Package 'mapSpain'

September 10, 2021

Type Package

Title Administrative Boundaries of Spain

Version 0.3.1

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

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

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

Depends R (>= 3.6.0)

Imports countrycode (>= 1.2.0), giscoR (>= 0.2.4), leaflet (>= 2.0.0), png (>= 0.1-5), rappdirs (>= 0.3.0), raster (>= 3.0), sf (>= 0.9), slippymath (>= 0.3.1), utils

Suggests knitr, rgdal, rmarkdown, testthat (>= 3.0.0), tibble, tmap (>= 3.0.0)

VignetteBuilder knitr

Config/testthat/edition 3

Encoding UTF-8

LazyData true

RoxygenNote 7.1.2

X-schema.org-applicationCategory cartography

X-schema.org-isPartOf https://ropenspain.es/

X-schema.org-keywords rOpenSpain, tiles, r, maps, spatial, rstats, r-package, municipalities, Spain, gisco, provinces, ign, administrative-boundaries, ccaa, static-tiles

NeedsCompilation no

Author Diego Hernangómez [aut, cre, cph]
(<https://orcid.org/0000-0001-8457-4658>, rOpenSpain),
EuroGeographics [cph] (for the administrative boundaries.),
Instituto Geográfico Nacional [cph] (for the administrative
boundaries.)

Maintainer Diego Hernangómez <diego.hernangomezherrero@gmail.com>

Repository CRAN

Date/Publication 2021-09-10 12:10:06 UTC

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	0.3.1
Date	2021
License	GPL-3
LazyLoad	yes

COPYRIGHT NOTICE (IGN)

This package uses data from CartoBase SIANE, provided by Instituto Geografico Nacional (IGN):

Atlas Nacional de España (ANE) CC BY 4.0 ign.es

COPYRIGHT NOTICE (GISCO)

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.

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 *Include base tiles of Spanish public administrations on a* **leaflet** *map*

Description

Include tiles of public Spanish organisms to a `leaflet::leaflet()` map.

Usage

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

providerEspTileOptions(...)
```

Arguments

map	A map widget created from `leaflet::leaflet()`.
provider	Name of the provider, see leaflet.providersESP.df for values available.
layerId	the layer id
group	The name of the group the newly created layers should belong to 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. See `leaflet::addTiles()`.
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

`providerEspTileOptions()` is a wrapper of `leaflet::providerTileOptions()`.

Value

A map object generated with `leaflet::leaflet()`.

Source

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

See Also

`leaflet::leaflet()`, `leaflet::addTiles()`

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

Other imagery utilities: `esp_getTiles()`, `leaflet.providersESP.df`

Examples

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

PuertadelSol
```

---

esp_clear_cache *Clear your* **mapSpain** *cache dir*

Description

Use this function with caution. This function would clear your cached data and configuration, specifically:

- Deletes the **mapSpain** config directory (`rappdirs::user_config_dir("mapSpain","R")`).
- Deletes the `cache_dir` directory.
- Deletes the values on stored on `Sys.getenv("MAPSPAIN_CACHE_DIR")` and `options(mapSpain_cache_dir)`.

Usage

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

Arguments

config	if TRUE, will delete the configuration folder of **mapSpain**.
cached_data	If this is set to TRUE, it will delete your cache_dir and all its content.
verbose	Logical, displays information. Useful for debugging, default is FALSE.

Details

This is an overkill function that is intended to reset your status as it you would never have installed and/or used **mapSpain**.

Value

Invisible. This function is called for its side effects.

See Also

Other cache utilities: esp_set_cache_dir()

Examples

```
# Don't run this! It would modify your current state
## Not run:
esp_clear_cache(verbose = TRUE)

## End(Not run)

Sys.getenv("MAPSPAIN_CACHE_DIR")
```

---

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

A data frame with 59 rows 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

See Also

Other datasets: esp_munic.sf, esp_nuts.sf, leaflet.providersESP.df, pobmun19

Other political: esp_get_can_box(), esp_get_capimun(), esp_get_ccaa(), esp_get_country(), esp_get_gridmap, esp_get_munic(), esp_get_nuts(), esp_get_prov()

Other dictionary: esp_dict_region_code()

Examples

```
data("esp_codelist")

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`.

See Also

Other dictionary: `esp_codelist`

Other dictionary: `esp_codelist`

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 static 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,
  zoommin = 0,
  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. If set, it overrides zoommin. Only valid for WMTS tiles. On a single point it applies a buffer to the point and on zoom = NULL the function set a zoom level of 18. See **Details**.
zoommin	Delta on default zoom. The default value is designed to download fewer tiles than you probably want. Use 1 or 2 to increase the resolution.
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 file.
cache_dir	A path to a cache directory. See **About caching**.
verbose	Logical, displays information. Useful for debugging, default is FALSE.

Details

Zoom levels are described on the OpenStreetMap wiki:

zoom	area to represent
0	whole world
3	large country
5	state
8	county
10	metropolitan area
11	city
13	village or suburb
16	streets
18	some buildings, trees

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)
```

Value

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

About caching

You can set your cache_dir with esp_set_cache_dir().

