Package ‘mapSpain’

April 12, 2021

Type Package

Title Administrative Boundaries of Spain

Version 0.2.1.9000

Description Administrative Boundaries of Spain at several levels (CCAA,
Provinces, Municipalities) based on the GISCO Eurostat database
<https://ec.europa.eu/eurostat/web/gisco> and 'CartoBase SIANE'
from 'Instituto Geografico Nacional' <https://www.ign.es/>.
It also provides a 'leaflet' plugin and the ability of
downloading and processing static tiles.

License GPL-3

Encoding UTF-8

LazyData true

RoxygenNote 7.1.1

Roxygen list(markdown = TRUE)

BugReports `https://github.com/rOpenSpain/mapSpain/issues`

URL `https://ropenspain.github.io/mapSpain/`, `https://github.com/rOpenSpain/mapSpain`

Depends R (>= 3.6.0)

Imports sf (>= 0.9),
countrycode (>= 1.2.0),
giscoR (>= 0.2.0),
raster (>= 3.0),
png (>= 0.1-5),
slippymath (>= 0.3.1),
leaflet (>= 2.0.0)

Suggests tmap (>= 3.0.0),
cartography (>= 2.4),
rgdal,
tinytest,
tibble

R topics documented:

`mapSpain-package` *mapSpain package*

Description

This package provides Administrative Boundaries of Spain based on the GISCO (Geographic Information System of the Commission) Eurostat database and CartoBase SIANE from Instituto Geográfico Nacional.

Details

Package	mapSpain
Type	Package
Version	See sessionInfo() or DESCRIPTION file
Date	2021
License	GPL-3
LazyLoad	yes

Note

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, 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 license agreements.

Author(s)

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

Source

GISCO webpage

References

See `citation("mapSpain")`.

See Also

Useful links:

- `https://ropenspain.github.io/mapSpain/`
- `https://github.com/rOpenSpain/mapSpain`
- Report bugs at `https://github.com/rOpenSpain/mapSpain/issues`

`addProviderEspTiles` *Leaflet plugin - Spanish providers*

Description

Add tiles of https://dieghernan.github.io/leaflet-providersESP/ to a **R** `leaflet::leaflet()` map.

Usage

```
addProviderEspTiles(
  map,
  provider,
  layerId = NULL,
  group = NULL,
  options = providerEspTileOptions()
)

providerEspTileOptions(...)
```

Arguments

`map`	a map widget object created from `leaflet()`
`provider`	Name of the provider, see `leaflet.providersESP.df`.
`layerId`	the layer id
`group`	the name of the group the newly created layers should belong to (for `clearGroup` and `addLayersControl` purposes). Human-friendly group names are permitted–they need not be short, identifier-style names. Any number of layers and even different types of layers (e.g. markers and polygons) can share the same group name.
`options`	a list of extra options for tile layers, popups, paths (circles, rectangles, polygons, ...), or other map elements
`...`	Arguments passed on to `leaflet::providerTileOptions` `errorTileUrl` the tile layer options; see https://leafletjs.com/reference-1.3.4.html#tilelayer `noWrap` the tile layer options; see https://leafletjs.com/reference-1.3.4.html#tilelayer `opacity` the tile layer options; see https://leafletjs.com/reference-1.3.4.html#tilelayer `zIndex` the tile layer options; see https://leafletjs.com/reference-1.3.4.html#tilelayer `updateWhenIdle` the tile layer options; see https://leafletjs.com/reference-1.3.4.html#tilelayer `detectRetina` the tile layer options; see https://leafletjs.com/reference-1.3.4.html#tilelayer

Details

Wrapper of `leaflet::providerTileOptions()`

Value

Modified `map` object.

Author(s)

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

Source

`https://dieghernan.github.io/leaflet-providersESP/` leaflet plugin, **v1.2.0**.

See Also

`leaflet.providersESP.df`, `esp_getTiles()`

`leaflet::providerTileOptions()`, `leaflet::tileOptions()`

Examples

```
library(leaflet)
PuertadelSol <-
  leaflet() %>%
  setView(
    lat = 40.4166,
    lng = -3.7038400,
    zoom = 18
  ) %>%
  addProviderEspTiles(provider = "IGNBase.Gris") %>%
  addProviderEspTiles(provider = "RedTransporte.Carreteras")

PuertadelSol
```

`esp_codelist` *Spanish Code Translation Data Frame*

Description

A data frame used internally for translating codes and names of the different subdivisions of Spain. The data frame provides the hierarchy of the subdivisions including NUTS1 level, Autonomous Communities (equivalent to NUTS2), Provinces and NUTS3 level. See Note.

Format

data frame with codes as columns

- **nuts*.code**: NUTS code of each subdivision.
- **nuts*.code**: NUTS name of each subdivision.
- **codauto**: INE code of each autonomous community.
- **iso2.*.code**: ISO2 code of each autonomous community and province.
- **ine.*.name**: INE name of each autonomous community and province.
- **iso2.*.name.(lang)**: ISO2 name of each autonomous community and province. Several languages available.
- **cldr.*.name.(lang)**: CLDR name of each autonomous community and province. Several languages available.
- **ccaa.short.***: Short (common) name of each autonomous community. Several languages available.
- **cpro**: INE code of each province.
- **prov.shortname.***: Short (common) name of each province. Several languages available.

Note

Languages available are:

- "en": English
- "es": Spanish
- "ca": Catalan
- "ga": Galician
- "eu": Basque

Although NUTS2 matches the first subdivision level of Spain (CCAA - Autonomous Communities), it should be noted that NUTS3 does not match the second subdivision level of Spain (Provinces). NUTS3 provides a dedicated code for major islands whereas the Provinces doesn't.

Ceuta and Melilla has an specific status (Autonomous Cities) but are considered as communities with a single province (as Madrid, Asturias or Murcia) on this dataset.

Source

- INE: Instituto Nacional de Estadistica: `https://www.ine.es/`
- Eurostat (NUTS): `https://ec.europa.eu/eurostat/web/nuts/background`
- ISO: `https://www.iso.org/obp/ui/#iso:code:3166:ES`
- CLDR: `https://unicode-org.github.io/cldr-staging/charts/38/index.html`

Examples

```
library(tibble)

glimpse(as_tibble(esp_codelist))
```

`esp_dict_region_code` *Convert and translate Subdivision Names*

Description

Converts long subdivision names into different coding schemes and languages.

Usage

```
esp_dict_region_code(sourcevar, origin = "text", destination = "text")

esp_dict_translate(sourcevar, lang = "en", all = FALSE)
```

Arguments

`sourcevar`	Vector which contains the subdivision names to be converted.
`origin, destination`	One of "text", "nuts", "iso2", "codauto" and "cpro".
`lang`	Language of translation. Available languages are: • "es": Spanish • "en": English • "ca": Catalan • "ga": Galician • "eu": Basque
`all`	Logical. Should the function return all names or not? On `FALSE` it returns a character vector. See Value

Details

If no match is found for any value, the function displays a warning and returns `NA` for those values.

Note that mixing names of different administrative levels (e.g. "Catalonia" and "Barcelona") may return empty values, depending on the `destination` values.

Value

`esp_dict_region_code` returns a vector of characters.

`esp_dict_translate` returns a character vector or a named list with each of the possible names of each `sourcevar` on the required language `lang`.

