System of the COmmission)

## Details

giscoR package

| | |
|---|---|
| Package: | giscoR |
| Type: | Package |
| Version: | See sessionInfo() or DESCRIPTION file |
| Date: | 2020 |
| License: | GPL-3 |
| LazyLoad: | yes |

## Note

COPYRIGHT NOTICE

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

- EN: (C) EuroGeographics for the administrative boundaries
- FR: (C) EuroGeographics pour les limites administratives
- DE: (C) 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 regarding their licence agreements.

## Author(s)

dieghernan, https://github.com/dieghernan/

## Source

GISCO webpage

## References

See `citation("giscoR")`

## See Also

Useful links:

- https://dieghernan.github.io/giscoR/
- https://github.com/dieghernan/giscoR
- Report bugs at https://github.com/dieghernan/giscoR/issues

---

# gisco_attributions *Attribution when publishing GISCO data*

---

## Description

Get the legal text to be used along with the data downloaded with this package

## Usage

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

## Arguments

| | |
|---|---|
| `lang` | Language (two-letter ISO_639-1 code). See details. |
| `copyright` | Boolean. Wheter to display the copyright notice or not on the console. |

## Details

Current languages supported are "en" (English), "da" (Danish), "de" (German), "es" (Spanish), "fi" (Finish), "fr" (French), "no" (Norwegian) and "sv" (Swedish).

Consider contributing if you spot any mistake or want to add a new language.

## Value

A string with the attribution to be used.

## Note

COPYRIGHT NOTICE

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

- EN: (C) EuroGeographics for the administrative boundaries
- FR: (C) EuroGeographics pour les limites administratives
- DE: (C) 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 regarding their licence agreements.

## Examples

```
en <- gisco_attributions()
gisco_attributions(lang = "es", copyright = FALSE )
gisco_attributions(lang = "XXX")
```

---

`gisco_coastallines` *World coastal lines* `LINESTRING` *object*

---

## Description

A `sf` object including the coast lines as provided by GISCO (2016 version).

## Format

A `LINESTRING` data frame (resolution: 1:20million, EPSG:4326) object with 8 variables:

**EFTA_FLAG** Coast belonging to EFTA countries

**OTHR_FLAG** Coast belonging to other countries

**EU_FLAG** Coast belonging to EU countries

**COAS_FLAG** Coast flag

**CNTR_BN_ID** CNTR_BN_ID

**CC_FLAG** Coast belonging to EU candidate countries

**FID** FID

**geometry** geometry field

## Source

GISCO .geojson source

## See Also

gisco_get_countries

## Examples

```
library(sf)

coasts <- gisco_coastallines

plot(
  st_geometry(coasts),
  xlim = c(100, 120),
  ylim = c(-24, 24),
  col = "deepskyblue4",
  lwd = 2
)
box()
title(
  main = "Coasts on Southeastern Asia",
  sub = gisco_attributions(copyright = FALSE),
  cex.sub = 0.7,
  line = 1
)
```

---

# gisco_countries *World countries* POLYGON *object*

---

## Description

A sf object including all countries as provided by GISCO (2016 version).

## Format

A MULTIPOLYGON data frame (resolution: 1:20million, EPSG:4326) object with 257 rows and 7 variables:

**id** row ID

**CNTR_NAME** Official country name on local language

**ISO3_CODE** ISO 3166-1 alpha-3 code of each country, as provided by GISCO

**CNTR_ID** Country ID

**NAME_ENGL** Country name in English

**FID** FID

**geometry** geometry field

## Source

GISCO .geojson source

## See Also

gisco_get_countries

## Examples

```
library(sf)

cntry <- gisco_countries
GBR <- subset(cntry, ISO3_CODE == "GBR")

plot(st_geometry(GBR), col = "red3", border = "blue4")
title(sub = gisco_attributions(), line = 1)
```

---

# gisco_countrycode *Dataframe including Eurostat and ISO2 and ISO3 codes for countries and world regions*

---

## Description

A dataframe containing conversions between different country codification systems (Eurostat/ISO2 and 3) as well as geographic regions as provided by the World Bank and the UN (M49).