Sometimes cached files may be corrupt. On that case, try re-downloading the data setting update_cache = TRUE.

If you experience any problem on download, try to download the corresponding .geojson file by any other method and save it on your cache_dir. Use the option verbose = TRUE for debugging the API query.

Source

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

For plotting, you can use raster::plotRGB(), tmap::tm_rgb().

See Also

raster::brick().

Other imagery utilities: addProviderEspTiles(), leaflet.providersESP.df

Examples

```
## Not run:
# This script downloads tiles to your local machine
# Run only if you are online

Murcia <- esp_get_ccaa_siane("Murcia", epsg = 3857)
```

```
Tile <- esp_getTiles(Murcia)

library(tmap)

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

## End(Not run)
```

---

esp_get_can_box *Get* sf *lines and polygons for insetting the 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 lines and polygons to be used when the Canary Islands are displayed as an inset.

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

Value

A sf polygon or line depending of style parameter.

esp_get_can_provinces returns a LINESTRING object.

Displacing the Canary Islands

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

Source

esp_get_can_provinces extracted from CartoBase ANE, se89_mult_admin_provcan_l.shp file.

See Also

Other political: esp_codelist, esp_get_capimun(), esp_get_ccaa(), esp_get_country(), esp_get_gridmap, esp_get_munic(), esp_get_nuts(), esp_get_prov()

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)

# Low resolution map
res <- "20"

Countries <-
  gisco_get_countries(
    res = res,
    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 = res,
  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* sf *points of the municipalities of Spain*

Description

Get a sf point with 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 municipality, 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`. See **About caching**.
`update_cache`	A logical whether to update cache. Default is `FALSE`. When set to `TRUE` it would force a fresh download of the source file.
`cache_dir`	A path to a cache directory. See **About caching**.
`verbose`	Logical, displays 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. See **Displacing the Canary Islands**.
`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". See esp_codelist

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

Value

A `sf` point object.

About caching

You can set your `cache_dir` with `esp_set_cache_dir()`.

Sometimes cached files may be corrupt. On that case, try re-downloading the data setting `update_cache = TRUE`.

If you experience any problem on download, try to download the corresponding .geojson file by any other method and save it on your `cache_dir`. Use the option `verbose = TRUE` for debugging the API query.

Displacing the Canary Islands

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

Source

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

See Also

Other political: `esp_codelist`, `esp_get_can_box()`, `esp_get_ccaa()`, `esp_get_country()`, `esp_get_gridmap`, `esp_get_munic()`, `esp_get_nuts()`, `esp_get_prov()`

Other municipalities: `esp_get_munic()`, `esp_munic.sf`

Examples

```
## Not run:
# This code compares centroids of municipalities against esp_get_capimun
# It also download tiles, make sure you are online

library(sf)

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

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

# Extract centroid
centroid <- sf::st_centroid(area)
centroid$type <- "Centroid"
```

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

# Get a tile to check
tile <- esp_getTiles(area, zoommin = 2)

# Join both point geometries
points <- rbind(
  centroid[, "type"],
  capimun[, "type"]
)

# Check on plot
library(tmap)

tm_shape(tile, raster.downsample = FALSE) +
  tm_rgb() +
  tm_shape(area) +
  tm_borders(col = "grey40") +
  tm_shape(points) +
  tm_symbols(col = "type", alpha = 0.8, pal = "RdBu") +
  tm_layout(
    main.title = "Centroid vs. capimun",
    legend.outside = TRUE,
    legend.outside.size = 0.3,
    legend.text.size = 1
  )