Author(s)

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

Examples

```
vals <- c("Errioxa", "Coruna", "Gerona", "Madrid")

esp_dict_region_code(vals)
esp_dict_region_code(vals, destination = "nuts")
esp_dict_region_code(vals, destination = "cpro")
esp_dict_region_code(vals, destination = "iso2")

# From ISO2 to another codes

iso2vals <- c("ES-M", "ES-S", "ES-SG")
esp_dict_region_code(iso2vals, origin = "iso2")
esp_dict_region_code(iso2vals,
  origin = "iso2",
  destination = "nuts"
)
esp_dict_region_code(iso2vals,
  origin = "iso2",
  destination = "cpro"
)

# Mixing levels
valsmix <- c("Centro", "Andalucia", "Seville", "Menorca")
esp_dict_region_code(valsmix, destination = "nuts")
## Not run:

# Warning

esp_dict_region_code(valsmix, destination = "codauto")
esp_dict_region_code(valsmix, destination = "iso2")

## End(Not run)


vals <- c(
  "La Rioja", "Sevilla", "Madrid",
  "Jaen", "Orense", "Baleares"
)
esp_dict_translate(vals)
esp_dict_translate(vals, lang = "es")
esp_dict_translate(vals, lang = "ca")
esp_dict_translate(vals, lang = "eu")
esp_dict_translate(vals, lang = "ga")

esp_dict_translate(vals, lang = "ga", all = TRUE)
```

`esp_getTiles` *Get Tiles from Public Administrations of Spanish.*

Description

Get static map tiles based on a spatial object. Maps can be fetched from various open map servers.

This function is a implementation of the javascript plugin leaflet-providersESP **v1.2.0**.

Usage

```
esp_getTiles(
  x,
  type = "IDErioja",
  zoom = NULL,
  crop = TRUE,
  res = 512,
  bbox_expand = 0.05,
  transparent = TRUE,
  mask = FALSE,
  update_cache = FALSE,
  cache_dir = NULL,
  verbose = FALSE
)
```

Arguments

`x`	An `sf` object.
`type`	Name of the provider. See `leaflet.providersESP.df`.
`zoom`	Zoom level. If `NULL`, it is determined automatically. Only valid for WMTS.
`crop`	`TRUE` if results should be cropped to the specified `x` extent, `FALSE` otherwise. If `x` is an `sf` object with one `POINT`, crop is set to `FALSE`.
`res`	Resolution (in pixels) of the final tile. Only valid for WMS.
`bbox_expand`	A numeric value that indicates the expansion percentage of the bounding box of `x`.
`transparent`	Logical. Provides transparent background, if supported. Depends on the selected provider on `type`.
`mask`	`TRUE` if the result should be masked to `x`.
`update_cache`	A logical whether to update cache. Default is `FALSE`. When set to `TRUE` it would force a fresh download of the source `.geojson` file.
`cache_dir`	A path to a cache directory. The directory can also be set globally with: • `options(mapSpain_cache_dir = "path/to/dir")`. See Details on `esp_get_nuts()`.
`verbose`	Display information. Useful for debugging, default is `FALSE`.

Details

Zoom levels are described on the OpenStreetMap wiki: `https://wiki.openstreetmap.org/wiki/Zoom_levels`.

Results of esp_getTiles could be plotted using raster::plotRGB(), tmap::tm_rgb(), etc.

For a complete list of providers see leaflet.providersESP.df.

Most WMS/WMTS providers provide tiles on EPSG:3857. In case that the tile looks deformed, try projecting first x:

x <-sf::st_transform(x,3857)

Tiles are cached under the path cache_dir/[type].

Value

A RasterBrick is returned, with 3 (RGB) or 4 (RGBA) layers, depending on the provider. .

Author(s)

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

Source

https://dieghernan.github.io/leaflet-providersESP/ leaflet plugin, **v1.2.0**.

See Also

leaflet.providersESP.df, addProviderEspTiles(), raster::plotRGB(), tmap::tm_rgb()

Examples

```
library(sf)

Murcia <- esp_get_ccaa("Murcia")
Murcia <- st_transform(Murcia, 3857)

Tile <- esp_getTiles(Murcia)

library(tmap)

tm_shape(Tile, raster.downsample = FALSE) +
  tm_rgb(interpolate = FALSE) +
  tm_shape(Murcia) +
  tm_borders()
```

esp_get_can_box *Get complementary lines when plotting Canary Islands.*

Description

When plotting Spain, it is usual to represent the Canary Islands as an inset (see `moveCAN` on `esp_get_nuts()`. These functions provides complementary borders when Canary Islands are displaced.

`esp_get_can_box` is used to draw lines around the displaced Canary Islands.

`esp_get_can_provinces` is used to draw a separator line between the two provinces of the Canary Islands.

Usage

```
esp_get_can_box(style = "right", moveCAN = TRUE, epsg = "4258")

esp_get_can_provinces(moveCAN = TRUE, epsg = "4258")
```

Arguments

`style`	Style of line around Canary Islands. Four options available: "left", "right", "box" or "poly".
`moveCAN`	A logical `TRUE/FALSE` or a vector of coordinates `c(lat,lon)`. It places the Canary Islands close to Spain's mainland. Initial position can be adjusted using the vector of coordinates.
`epsg`	projection of the map: 4-digit EPSG code. One of: • "4258": ETRS89 • "4326": WGS84 • "3035": ETRS89 / ETRS-LAEA • "3857": Pseudo-Mercator

Value

A `LINESTRING` or `POLYGON` object if `style = "poly"`.

`esp_get_can_provinces` returns a `LINESTRING` object.

Author(s)

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

Source

`esp_get_can_provinces` extracted from CartoBase ANE, `se89_mult_admin_provcan_l.shp` file.

See Also

esp_get_nuts(), esp_get_ccaa().

Examples

```
Provs <- esp_get_prov()
Box <- esp_get_can_box()
Line <- esp_get_can_provinces()

# Plot

library(tmap)


tm_shape(Provs) +
  tm_polygons() +
  tm_shape(Box) +
  tm_lines() +
  tm_shape(Line) +
  tm_lines()



# Displacing Canary

Provs_D <- esp_get_prov(moveCAN = c(15, 0))
Box_D <- esp_get_can_box(style = "left", moveCAN = c(15, 0))
Line_D <- esp_get_can_provinces(moveCAN = c(15, 0))

tm_shape(Provs_D) +
  tm_polygons() +
  tm_shape(Box_D) +
  tm_lines() +
  tm_shape(Line_D) +
  tm_lines()


# Example with poly option

# Get countries with giscoR

library(giscoR)

Countries <-
  gisco_get_countries(
    res = "01",
    epsg = "4326",
    country = c("France", "Portugal", "Andorra", "Morocco", "Argelia")
  )
CANbox <-
  esp_get_can_box(
    style = "poly",
    epsg = "4326",
    moveCAN = c(12.5, 0)
  )

CCAA <- esp_get_ccaa(
  res = "01",
  epsg = "4326",
  moveCAN = c(12.5, 0) #' Same displacement factor)
)

# Plot

tm_shape(Countries, bbox = c(-10, 34.6, 4.3, 44)) +
  tm_polygons(col = "#DFDFDF") +
  tm_shape(CANbox) +
  tm_polygons(col = "#C7E7FB") +
  tm_shape(CANbox) +
  tm_borders(lwd = 2) +
  tm_shape(CCAA) +
  tm_polygons("#FDFBEA") +
  tm_graticules(lines = FALSE) +
  tm_layout(bg.color = "#C7E7FB", frame.double.line = TRUE)
```

esp_get_capimun *Get the location of municipalities of Spain*

Description

Get the location of the political powers for each municipality (possibly the center of the municipality).