## Format

A data frame object with 249 rows and 12 variables:

**CNTR_CODE** Eurostat code of each country

**iso2c** ISO 3166-1 alpha-2 code of each country

**ISO3_CODE** ISO 3166-1 alpha-3 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** As provided by the World Bank

**un.region.code** Numeric region code UN (M49)

**un.region.name** Region name UN (M49)

**un.regionintermediate.code** Numeric intermediate Region code UN (M49)

**un.regionintermediate.name** Intermediate Region name UN (M49)

**un.regionsub.code** Numeric sub-region code UN (M49)

**un.regionsub.name** Sub-Region name UN (M49)

## Source

`codelist` dataset from the `countrycode` `v1.2.0` package.

## See Also

codelist

## Examples

```
# Head
head(gisco_countrycode)
```

---

`gisco_get_coastallines` *Download Coastal Lines from GISCO*

---

## Description

Downloads a simple feature (`sf`) object.

## Usage

```
gisco_get_coastallines(
  resolution = "60",
  year = "2016",
  epsg = "4326",
  cache = TRUE,
  update_cache = FALSE,
  cache_dir = NULL
)
```

## Arguments

| | |
|---|---|
| `resolution` | Resolution of the geospatial data. One of<br>• "60" (1:60million),<br>• "20" (1:20million)<br>• "10" (1:10million)<br>• "03" (1:3million) or<br>• "01" (1:1million). |
| `year` | Release year. One of "2006", "2010", "2013" or "2016" |
| `epsg` | projection of the map: 4-digit EPSG code. One of:<br>• "4326" - WGS84<br>• "3035" - ETRS89 / ETRS-LAEA<br>• "3857" - Pseudo-Mercator |
| `cache` | a logical whether to do caching. Default is `TRUE`. |
| `update_cache` | a logical whether to update cache. |
| `cache_dir` | a path to a cache directory. The directory have to exist. The `NULL` (default) uses and creates `/gisco` directory in the temporary directory from `tempdir`. The directory can also be set with `options(gisco_cache_dir = <path>`. |

## Value

a POLYGON object on sf format.

## Note

Please check the download and usage provisions on gisco_attributions.

## Author(s)

dieghernan, https://github.com/dieghernan/

## Source

GISCO Coastal Lines

## Examples

```
library(sf)

coastlines <- gisco_get_coastallines()
plot(st_geometry(coastlines), col = "palegreen", border = "lightblue3")
title(main = "Coastal Lines",
      sub = gisco_attributions(copyright = FALSE),
      line = 1)
```

---

# gisco_get_communes *Download Geospatial Communes Data from GISCO*

---

## Description

Downloads a simple feature (sf) object.

## Usage

```
gisco_get_communes(
  year = "2016",
  epsg = "4326",
  update_cache = FALSE,
  cache_dir = NULL,
  spatialtype = "RG",
  country = NULL
)
```

## Arguments

| | |
|---|---|
| year | Release year. One of "2001", "2004", "2006", "2008", "2010", "2013" or 2016 |
| epsg | projection of the map: 4-digit EPSG code. One of:<br>• "4326" - WGS84<br>• "3035" - ETRS89 / ETRS-LAEA<br>• "3857" - Pseudo-Mercator |
| update_cache | a logical whether to update cache. |
| cache_dir | a path to a cache directory. The directory have to exist. The NULL (default) uses and creates /gisco directory in the temporary directory from tempdir. The directory can also be set with options(gisco_cache_dir = <path>). |
| spatialtype | Type of geometry to be returned:<br>• RG: Regions - Multipolygon<br>• LB: Labels - Point<br>• BN: Boundaries - Multilines<br>• COASTL: coastlines - Multilines<br>• INLAND: inland boundaries - Multilines |
| country | Optional. A character vector of country codes. See Details. |

## Details

country only available on specific datasets. Some spatialtype datasets (as Multilines data-types) may not have country-level identifies.

country could be either a vector of country names, a vector of ISO3 country codes or a vector of Eurostat country codes.