## End(Not run)
```

---

esp_get_ccaa *Get Autonomous Communities of Spain as* sf *polygons and points*

Description

Returns Autonomous Communities of Spain as polygons and points at a specified scale.

- esp_get_ccaa() uses GISCO (Eurostat) as source. Please use giscoR::gisco_attributions()
- 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: • "LB": Labels - point object. • "RG": Regions - polygon 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. See **About caching**.
update_cache	A logical whether to update cache. Default is FALSE. When set to TRUE it would force a fresh download of the source file.
cache_dir	A path to a cache directory. See **About caching**.
verbose	Logical, displays 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. See **Displacing the Canary Islands**.
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" (see esp_codelist). Ceuta and Melilla are considered as Autonomous Communities on this function.

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 sf object specified by spatialtype.

About caching

You can set your cache_dir with esp_set_cache_dir().

Sometimes cached files may be corrupt. On that case, try re-downloading the data setting update_cache = TRUE.

If you experience any problem on download, try to download the corresponding .geojson file by any other method and save it on your cache_dir. Use the option verbose = TRUE for debugging the API query.

Displacing the Canary Islands

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

Source

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

See Also

Other political: esp_codelist, esp_get_can_box(), esp_get_capimun(), esp_get_country(), esp_get_gridmap, esp_get_munic(), esp_get_nuts(), esp_get_prov()

Examples

```
ccaa <- esp_get_ccaa()

library(tmap)

qtm(ccaa)

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


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"))

qtm(Mix)

# 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_graticules() +
  tm_polygons("#DFDFDF", border.col = "#656565") +
  tm_shape(ccaa) +
  tm_polygons("#FDFBEA", border.col = "#656565") +
  tm_layout(bg.color = "#C7E7FB")
```

`esp_get_country` *Get the borders of Spain as a* `sf` *polygon*

Description

Returns the boundaries of Spain as a single `sf` polygon at a specified scale.

Usage

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

Arguments

`...` Arguments passed on to `esp_get_nuts`

`year` Release year of the file. 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`. See **About caching**.

`update_cache` A logical whether to update cache. Default is `FALSE`. When set to `TRUE` it would force a fresh download of the source file.

`cache_dir` A path to a cache directory. See **About caching**.

`verbose` Logical, displays 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

`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. See **Displacing the Canary Islands**.

Value

A `sf` polygon object.

About caching

You can set your `cache_dir` with `esp_set_cache_dir()`.

Sometimes cached files may be corrupt. On that case, try re-downloading the data setting `update_cache = TRUE`.

If you experience any problem on download, try to download the corresponding .geojson file by any other method and save it on your `cache_dir`. Use the option `verbose = TRUE` for debugging the API query.

Displacing the Canary Islands

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

See Also

Other political: esp_codelist, esp_get_can_box(), esp_get_capimun(), esp_get_ccaa(), esp_get_gridmap, esp_get_munic(), esp_get_nuts(), esp_get_prov()

Examples

```
OriginalCan <- esp_get_country(moveCAN = FALSE)

# One row only

nrow(OriginalCan)

library(tmap)
qtm(OriginalCan, fill = "grey70")

# Less resolution

MovedCan <- esp_get_country(moveCAN = TRUE, resolution = "20")

qtm(MovedCan, fill = "grey70")
```

---

esp_get_gridmap *Get a* sf *hexbin or squared polygon 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.

You can use and mix names, ISO codes, "codauto"/"cpro" codes (see esp_codelist) and NUTS codes of different levels.

When using a code corresponding of a higher level (e.g. `esp_get_prov("Andalucia")`) all the corresponding units of that level are provided (in this case , all the provinces of Andalucia).

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

Value

A `sf` POLYGON object.

About caching

You can set your `cache_dir` with `esp_set_cache_dir()`.

Sometimes cached files may be corrupt. On that case, try re-downloading the data setting `update_cache = TRUE`.

If you experience any problem on download, try to download the corresponding .geojson file by any other method and save it on your `cache_dir`. Use the option `verbose = TRUE` for debugging the API query.

Displacing the Canary Islands

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

See Also

Other political: `esp_codelist`, `esp_get_can_box()`, `esp_get_capimun()`, `esp_get_ccaa()`, `esp_get_country()`, `esp_get_munic()`, `esp_get_nuts()`, `esp_get_prov()`

Examples

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

library(tmap)


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") +
  tm_layout(main.title = "Hexbin: CCAA")



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") +
  tm_layout(main.title = "Hexbin: Provinces")



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") +
  tm_layout(main.title = "Grid: CCAA")

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") +
  tm_layout(main.title = "Grid: Provinces")
```

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

Description

Loads a `sf` polygon object containing areas with the required hydrographic 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. See **About caching**.
update_cache	A logical whether to update cache. Default is FALSE. When set to TRUE it would force a fresh download of the source file.
cache_dir	A path to a cache directory. See **About caching**.
verbose	Logical, displays 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 sf polygon object.