Note that this differs of the centroid of the boundaries of the municipallity, returned by `esp_get_munic()`.

Usage

```
esp_get_capimun(
  year = Sys.Date(),
  epsg = "4258",
  cache = TRUE,
  update_cache = FALSE,
  cache_dir = NULL,
  verbose = FALSE,
  region = NULL,
  munic = NULL,
  moveCAN = TRUE,
  rawcols = FALSE
)
```

Arguments

`year`	Release year. See Details for years available.
`epsg`	projection of the map: 4-digit EPSG code. One of: • "4258": ETRS89 • "4326": WGS84 • "3035": ETRS89 / ETRS-LAEA • "3857": Pseudo-Mercator
`cache`	A logical whether to do caching. Default is `TRUE`.
`update_cache`	A logical whether to update cache. Default is `FALSE`. When set to `TRUE` it would force a fresh download of the source `.geojson` file.
`cache_dir`	A path to a cache directory. The directory can also be set globally with: • `options(mapSpain_cache_dir = "path/to/dir")`. See Details on `esp_get_nuts()`.
`verbose`	Display information. Useful for debugging, default is `FALSE`.
`region`	Optional. A vector of region names, NUTS or ISO codes (see `esp_dict_region_code()`.
`munic`	A name or regex expression with the names of the required municipalities. `NULL` would not produce any filtering.
`moveCAN`	A logical `TRUE/FALSE` or a vector of coordinates `c(lat,lon)`. It places the Canary Islands close to Spain's mainland. Initial position can be adjusted using the vector of coordinates.
`rawcols`	Logical. Setting this to `TRUE` would add the raw columns of the dataset provided by IGN.

Details

`year` could be passed as a single year ("YYYY" format, as end of year) or as a specific date ("YYYY-MM-DD" format). Historical information starts as of 2005.

When using `region` you can use and mix names and NUTS codes (levels 1, 2 or 3), ISO codes (corresponding to level 2 or 3) or `cpro`.

When calling a superior level (Province, Autonomous Community or NUTS1) , all the municipalities of that level would be added.

Value

A `POINT` object.

Note

While `moveCAN` is useful for visualization, it would alter the actual geographical position of the Canary Islands.

Author(s)

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

Source

IGN data via a custom CDN (see https://github.com/rOpenSpain/mapSpain/tree/sianedata.

See Also

esp_get_munic(), esp_munic.sf, esp_codelist

Examples

```
# This code compares centroid of municipalities against esp_get_capimun
library(sf)

# Get shape
area <- esp_get_munic_siane(munic = "Valladolid", epsg = 3857)

# Area in km2
print(paste0(round(as.double(st_area(area)) / 1000000, 2), " km2"))

# Extract centroid
centroid <- st_centroid(area)

# Compare with capimun
capimun <- esp_get_capimun(munic = "Valladolid", epsg = 3857)


# Get a tile to check
tile <- esp_getTiles(area, zoom = 11)

# Check on plot
library(tmap)

tm_shape(tile, raster.downsample = FALSE) +
  tm_rgb() +
  tm_shape(area) +
  tm_borders(col = "grey40") +
  tm_shape(centroid) +
  tm_symbols(col = "red", alpha = 0.4, shape = 19) +
  tm_shape(capimun) +
  tm_symbols(col = "blue", alpha = 0.4, shape = 19)


# Blue dot is located onto the actual city while red dot is located
# in the centroid of the boundaries
```

---

esp_get_ccaa *Get Autonomous Communities boundaries of Spain*

Description

Loads a simple feature (`sf`) object containing the autonomous communities boundaries of Spain.

`esp_get_ccaa` uses GISCO (Eurostat) as source

`esp_get_ccaa_siane` uses CartoBase ANE as source, provided by Instituto Geografico Nacional (IGN), `http://www.ign.es/web/ign/portal`.

Years available are 2005 up to today.

Usage

```
esp_get_ccaa(ccaa = NULL, ...)

esp_get_ccaa_siane(
  ccaa = NULL,
  year = Sys.Date(),
  epsg = "4258",
  cache = TRUE,
  update_cache = FALSE,
  cache_dir = NULL,
  verbose = FALSE,
  resolution = "3",
  moveCAN = TRUE,
  rawcols = FALSE
)
```

Arguments

`ccaa`	A vector of names and/or codes for autonomous communities or `NULL` to get all the autonomous communities. See Details.
`...`	Arguments passed on to `esp_get_nuts` `spatialtype` Type of geometry to be returned: • "RG": Regions - `MULTIPOLYGON/POLYGON` object. • "LB": Labels - `POINT` object.
`year`	Release year. See `esp_get_nuts()` for `esp_get_ccaa` and Details for `esp_get_ccaa_siane`
`epsg`	projection of the map: 4-digit EPSG code. One of: • "4258": ETRS89 • "4326": WGS84 • "3035": ETRS89 / ETRS-LAEA • "3857": Pseudo-Mercator
`cache`	A logical whether to do caching. Default is `TRUE`.
`update_cache`	A logical whether to update cache. Default is `FALSE`. When set to `TRUE` it would force a fresh download of the source `.geojson` file.
`cache_dir`	A path to a cache directory. The directory can also be set globally with: • `options(mapSpain_cache_dir = "path/to/dir")`. See Details on `esp_get_nuts()`.

verbose	Display information. Useful for debugging, default is `FALSE`.
resolution	Resolution of the polygon. Values available are "3", "6.5" or "10".
moveCAN	A logical `TRUE/FALSE` or a vector of coordinates `c(lat,lon)`. It places the Canary Islands close to Spain's mainland. Initial position can be adjusted using the vector of coordinates.
rawcols	Logical. Setting this to `TRUE` would add the raw columns of the dataset provided by IGN.

Details

When using `ccaa` you can use and mix names and NUTS codes (levels 1 or 2), ISO codes (corresponding to level 2) or `codauto`. Ceuta and Melilla are considered as Autonomous Communities on this dataset.

When calling a NUTS1 level, all the Autonomous Communities of that level would be added.

On `esp_get_ccaa_siane`, `year` could be passed as a single year ("YYYY" format, as end of year) or as a specific date ("YYYY-MM-DD" format). Historical information starts as of 2005.

Value

A `POLYGON/POINT` object.

Author(s)

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

Source

IGN data via a custom CDN (see https://github.com/rOpenSpain/mapSpain/tree/sianedata.