If you experience any problem on download, try to download the file by any other method and set cache_dir = <folder>.

## Value

a sf object.

## Note

Please check the download and usage provisions on gisco_attributions.

## Author(s)

dieghernan, https://github.com/dieghernan/

## Source

GISCO Communes

## See Also

gisco_get_lau

## Examples

```
library(sf)

communes <- gisco_get_communes(country = c("BEL", "NLD", "LUX"))

plot(
  communes[, "CNTR_ID"],
  pal = c("black", "deepskyblue2", "orange"),
  border = "grey90",
  main = "Communes on Benelux (2016)",
  key.pos = NULL
)
title(sub = gisco_attributions(copyright = FALSE),
      line = 1.2,
      cex.sub = 0.8)
```

---

# gisco_get_countries *Download Geospatial Country Data from GISCO*

---

## Description

Downloads a simple feature (sf) object.

## Usage

```
gisco_get_countries(
  resolution = "60",
  year = "2016",
  epsg = "4326",
  cache = TRUE,
  update_cache = FALSE,
  cache_dir = NULL,
  spatialtype = "RG",
  country = NULL,
  region = NULL
)
```

## Arguments

| | |
|---|---|
| resolution | Resolution of the geospatial data. One of<br>• "60" (1:60million),<br>• "20" (1:20million)<br>• "10" (1:10million)<br>• "03" (1:3million) or<br>• "01" (1:1million). |

| | |
|---|---|
| year | Release year. One of "2001", "2006", "2010", "2013", "2016" or "2020" |
| epsg | projection of the map: 4-digit EPSG code. One of:<br>• "4326" - WGS84<br>• "3035" - ETRS89 / ETRS-LAEA<br>• "3857" - Pseudo-Mercator |
| cache | a logical whether to do caching. Default is TRUE. |
| update_cache | a logical whether to update cache. |
| cache_dir | a path to a cache directory. The directory have to exist. The NULL (default) uses and creates /gisco directory in the temporary directory from tempdir. The directory can also be set with options(gisco_cache_dir = <path>). |
| spatialtype | Type of geometry to be returned:<br>• RG: Regions - Multipolygon<br>• LB: Labels - Point<br>• BN: Boundaries - Multilines<br>• COASTL: coastlines - Multilines<br>• INLAND: inland boundaries - Multilines |
| country | Optional. A character vector of countries. See Details |
| region | Optional. A character vector of UN M49 region codes. Possible values are "Africa", "Americas", "Asia", "Europe", "Oceania". See Details and gisco_countrycode |

## Details

country and region only available when applicable.

country could be either a vector of country names, a vector of ISO3 country codes or a vector of Eurostat country codes.

## Value

a sf object.

## Note

Please check the download and usage provisions on gisco_attributions.

## Author(s)

dieghernan, https://github.com/dieghernan/

## Source

GISCO Countries

## See Also

gisco_countrycode

## Examples

```
library(sf)

sf_world <- gisco_get_countries()
plot(st_geometry(sf_world), col = "seagreen2")
title(sub = gisco_attributions(), line = 1)


sf_africa <- gisco_get_countries(region = 'Africa')
plot(st_geometry(sf_africa),
     col = c("springgreen4", "darkgoldenrod1", "red2"))
title(sub = gisco_attributions(), line = 1)


sf_benelux <-
  gisco_get_countries(resolution = "20",
                      country = c('Belgium', 'Netherlands', 'Luxembourg'))
plot(st_geometry(sf_benelux),
     col = c("grey10", "orange", "deepskyblue2"))
title(sub = gisco_attributions(), line = 1)
```

---

# gisco_get_lau *Download Geospatial Local Administrative Units Data from GISCO*

---

## Description

Downloads a simple feature (`sf`) object.