About caching

You can set your cache_dir with esp_set_cache_dir().

Sometimes cached files may be corrupt. On that case, try re-downloading the data setting update_cache = TRUE.

If you experience any problem on download, try to download the corresponding .geojson file by any other method and save it on your cache_dir. Use the option verbose = TRUE for debugging the API query.

Source

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

See Also

Other natural: esp_get_hypsobath(), esp_get_rivers()

Examples

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

library(tmap)

tm_shape(hydrolandsea, bbox = c(-9.5, 35, 4.5, 44)) +
  tm_fill("skyblue4") +
  tm_shape(all) +
  tm_polygons("grey90") +
  tm_shape(hydroland) +
  tm_polygons("skyblue", alpha = 0.5, border.col = "blue") +
  tm_text(
    text = "rotulo",
    remove.overlap = TRUE,
    size = 0.5,
    fontface = "bold",
    shadow = TRUE
  ) +
  tm_layout(bg.color = "grey95")
```

---

esp_get_hypsobath *Get* sf *polygons and lines with the hypsometry and bathymetry of Spain*

Description

Loads a sf polygon or line object representing the hypsometry and bathymetry of Spain.

- **Hypsometry** represents the the elevation and depth of features of the Earth's surface relative to mean sea level.
- **Bathymetry** is the measurement of the depth of water in oceans, rivers, or lakes.

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. See **About caching**.
update_cache	A logical whether to update cache. Default is FALSE. When set to TRUE it would force a fresh download of the source file.
cache_dir	A path to a cache directory. See **About caching**.
verbose	Logical, displays 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 lines.

Details

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

Value

A sf polygon or line object.

About caching

You can set your cache_dir with esp_set_cache_dir().

Sometimes cached files may be corrupt. On that case, try re-downloading the data setting update_cache = TRUE.

If you experience any problem on download, try to download the corresponding .geojson file by any other method and save it on your cache_dir. Use the option verbose = TRUE for debugging the API query.

Source

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

See Also

Other natural: esp_get_hydrobasin(), esp_get_rivers()

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.reverse = TRUE
  ) +
```

```
  tm_layout(
    legend.outside = TRUE
  )


# 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 of Spain as* `sf` *polygons*

Description

Returns municipalities of Spain as polygons at a specified scale.

- `esp_get_munic()` uses GISCO (Eurostat) as source. Please use `giscoR::gisco_attributions()`
- `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. See **About caching**.
update_cache	A logical whether to update cache. Default is FALSE. When set to TRUE it would force a fresh download of the source file.
cache_dir	A path to a cache directory. See **About caching**.
verbose	Logical, displays 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. See **Displacing the Canary Islands**.
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

The years available are:

- esp_get_munic(): year could be one of "2001", "2004", "2006", "2008", "2010", "2013" and any year between 2016 and 2019. See giscoR::gisco_get_lau(), giscoR::gisco_get_communes().
- 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.

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" (see esp_codelist).

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

Value

A sf polygon

About caching

You can set your cache_dir with esp_set_cache_dir().

Sometimes cached files may be corrupt. On that case, try re-downloading the data setting update_cache = TRUE.

If you experience any problem on download, try to download the corresponding .geojson file by any other method and save it on your cache_dir. Use the option verbose = TRUE for debugging the API query.

Displacing the Canary Islands

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

Source

GISCO API

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

See Also

giscoR::gisco_get_lau(), base::regex().

Other political: esp_codelist, esp_get_can_box(), esp_get_capimun(), esp_get_ccaa(), esp_get_country(), esp_get_gridmap, esp_get_nuts(), esp_get_prov()

Other municipalities: esp_get_capimun(), esp_munic.sf

Examples

```
# Get munics
Base <- esp_get_munic(year = "2019", region = "Castilla y Leon")

# Provs for delimiting
provs <- esp_get_prov(prov = "Castilla y Leon")

# Load population data
data("pobmun19")

# Arrange and create breaks

Base_pop <- merge(Base, pobmun19,
  by = c("cpro", "cmun"),
  all.x = TRUE
)
```

```
br <- sort(c(
  0, 50, 100, 200, 500,
  1000, 5000, 50000, 100000,
  max(Base_pop$pob19)
))