See Also

`esp_get_hex_ccaa()`, `esp_get_nuts()`, `esp_get_prov()`, `esp_get_munic()`, `esp_codelist`

Examples

```
# Random CCAA
Random <- esp_get_ccaa(ccaa = c(
  "Euskadi",
  "Catalunya",
  "ES-EX",
  "Canarias",
  "ES52",
  "01"
))

library(tmap)

tm_shape(Random) +
  tm_polygons(col = "codauto", legend.show = FALSE) +
  tm_shape(Random, point.per = "feature") +
  tm_text("codauto",
    auto.placement = TRUE,
    shadow = TRUE
  )


# All CCAA of a Zone plus an addition

Mix <-
  esp_get_ccaa(
    ccaa = c("La Rioja", "Noroeste"),
    resolution = "20"
  )

# Base plot
plot(
  Mix[, "nuts1.code"],
  pal = hcl.colors(2),
  key.pos = NULL,
  main = NULL,
  border = "white"
)

# Combine with giscoR to get countries

library(giscoR)
library(sf)

res <- 20 # Set same resoluion

europe <- gisco_get_countries(resolution = res)
ccaa <- esp_get_ccaa(moveCAN = FALSE, resolution = res)

# Transform to same CRS
europe <- st_transform(europe, 3035)
ccaa <- st_transform(ccaa, 3035)

tm_shape(europe, bbox = c(23, 14, 74, 55) * 10e4) +
  tm_polygons("#DFDFDF", border.col = "#656565") +
  tm_shape(ccaa) +
  tm_polygons("#FDFBEA", border.col = "#656565") +
  tm_layout(bg.color = "#C7E7FB")
```

`esp_get_country` *Get boundaries of Spain*

Description

Loads a single `sf` object containing the boundaries of Spain.

Usage

```
esp_get_country(...)
```

Arguments

`...`	Additional parameters from `esp_get_nuts()`.

Value

A `MULTIPOLYGON/MULTIPOINT` object.

Author(s)

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

See Also

`esp_get_nuts()`

Examples

```
library(sf)

OriginalCan <- esp_get_country(moveCAN = FALSE)

plot(OriginalCan$geometry, col = "grey70")

MovedCan <- esp_get_country(moveCAN = TRUE)

plot(MovedCan$geometry, col = "grey70")
```

`esp_get_hex_prov` *Get an hexbin or a map of squares of Spain*

Description

Loads a hexbin map (`sf` object) or a map of squares with the boundaries of the provinces or autonomous communities of Spain.

Usage

```
esp_get_hex_prov(prov = NULL)

esp_get_hex_ccaa(ccaa = NULL)

esp_get_grid_prov(prov = NULL)

esp_get_grid_ccaa(ccaa = NULL)
```

Arguments

`prov`	A vector of names and/or codes for provinces or `NULL` to get all the provinces. See Details.
`ccaa`	A vector of names and/or codes for autonomous communities or `NULL` to get all the autonomous communities. See Details.

Details

Hexbin or grid map has an advantage over usual choropleth maps. In choropleths, a large polygon data looks more emphasized just because of its size, what introduces a bias. Here with hexbin, each region is represented equally dismissing the bias.

Results are provided in **EPSG:4258**, use `sf::st_transform()` to change the projection.

See `esp_get_ccaa()`, `esp_get_prov()` for Details.

Value

A `POLYGON` object.

Author(s)

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

See Also

`esp_get_nuts()`, `esp_get_ccaa()`, `esp_get_prov()`, `esp_get_munic()`, `esp_codelist`

Examples

```
library(tmap)

esp <- esp_get_country()
hexccaa <- esp_get_hex_ccaa()


tm_shape(esp, bbox = c(-13.5, 32, 7, 45)) +
  tm_polygons() +
  tm_shape(hexccaa) +
  tm_polygons("codauto", alpha = 0.6, legend.show = FALSE) +
  tm_shape(hexccaa) +
  tm_text("label")



hexprov <- esp_get_hex_prov()

tm_shape(esp, bbox = c(-13.5, 32, 7, 45)) +
  tm_polygons() +
  tm_shape(hexprov) +
  tm_polygons("cpro", alpha = 0.6, legend.show = FALSE) +
  tm_shape(hexprov) +
  tm_text("label")



gridccaa <- esp_get_grid_ccaa()

tm_shape(esp, bbox = c(-13.5, 32, 7, 45)) +
  tm_polygons() +
  tm_shape(gridccaa) +
  tm_polygons("codauto", alpha = 0.6, legend.show = FALSE) +
  tm_shape(gridccaa) +
  tm_text("label")

gridprov <- esp_get_grid_prov()

tm_shape(esp, bbox = c(-13.5, 32, 7, 45)) +
  tm_polygons() +
  tm_shape(gridprov) +
  tm_polygons("cpro", alpha = 0.6, legend.show = FALSE) +
  tm_shape(gridprov) +
  tm_text("label")
```

`esp_get_hydrobasin` *Get the drainage basin demarcations of Spain*

Description

Loads a simple feature (`sf`) object containing areas with the required hydrograpic elements of Spain.

Usage

```
esp_get_hydrobasin(
  epsg = "4258",
  cache = TRUE,
  update_cache = FALSE,
  cache_dir = NULL,
  verbose = FALSE,
  resolution = "3",
  domain = "land"
)
```

Arguments

`epsg`	projection of the map: 4-digit EPSG code. One of: • "4258": ETRS89 • "4326": WGS84 • "3035": ETRS89 / ETRS-LAEA • "3857": Pseudo-Mercator

cache	A logical whether to do caching. Default is TRUE.
update_cache	A logical whether to update cache. Default is FALSE. When set to TRUE it would force a fresh download of the source .geojson file.
cache_dir	A path to a cache directory. The directory can also be set globally with: • options(mapSpain_cache_dir = "path/to/dir"). See Details on esp_get_nuts().
verbose	Display information. Useful for debugging, default is FALSE.
resolution	Resolution of the polygon. Values available are "3", "6.5" or "10".
domain	Possible values are "land", that includes only the ground part or the ground or "landsea", that includes both the ground and the related sea waters of the basin

Details

Metadata available on https://github.com/rOpenSpain/mapSpain/tree/sianedata/.

Value

A POLYGON object.

Author(s)

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

Source

IGN data via a custom CDN (see https://github.com/rOpenSpain/mapSpain/tree/sianedata.

Examples

```
all <- esp_get_prov(moveCAN = FALSE)
hydroland <- esp_get_hydrobasin(domain = "land")
hydrolandsea <- esp_get_hydrobasin(domain = "landsea")

tm_shape(hydrolandsea, bbox = c(-9.5, 35, 4.5, 44)) +
  tm_fill("skyblue3") +
  tm_shape(all) +
  tm_polygons("grey90") +
  tm_shape(hydroland) +
  tm_polygons("skyblue", alpha = 0.7, border.col = "blue") +
  tm_layout(bg.color = "grey90")
```

esp_get_hypsobath *Get hypsometry and bathymetry of Spain*

Description

Loads a simple feature (`sf`) object containing lines or areas with the hypsometry and bathymetry of Spain.

Usage

```
esp_get_hypsobath(
  epsg = "4258",
  cache = TRUE,
  update_cache = FALSE,
  cache_dir = NULL,
  verbose = FALSE,
  resolution = "3",
  spatialtype = "area"
)
```

Arguments

`epsg`	projection of the map: 4-digit EPSG code. One of: • "4258": ETRS89 • "4326": WGS84 • "3035": ETRS89 / ETRS-LAEA • "3857": Pseudo-Mercator
`cache`	A logical whether to do caching. Default is `TRUE`.
`update_cache`	A logical whether to update cache. Default is `FALSE`. When set to `TRUE` it would force a fresh download of the source `.geojson` file.
`cache_dir`	A path to a cache directory. The directory can also be set globally with: • `options(mapSpain_cache_dir = "path/to/dir")`. See Details on `esp_get_nuts()`.
`verbose`	Display information. Useful for debugging, default is `FALSE`.
`resolution`	Resolution of the shape. Values available are "3" or "6.5".
`spatialtype`	Spatial type of the output. Use "area" for `POLYGONS` or "line" for `LINESTRING`.