## Usage

```
gisco_get_lau(
  year = "2016",
  epsg = "4326",
  update_cache = FALSE,
  cache_dir = NULL,
  country = NULL,
  gisco_id = NULL
)
```

## Arguments

| | |
|---|---|
| `year` | Release year. One of "2016", "2017", "2018" or "2019" |
| `epsg` | projection of the map: 4-digit EPSG code. One of:<br>• "4326" - WGS84<br>• "3035" - ETRS89 / ETRS-LAEA<br>• "3857" - Pseudo-Mercator |

| | |
|---|---|
| update_cache | a logical whether to update cache. |
| cache_dir | a path to a cache directory. The directory have to exist. The NULL (default) uses and creates /gisco directory in the temporary directory from tempdir. The directory can also be set with options(gisco_cache_dir = <path>). |
| country | Optional. A character vector of country codes. See details |
| gisco_id | Optional. A character vector of GISCO_ID LAU values. |

## Details

See https://ec.europa.eu/eurostat/web/nuts/local-administrative-units for more detail about LAUs.

If you experience any problem on download, try to download the file by any other method and set cache_dir = <folder>.

country could be either a vector of country names, a vector of ISO3 country codes or a vector of Eurostat country codes.

## Value

a POLYGON object on sf format.

## Note

Please check the download and usage provisions on gisco_attributions.

## Author(s)

dieghernan, https://github.com/dieghernan/

## Source

GISCO Local Administrative Units

## See Also

gisco_get_communes

## Examples

```
library(sf)

lau_esp <- gisco_get_lau(country = "España")

plot(
  st_geometry(lau_esp),
  xlim = c(0, 4),
  ylim = c(39, 42),
  col = "wheat",
  border = "grey50"
```

```
)

box()

title(
  main = "Spain LAU",
  sub = gisco_attributions(copyright = FALSE),
  line = 1,
  cex.sub = 0.8,
  font.sub = 3
)
```

---

`gisco_get_nuts` *Download Geospatial NUTS Data from GISCO*

---

## Description

Downloads a simple feature (`sf`) object.

## Usage

```
gisco_get_nuts(
  resolution = "60",
  year = "2016",
  epsg = "4326",
  nuts_level = "all",
  cache = TRUE,
  update_cache = FALSE,
  cache_dir = NULL,
  spatialtype = "RG",
  country = NULL,
  nuts_id = NULL
)
```

## Arguments

| | |
|---|---|
| `resolution` | Resolution of the geospatial data. One of<br>• "60" (1:60million),<br>• "20" (1:20million)<br>• "10" (1:10million)<br>• "03" (1:3million) or<br>• "01" (1:1million). |
| `year` | Release year. One of "2003", "2006", "2010", "2013", "2016" or "2021". |
| `epsg` | projection of the map: 4-digit EPSG code. One of:<br>• "4326" - WGS84 |

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

`nuts_level` NUTS level. One of "0" (Country-level), "1", "2" or "3". See https://ec.europa.eu/eurostat/web/nuts/background.

`cache` a logical whether to do caching. Default is `TRUE`.

`update_cache` a logical whether to update cache.

`cache_dir` a path to a cache directory. The directory have to exist. The `NULL` (default) uses and creates `/gisco` directory in the temporary directory from `tempdir`. The directory can also be set with `options(gisco_cache_dir = <path>)`.

`spatialtype` Type of geometry to be returned:

- RG: Regions - Multipolygon
- LB: Labels - Point
- BN: Boundaries - Multilines

`country` Optional. A character vector of country codes. See Details

`nuts_id` Optional. A character vector of NUTS IDs.

## Details

`country` only available when applicable. Some `spatialtype` datasets (as Multilines data-types) may not have country-level identifies.

`country` could be either a vector of country names, a vector of ISO3 country codes or a vector of Eurostat country codes.

## Value

a `sf` object.

## Note

Please check the download and usage provisions on gisco_attributions.