Details

Metadata available on `https://github.com/rOpenSpain/mapSpain/tree/sianedata/`.

Value

A `POLYGON` or `LINESTRING` object.

Author(s)

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

Source

IGN data via a custom CDN (see https://github.com/rOpenSpain/mapSpain/tree/sianedata.

Examples

```
# This code would produce a nice plot - It will take a few seconds to run
library(tmap)

hypsobath <- esp_get_hypsobath()

# Tints from Wikipedia
# https://en.wikipedia.org/wiki/Wikipedia:WikiProject_Maps/Conventions/Topographic_maps

bath_tints <- colorRampPalette(
  rev(
    c(
      "#D8F2FE", "#C6ECFF", "#B9E3FF",
      "#ACDBFB", "#A1D2F7", "#96C9F0",
      "#8DC1EA", "#84B9E3", "#79B2DE",
      "#71ABD8"
    )
  )
)

hyps_tints <- colorRampPalette(
  rev(
    c(
      "#F5F4F2", "#E0DED8", "#CAC3B8", "#BAAE9A",
      "#AC9A7C", "#AA8753", "#B9985A", "#C3A76B",
      "#CAB982", "#D3CA9D", "#DED6A3", "#E8E1B6",
      "#EFEBC0", "#E1E4B5", "#D1D7AB", "#BDCC96",
      "#A8C68F", "#94BF8B", "#ACD0A5"
    )
  )
)

levels <- sort(unique(hypsobath$val_inf))

# Create palette
br_bath <- length(levels[levels < 0])
br_terrain <- length(levels) - br_bath

bath_tints(br_bath)

pal <- c(bath_tints((br_bath)), hyps_tints((br_terrain)))
```

```
# Plot Canary Islands
tm_shape(hypsobath, bbox = c(-18.6, 27, -13, 29.5)) +
  tm_fill("val_inf",
    style = "cat",
    palette = pal,
    title = "Elevation",
    legend.is.portrait = FALSE
  ) +
  tm_layout(
    legend.outside = TRUE,
    legend.outside.position = "bottom",
    legend.position = c("center", "bottom"),
    legend.text.size = 0.43
  )


# Plot Mainland
tm_shape(hypsobath, bbox = c(-9.5, 35.8, 4.4, 44)) +
  tm_fill("val_inf",
    style = "cat",
    palette = pal,
    title = "Elevation",
    legend.reverse = TRUE
  ) +
  tm_layout(legend.outside = TRUE)
```

`esp_get_munic` *Get municipalities boundaries of Spain*

Description

Loads a simple feature (`sf`) object containing the municipalities boundaries of Spain.

`esp_get_munic` uses GISCO (Eurostat) as source.

`esp_get_munic_siane` uses CartoBase ANE as source, provided by Instituto Geografico Nacional (IGN), `http://www.ign.es/web/ign/portal`. Years available are 2005 up to today.

Usage

```
esp_get_munic(
  year = "2019",
  epsg = "4258",
  cache = TRUE,
  update_cache = FALSE,
  cache_dir = NULL,
  verbose = FALSE,
  region = NULL,
  munic = NULL,
  moveCAN = TRUE
)

esp_get_munic_siane(
  year = Sys.Date(),
  epsg = "4258",
  cache = TRUE,
  update_cache = FALSE,
  cache_dir = NULL,
  verbose = FALSE,
  resolution = 3,
  region = NULL,
  munic = NULL,
  moveCAN = TRUE,
  rawcols = FALSE
)
```

Arguments

`year`	Release year. See Details for years available.
`epsg`	projection of the map: 4-digit EPSG code. One of: • "4258": ETRS89 • "4326": WGS84 • "3035": ETRS89 / ETRS-LAEA • "3857": Pseudo-Mercator
`cache`	A logical whether to do caching. Default is `TRUE`.
`update_cache`	A logical whether to update cache. Default is `FALSE`. When set to `TRUE` it would force a fresh download of the source `.geojson` file.
`cache_dir`	A path to a cache directory. The directory can also be set globally with: • `options(mapSpain_cache_dir = "path/to/dir")`. See Details on `esp_get_nuts()`.
`verbose`	Display information. Useful for debugging, default is `FALSE`.
`region`	A vector of names and/or codes for provinces or `NULL` to get all the municipalities. See Details.
`munic`	A name or regex expression with the names of the required municipalities. `NULL` would not produce any filtering.
`moveCAN`	A logical `TRUE/FALSE` or a vector of coordinates `c(lat,lon)`. It places the Canary Islands close to Spain's mainland. Initial position can be adjusted using the vector of coordinates.
`resolution`	Resolution of the polygon. Values available are "3", "6.5" or "10".
`rawcols`	Logical. Setting this to `TRUE` would add the raw columns of the dataset provided by IGN.

Details

When using `region` you can use and mix names and NUTS codes (levels 1, 2 or 3), ISO codes (corresponding to level 2 or 3) or `cpro`.

When calling a superior level (Province, Autonomous Community or NUTS1) , all the municipalities of that level would be added.

On `esp_get_munic` years available are: 2001, 2004, 2006, 2008, 2010, 2013 and any year between 2016 and 2019.

On `esp_get_munic_siane`, `year` could be passed as a single year ("YYYY" format, as end of year) or as a specific date ("YYYY-MM-DD" format). Historical information starts as of 2005.

Value

A `POLYGON` object.

Author(s)

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

Source

GISCO API

IGN data via a custom CDN (see `https://github.com/rOpenSpain/mapSpain/tree/sianedata`.

See Also

`esp_get_nuts()`, `esp_munic.sf`, `esp_codelist`.

Examples

```
Base <- esp_get_munic(region = c("Castilla y Leon"))
SAN <-
  esp_get_munic(
    region = c("Castilla y Leon"),
    munic = c("^San ", "^Santa ")
  )

library(tmap)
tm_shape(Base) +
  tm_polygons("#FDFBEA", border.col = "#656565", border.alpha = 0.3) +
  tm_shape(SAN) +
  tm_polygons("#C12838", border.col = "#656565") +
  tm_layout(
    main.title = paste0(
      "Municipalities named under Saints (San, Santa)",
      "\nCastilla y Leon, Spain"
    ),
    main.title.size = .8
  )
```

esp_get_nuts *Get NUTS boundaries of Spain*

Description

Loads a simple feature (`sf`) object containing the NUTS boundaries of Spain.

Usage

```
esp_get_nuts(
  year = "2016",
  epsg = "4258",
  cache = TRUE,
  update_cache = FALSE,
  cache_dir = NULL,
  verbose = FALSE,
  resolution = "01",
  spatialtype = "RG",
  region = NULL,
  nuts_level = "all",
  moveCAN = TRUE
)
```

Arguments

`year`	Release year. One of "2003", "2006",‘"2010", "2013", "2016" or "2021".
`epsg`	projection of the map: 4-digit EPSG code. One of: • "4258": ETRS89 • "4326": WGS84 • "3035": ETRS89 / ETRS-LAEA • "3857": Pseudo-Mercator
`cache`	A logical whether to do caching. Default is `TRUE`.
`update_cache`	A logical whether to update cache. Default is `FALSE`. When set to `TRUE` it would force a fresh download of the source `.geojson` file.
`cache_dir`	A path to a cache directory. The directory can also be set globally with: • `options(mapSpain_cache_dir = "path/to/dir")`. See Details on `esp_get_nuts()`.
`verbose`	Display information. Useful for debugging, default is `FALSE`.
`resolution`	Resolution of the geospatial data. One of • "60": 1:60million • "20": 1:20million • "10": 1:10million • "03": 1:3million

- "01": 1:1million

`spatialtype`	Type of geometry to be returned: • "RG": Regions - `MULTIPOLYGON/POLYGON` object. • "LB": Labels - `POINT` object.
`region`	Optional. A vector of region names, NUTS or ISO codes (see `esp_dict_region_code()`.
`nuts_level`	NUTS level. One of "0" (Country-level), "1", "2" or "3". See `https://ec.europa.eu/eurostat/web/nuts/background`.
`moveCAN`	A logical `TRUE/FALSE` or a vector of coordinates `c(lat,lon)`. It places the Canary Islands close to Spain's mainland. Initial position can be adjusted using the vector of coordinates.

Details

`cache_dir = NULL` (default) uses and creates /mapSpain directory in the temporary directory `tempdir()`. The directory can also be set via options with `options(mapSpain_cache_dir = "path/to/dir")` or `options(gisco_cache_dir = "path/to/dir")` (See `giscoR::gisco_get()`)

Sometimes cached files may be corrupt. On that case, try redownloading the data using `update_cache = TRUE`.

Value

A `POLYGON/POINT` object.

Note

Please check the download and usage provisions on `giscoR::gisco_attributions()`

While `moveCAN` is useful for visualization, it would alter the actual geographical position of the Canary Islands. When using the output for spatial analysis or using tiles (`esp_getTiles()` or `addProviderEspTiles()`) this option should be set to `FALSE` in order to get the actual coordinates.

Author(s)

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

Source

GISCO API

See Also

`esp_nuts.sf`, `esp_dict_region_code()`, `esp_codelist`, `giscoR::gisco_get()`.

Examples

```
library(sf)

pal <- hcl.colors(5, palette = "Lisbon")

NUTS1 <- esp_get_nuts(nuts_level = 1, moveCAN = TRUE)
plot(st_geometry(NUTS1), col = pal)

NUTS1_alt <- esp_get_nuts(nuts_level = 1, moveCAN = c(15, 0))
plot(st_geometry(NUTS1_alt), col = pal)

NUTS1_orig <- esp_get_nuts(nuts_level = 1, moveCAN = FALSE)
plot(st_geometry(NUTS1_orig), col = pal)

AndOriental <-
  esp_get_nuts(region = c("Almeria", "Granada", "Jaen", "Malaga"))
plot(st_geometry(AndOriental), col = pal)

RandomRegions <- esp_get_nuts(region = c("ES1", "ES300", "ES51"))
plot(st_geometry(RandomRegions), col = pal)

MixingCodes <- esp_get_nuts(region = c("ES4", "ES-PV", "Valencia"))
plot(st_geometry(MixingCodes), col = pal)
```

`esp_get_prov` *Get Provinces boundaries of Spain*

Description

Loads a simple feature (`sf`) object containing the province boundaries of Spain.

`esp_get_prov` uses GISCO (Eurostat) as source

`esp_get_prov_siane` use CartoBase ANE as source, provided by Instituto Geografico Nacional (IGN), http://www.ign.es/web/ign/portal. Years available are 2005 up to today.

Usage

```
esp_get_prov(prov = NULL, ...)

esp_get_prov_siane(
  prov = NULL,
  year = Sys.Date(),
  epsg = "4258",
  cache = TRUE,
  update_cache = FALSE,
  cache_dir = NULL,
  verbose = FALSE,
  resolution = "3",
  moveCAN = TRUE,
  rawcols = FALSE
)
```

Arguments

`prov`	A vector of names and/or codes for provinces or `NULL` to get all the provinces. See Details.
`...`	Additional parameters from `esp_get_nuts()`.
`year`	Release year. See `esp_get_nuts()` for `esp_get_prov` and Details for `esp_get_prov_siane`
`epsg`	projection of the map: 4-digit EPSG code. One of: • "4258": ETRS89 • "4326": WGS84 • "3035": ETRS89 / ETRS-LAEA • "3857": Pseudo-Mercator
`cache`	A logical whether to do caching. Default is `TRUE`.
`update_cache`	A logical whether to update cache. Default is `FALSE`. When set to `TRUE` it would force a fresh download of the source `.geojson` file.
`cache_dir`	A path to a cache directory. The directory can also be set globally with: • `options(mapSpain_cache_dir = "path/to/dir")`. See Details on `esp_get_nuts()`.
`verbose`	Display information. Useful for debugging, default is `FALSE`.
`resolution`	Resolution of the polygon. Values available are "3", "6.5" or "10".
`moveCAN`	A logical `TRUE/FALSE` or a vector of coordinates `c(lat,lon)`. It places the Canary Islands close to Spain's mainland. Initial position can be adjusted using the vector of coordinates.
`rawcols`	Logical. Setting this to `TRUE` would add the raw columns of the dataset provided by IGN.

Details

When using `prov` you can use and mix names and NUTS codes (levels 1, 2 or 3), ISO codes (corresponding to level 2 or 3) or `cpro`.

Ceuta and Melilla are considered as provinces on this dataset.

When calling a superior level (Autonomous Community or NUTS1) , all the provinces of that level would be added.

On `esp_get_prov_siane`, `year` could be passed as a single year ("YYYY" format, as end of year) or as a specific date ("YYYY-MM-DD" format). Historical information starts as of 2005.

Value

A `POLYGON/POINT` object.

Author(s)

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

Source

IGN data via a custom CDN (see https://github.com/rOpenSpain/mapSpain/tree/sianedata).

See Also

esp_get_hex_prov(), esp_get_nuts(), esp_get_ccaa(), esp_get_munic(), esp_codelist.

Examples

```
library(sf)

# Random Provinces

Random <-
  esp_get_prov(prov = c(
    "Zamora",
    "Palencia",
    "ES-GR",
    "ES521",
    "01"
  ))
plot(st_geometry(Random), col = hcl.colors(6))

# All Provinces of a Zone plus an addition
# Low resolution (20M)

Mix <-
  esp_get_prov(
    prov = c(
      "Noroeste",
      "Castilla y Leon", "La Rioja"
    ),
    resolution = "20"
  )
plot(
  Mix[, "nuts1.code"],
  pal = hcl.colors(3),
  key.pos = NULL,
  main = NULL,
  border = "white"
)
# ISO codes available

allprovs <- esp_get_prov()

library(tmap)
```

```
tmap_style("cobalt")

tm_shape(allprovs, point.per = "feature") +
  tm_polygons() +
  tm_text("iso2.prov.code", remove.overlap = TRUE)

tmap_options_reset()
```

`esp_get_railway` *Get railways of Spain*

Description

Extract nodes and railways of mainland Spain and the Balearic Islands.