## Author(s)

dieghernan, https://github.com/dieghernan/

## Source

GISCO NUTS

## See Also

gisco_countrycode, gisco_nuts

## Examples

```
library(sf)

nuts0 <- gisco_get_nuts(nuts_level = "0", epsg = "3035")
nuts3 <- gisco_get_nuts(nuts_level = "3", epsg = "3035")

plot(
  st_geometry(nuts3),
  xlim = c(2200000, 7150000),
  ylim = c(1380000, 5500000),
  lwd = 0.05,
  border = "grey40"
)
plot(st_geometry(nuts0),
     lwd = 2,
     add = TRUE)
title(
  main = "NUTS3 Levels on Europe",
  sub = gisco_attributions(copyright = FALSE),
  cex.sub = 0.7,
  line = 1
)
```

---

# gisco_get_urban_audit *Download Geospatial Urban Audit Data from GISCO*

---

## Description

Downloads a simple feature (sf) object.

## Usage

```
gisco_get_urban_audit(
  year = "2018",
  epsg = "4326",
  update_cache = FALSE,
  cache_dir = NULL,
  spatialtype = "RG",
  level = NULL,
  country = NULL
)
```

## Arguments

| | |
|---|---|
| year | Release year. One of "2014", "2018" or "2020" |
| epsg | projection of the map: 4-digit EPSG code. One of:<br>• "4326" - WGS84<br>• "3035" - ETRS89 / ETRS-LAEA |

- "3857" - Pseudo-Mercator

| | |
|---|---|
| update_cache | a logical whether to update cache. |
| cache_dir | a path to a cache directory. The directory have to exist. The NULL (default) uses and creates /gisco directory in the temporary directory from tempdir. The directory can also be set with options(gisco_cache_dir = <path>). |
| spatialtype | Type of geometry to be returned:<br>• RG: Regions - Multipolygon<br>• LB: Labels - Point |
| level | Level of Urban Audit. Possible values are 'CITIES', 'FUA', 'GREATER_CITIES' or NULL. See Details. |
| country | Optional. A character vector of country codes. See details. |

## Details

level = NULL would download the whole dataset including all levels

country could be either a vector of country names, a vector of ISO3 country codes or a vector of Eurostat country codes.

## Value

a sf object.

## Note

Please check the download and usage provisions on gisco_attributions.

## Author(s)

dieghernan, https://github.com/dieghernan/

## Source

GISCO Urban Audit

## Examples

```
library(sf)

FUA <-
  gisco_get_urban_audit(
    year = "2020",
    epsg = "3035",
    level = "FUA",
    country = "Deutschland"
  )

countries <- gisco_get_countries(
  resolution = "20",
```

```
  year = "2020",
  epsg = "3035",
  country = "Deutschland"
)

plot(st_geometry(countries)  ,
     col = "grey10", )
plot(st_geometry(FUA),
     add = TRUE,
     col = "darkgoldenrod3")
title(
  main = "FUA (Functional Urban Areas) \non Germany (2020)",
  sub = gisco_attributions(copyright = FALSE),
  cex.main = 0.8,
  cex.sub = 0.7,
  line = 1
)
```

---

`gisco_nuts` *All NUTS* `POLYGON` *object*

---

## Description

A `sf` object including all NUTS levels as provided by GISCO (2016 version).

## Format

A `POLYGON` data frame (resolution: 1:20million, EPSG:4326) object with 11 variables:

**id** row ID

**COAST_TYPE** COAST_TYPE

**MOUNT_TYPE** MOUNT_TYPE

**NAME_LATN** Name on Latin characters

**CNTR_CODE** Eurostat Country code

**FID** FID

**NUTS_ID** NUTS identifier

**NUTS_NAME** NUTS name on local alphabet

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

**URBN_TYPE** URBN_TYPE

**geometry** geometry field

## Source

GISCO .geojson source

## See Also

gisco_get_nuts

## Examples

```
library(sf)

nuts <- gisco_nuts

italy <- subset(nuts, CNTR_CODE == "IT" & LEVL_CODE == 3)

plot(st_geometry(italy), col = c("springgreen4", "ivory", "red2"))
title(
  sub = gisco_attributions(copyright = FALSE),
  cex.sub = 0.7,
  font.sub = 3
)
```

# Index