Usage

```
esp_get_railway(
  year = Sys.Date(),
  epsg = "4258",
  cache = TRUE,
  update_cache = FALSE,
  cache_dir = NULL,
  verbose = FALSE,
  spatialtype = "line"
)
```

Arguments

`year`	Release year.
`epsg`	projection of the map: 4-digit EPSG code. One of: • "4258": ETRS89 • "4326": WGS84 • "3035": ETRS89 / ETRS-LAEA • "3857": Pseudo-Mercator
`cache`	A logical whether to do caching. Default is `TRUE`.
`update_cache`	A logical whether to update cache. Default is `FALSE`. When set to `TRUE` it would force a fresh download of the source `.geojson` file.
`cache_dir`	A path to a cache directory. The directory can also be set globally with: • `options(mapSpain_cache_dir = "path/to/dir")`. See Details on `esp_get_nuts()`.
`verbose`	Display information. Useful for debugging, default is `FALSE`.
`spatialtype`	Spatial type of the output. Use "line" for extracting the railway and "point" for the stations.

Details

Details on caching can be found on `esp_get_nuts()`

Value

A `MULTILINESTRING` or `POINT` object.

Author(s)

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

Source

IGN data via a custom CDN (see `https://github.com/rOpenSpain/mapSpain/tree/sianedata`.

See Also

`esp_get_roads()`

Examples

```
provs <- esp_get_prov()
ccaa <- esp_get_ccaa()

# Railways
rails <- esp_get_railway()

# Stations
stations <- esp_get_railway(spatialtype = "point")

# Map

library(tmap)

tm_shape(provs, bbox = c(-7.5, 38, -2.5, 41)) +
  tm_polygons(col = "grey90", border.col = "grey50") +
  tm_shape(ccaa) +
  tm_borders("black") +
  tm_shape(rails) +
  tm_lines("tipo",
    legend.col.show = FALSE, lwd = 2,
    palette = "Dark2"
  ) +
  tm_shape(stations) +
  tm_symbols("red", size = .3)
```

`esp_get_rivers` *Get rivers, channels, reservoirs and other wetlands of Spain*

Description

Loads a simple feature (`sf`) object containing lines or areas with the required hydrograpic elements of Spain.

Usage

```
esp_get_rivers(
  epsg = "4258",
  cache = TRUE,
  update_cache = FALSE,
  cache_dir = NULL,
  verbose = FALSE,
  resolution = "3",
  spatialtype = "line",
  name = NULL
)
```

Arguments

`epsg`	projection of the map: 4-digit EPSG code. One of: • "4258": ETRS89 • "4326": WGS84 • "3035": ETRS89 / ETRS-LAEA • "3857": Pseudo-Mercator
`cache`	A logical whether to do caching. Default is `TRUE`.
`update_cache`	A logical whether to update cache. Default is `FALSE`. When set to `TRUE` it would force a fresh download of the source `.geojson` file.
`cache_dir`	A path to a cache directory. The directory can also be set globally with: • `options(mapSpain_cache_dir = "path/to/dir")`. See Details on `esp_get_nuts()`.
`verbose`	Display information. Useful for debugging, default is `FALSE`.
`resolution`	Resolution of the polygon. Values available are "3", "6.5" or "10".
`spatialtype`	Spatial type of the output. Use "area" for `POLYGONS` or "line" for `LINESTRING`.
`name`	Optional. A character or regex expresion with the name of the element to be extracted. See Details

Details

Metadata available on `https://github.com/rOpenSpain/mapSpain/tree/sianedata/`.

`name` admits regex expressions. See `help("regex",package = "base")` for more information.

Value

A `POLYGON` or `LINESTRING` object.

Author(s)

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

Source

IGN data via a custom CDN (see https://github.com/rOpenSpain/mapSpain/tree/sianedata.

Examples

```
# This code would produce a nice plot - It will take a few seconds to run
library(sf)

# Use of regex

regex1 <- esp_get_rivers(name = "Tajo|Segura")
unique(regex1$rotulo)

regex2 <- esp_get_rivers(name = "Tajo$| Segura")
unique(regex2$rotulo)

# Rivers in Spain
shapeEsp <- esp_get_country(moveCAN = FALSE)

MainRivers <-
  esp_get_rivers(name = "Tajo$|Ebro$|Ebre$|Duero|Guadiana$|Guadalquivir")

opar <- par(no.readonly = TRUE)
par(mar = c(0, 0, 0, 0))

plot(st_geometry(MainRivers), col = "skyblue", lwd = 1.5)
plot(st_geometry(shapeEsp), col = NA, add = TRUE)

# All wetlands

Wetlands <- esp_get_rivers(spatialtype = "area")
plot(st_geometry(Wetlands), col = "skyblue", border = NA)
plot(st_geometry(shapeEsp), col = NA, add = TRUE)

par(opar)
```

esp_get_roads *Get the roads of Spain*

Description

Get roads of Spain

Usage

```
esp_get_roads(
  year = Sys.Date(),
  epsg = "4258",
  cache = TRUE,
  update_cache = FALSE,
  cache_dir = NULL,
  verbose = FALSE,
  moveCAN = TRUE
)
```

Arguments

`year`	Release year. See Details for years available.
`epsg`	projection of the map: 4-digit EPSG code. One of: • "4258": ETRS89 • "4326": WGS84 • "3035": ETRS89 / ETRS-LAEA • "3857": Pseudo-Mercator
`cache`	A logical whether to do caching. Default is `TRUE`.
`update_cache`	A logical whether to update cache. Default is `FALSE`. When set to `TRUE` it would force a fresh download of the source `.geojson` file.
`cache_dir`	A path to a cache directory. The directory can also be set globally with: • `options(mapSpain_cache_dir = "path/to/dir")`. See Details on `esp_get_nuts()`.
`verbose`	Display information. Useful for debugging, default is `FALSE`.
`moveCAN`	A logical `TRUE/FALSE` or a vector of coordinates `c(lat,lon)`. It places the Canary Islands close to Spain's mainland. Initial position can be adjusted using the vector of coordinates.

Details

`year` could be passed as a single year ("YYYY" format, as end of year) or as a specific date ("YYYY-MM-DD" format).

Details on caching can be found on `esp_get_nuts()`

Value

A LINESTRING\MULTILINESTRING object.

Note

While `moveCAN` is useful for visualization, it would alter the actual geographical position of the Canary Islands.

Author(s)

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

Source

IGN data via a custom CDN (see `https://github.com/rOpenSpain/mapSpain/tree/sianedata`.

See Also

`esp_get_munic()`, esp_munic.sf, esp_codelist

Examples

```
CyL <- esp_get_prov("Castilla y Leon")
Roads <- esp_get_roads()

# Intersect roads
CyL_Roads <- st_intersection(CyL, Roads)

library(tmap)

tm_shape(CyL) +
  tm_polygons(col = "grey80", border.col = "grey50", lwd = 0.4) +
  tm_shape(CyL_Roads) +
  tm_lines("tipo",
    palette = c("#003399", "#003399", "#ff0000", "#ffff00")
  ) +
  tm_layout(legend.outside = TRUE, legend.outside.position = "bottom")
```

`esp_munic.sf` *All Municipalities* `POLYGON` *object of Spain*

Description

A `sf` object including all municipalities of Spain as provided by GISCO (2019 version).

Format

A POLYGON data frame (resolution: 1:1million, EPSG:4258) object:

- codauto: INE code of each autonomous community.
- ine.ccaa.name: INE name of each autonomous community.
- cpro: INE code of each province.
- ine.prov.name: INE name of each province.
- cmun: INE code of each municipality.
- name: Name of the municipality.
- LAU_CODE: LAU Code (GISCO) of the municipality.
- geometry: geometry field.

Source

https://ec.europa.eu/eurostat/web/gisco/geodata/reference-data/, LAU 2019 data.

See Also

esp_get_munic()

Examples

```
data("esp_munic.sf")
data("esp_nuts.sf")

Teruel.cpro <- esp_dict_region_code("Teruel", destination = "cpro")
Teruel.NUTS <- esp_dict_region_code(Teruel.cpro,
  origin = "cpro",
  destination = "nuts"
)

Teruel.sf <- esp_munic.sf[esp_munic.sf$cpro == Teruel.cpro, ]
Teruel.city <- Teruel.sf[Teruel.sf$name == "Teruel", ]

# Extract CCAA

CCAA <- esp_get_prov()


library(tmap)

tm_shape(CCAA, bbox = Teruel.sf) +
  tm_polygons("#DFDFDF") +
  tm_shape(Teruel.sf) +
  tm_polygons("#FDFBEA") +
  tm_shape(Teruel.city) +
  tm_fill("#C12838") +
  tm_layout(main.title = "Municipalities of Teruel")
```

esp_nuts.sf *All NUTS POLYGON object of Spain*

Description

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

Format

A POLYGON data frame (resolution: 1:1million, EPSG:4258) object with 86 rows and fields:

- COAST_TYPE: COAST_TYPE
- FID: FID
- NUTS_NAME: NUTS name on local alphabet
- MOUNT_TYPE: MOUNT_TYPE
- NAME_LATN: Name on Latin characters
- CNTR_CODE: Eurostat Country code
- URBN_TYPE: URBN_TYPE
- NUTS_ID: NUTS identifier
- LEVL_CODE: NUTS level code (0,1,2,3)
- geometry: geometry field

Source

https://gisco-services.ec.europa.eu/distribution/v2/nuts/, file NUTS_RG_20M_2016_4326.geojson.

See Also

esp_get_nuts()

Examples

```
nuts <- esp_nuts.sf

# Select NUTS 3
nuts3 <- esp_nuts.sf[esp_nuts.sf$LEVL_CODE == 3, ]

# Combine with full shape

spain <- esp_get_country(moveCAN = FALSE)

library(tmap)

tm_shape(nuts3) +
  tm_fill(
    "URBN_TYPE",
    style = "cat",
    title = "Urban Type",
    palette = hcl.colors(3, "Teal")
  ) +
  tm_shape(spain) +
  tm_borders(col = "black") +
  tm_graticules(lines = FALSE) +
  tm_layout(
    main.title = "NUTS3 levels of Spain",
    legend.outside = TRUE
  )
```

leaflet.providersESP.df

Public WMS and WMTS of Spain

Description

A data frame containing information of different public WMS and WMTS providers of Spain

This function is a implementation of the javascript plugin leaflet-providersESP **v1.2.0**.

Format

A data frame object with a list of the required parameters for calling the service:

- provider: Provider name.
- field: Description of `value`.
- value: INE code of each province.

Details

Providers available to be passed to `type` are:

- **IDErioja:** "IDErioja"
- **IGNBase:** "IGNBase.Todo", "IGNBase.Gris", "IGNBase.TodoNoFondo, IGNBase.Orto"
- **MDT:** "MDT.Elevaciones", "MDT.Relieve", "MDT.CurvasNivel"
- **PNOA:** "PNOA.MaximaActualidad", "PNOA.Mosaico"
- **OcupacionSuelo:** "OcupacionSuelo.Ocupacion", "OcupacionSuelo.Usos"
- **LiDAR:** "LiDAR"
- **MTN:** "MTN"
- **Geofisica:** "Geofisica.Terremotos10dias", "Geofisica.Terremotos30dias", "Geofisica.Terremotos365dias", "Geofisica.VigilanciaVolcanica"

- **CaminoDeSantiago:** "CaminoDeSantiago.CaminoFrances", "CaminoDeSantiago.CaminosTuronensis", "CaminoDeSantiago.CaminosGalicia", "CaminoDeSantiago.CaminosDelNorte", "CaminoDeSantiago.CaminosAndaluces", "CaminoDeSantiago.CaminosCentro", "CaminoDeSantiago.CaminosEste", "CaminoDeSantiago.CaminosCatalanes", "CaminoDeSantiago.CaminosSureste", "CaminoDeSantiago.CaminosInsulares", "CaminoDeSantiago.CaminosPiemonts", "CaminoDeSantiago.CaminosTolosana", "CaminoDeSantiago.CaminosPortugueses"
- **Catastro:** "Catastro.Catastro", "Catastro.Parcela", "Catastro.CadastralParcel", "Catastro.CadastralZoning", "Catastro.Address", "Catastro.Building"
- **RedTransporte:** "RedTransporte.Carreteras", "RedTransporte.Ferroviario", "RedTransporte.Aerodromo", "RedTransporte.AreaServicio", "RedTransporte.EstacionesFerroviario", "RedTransporte.Puertos"
- **Cartociudad:** "Cartociudad.CodigosPostales", "Cartociudad.Direcciones"
- **NombresGeograficos:** "NombresGeograficos"
- **UnidadesAdm:** "UnidadesAdm.Limites", "UnidadesAdm.Unidades"
- **Hidrografia:** "Hidrografia.MasaAgua", "Hidrografia.Cuencas", "Hidrografia.Subcuencas", "Hidrografia.POI", "Hidrografia.ManMade", "Hidrografia.LineaCosta", "Hidrografia.Rios", "Hidrografia.Humedales"
- **Militar:** "Militar.CEGET1M", "Militar.CEGETM7814", "Militar.CEGETM7815", "Militar.CEGETM682", "Militar.CECAF1M"
- **ADIF:** "ADIF.Vias", "ADIF.Nodos", "ADIF.Estaciones"
- **LimitesMaritimos:** "LimitesMaritimos.LimitesMaritimos", "LimitesMaritimos.LineasBase"
- **Copernicus:** "Copernicus.LandCover", "Copernicus.Forest", "Copernicus.ForestLeaf", "Copernicus.WaterWet", "Copernicus.SoilSeal", "Copernicus.GrassLand", "Copernicus.Local", "Copernicus.RiparianGreen", "Copernicus.RiparianLandCover", "Copernicus.Natura2k", "Copernicus.UrbanAtlas"
- **ParquesNaturales:** "ParquesNaturales.Limites", "ParquesNaturales.ZonasPerifericas"

Source

`https://dieghernan.github.io/leaflet-providersESP/` leaflet plugin, **v1.2.0**.

See Also

`esp_getTiles()`, `addProviderEspTiles()`.

`pobmun19` *Population by municipality (2019)*

Description

A data frame with 8131 rows containing the population data by municipality in Spain (2019).

Source

INE: Instituto Nacional de Estadistica `https://www.ine.es/`

Index