# Plot
library(tmap)
tm_shape(Base_pop) +
  tm_fill(
    col = "pob19", palette = "cividis",
    breaks = br,
    title = "Persons"
  ) +
  tm_shape(provs) +
  tm_borders(col = "white", alpha = 0.25) +
  tm_layout(
    legend.outside = TRUE,
    legend.position = c("right", "center"),
    main.title = "Population in Castilla y Leon (2019)",
    frame = FALSE
  )
```

`esp_get_nuts` *Get NUTS of Spain as* `sf` *polygons and points*

Description

Returns NUTS regions of Spain as polygons and points at a specified scale, as provided by GISCO (Geographic Information System of the Commission, depending of Eurostat).

NUTS are provided at three different levels:

- **"0"**: Country level
- **"1"**: Groups of autonomous communities
- **"2"**: Autonomous communities
- **"3"**: Roughly matches the provinces, but providing specific individual objects for each major island

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 of the file. 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. See **About caching**.
update_cache	A logical whether to update cache. Default is FALSE. When set to TRUE it would force a fresh download of the source file.
cache_dir	A path to a cache directory. See **About caching**.
verbose	Logical, displays 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: • "LB": Labels - point object. • "RG": Regions - polygon 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 **Description**.
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. See **Displacing the Canary Islands**.

Value

A sf object specified by spatialtype.

About caching

You can set your cache_dir with esp_set_cache_dir().

Sometimes cached files may be corrupt. On that case, try re-downloading the data setting update_cache = TRUE.

If you experience any problem on download, try to download the corresponding .geojson file by any other method and save it on your cache_dir. Use the option verbose = TRUE for debugging the API query.

Displacing the Canary Islands

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

Note

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

Source

GISCO API

See Also

giscoR::gisco_get_nuts(), esp_dict_region_code().

Other political: esp_codelist, esp_get_can_box(), esp_get_capimun(), esp_get_ccaa(), esp_get_country(), esp_get_gridmap, esp_get_munic(), esp_get_prov()

Other nuts: esp_nuts.sf

Examples

```
NUTS1 <- esp_get_nuts(nuts_level = 1, moveCAN = TRUE)


library(tmap)

tm_shape(NUTS1) +
  tm_graticules() +
  tm_polygons() +
  tm_credits(giscoR::gisco_attributions(),
    fontface = "italic",
    size = 0.7
  ) +
  tm_layout(
    main.title = "NUTS1: Displacing Canary Islands",
    main.title.size = 0.9,
    main.title.fontface = "bold",
```

```
    attr.outside = TRUE
  )

NUTS1_alt <- esp_get_nuts(nuts_level = 1, moveCAN = c(15, 0))

tm_shape(NUTS1_alt) +
  tm_graticules() +
  tm_polygons() +
  tm_credits(giscoR::gisco_attributions(),
    fontface = "italic",
    size = 0.7
  ) +
  tm_layout(
    main.title = "NUTS1: Displacing Canary Islands to the right",
    main.title.size = 0.9,
    main.title.fontface = "bold",
    attr.outside = TRUE
  )

NUTS1_orig <- esp_get_nuts(nuts_level = 1, moveCAN = FALSE)

tm_shape(NUTS1_orig) +
  tm_graticules() +
  tm_polygons() +
  tm_credits(giscoR::gisco_attributions(),
    fontface = "italic",
    size = 0.7
  ) +
  tm_layout(
    main.title = "NUTS1: Canary Islands on the true location",
    main.title.size = 0.9,
    main.title.fontface = "bold",
    attr.outside = TRUE
  )


AndOriental <-
  esp_get_nuts(region = c("Almeria", "Granada", "Jaen", "Malaga"))

qtm(AndOriental, main.title = "Andalucia Oriental")



RandomRegions <- esp_get_nuts(region = c("ES1", "ES300", "ES51"))
qtm(RandomRegions, main.title = "Random regions")

MixingCodes <- esp_get_nuts(region = c("ES4", "ES-PV", "Valencia"))
qtm(MixingCodes, main.title = "Mixing codes")
```

---

esp_get_prov *Get Provinces of Spain as* sf *polygons and points*

Description

Returns provinces of Spain as polygons and points at a specified scale.

- `esp_get_prov()` uses GISCO (Eurostat) as source. Please use `giscoR::gisco_attributions()`
- `esp_get_prov_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_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**.
`...`	Arguments passed on to `esp_get_nuts` `spatialtype` Type of geometry to be returned: • `"LB"`: Labels - point object. • `"RG"`: Regions - polygon object.
`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`. See **About caching**.
`update_cache`	A logical whether to update cache. Default is `FALSE`. When set to `TRUE` it would force a fresh download of the source file.
`cache_dir`	A path to a cache directory. See **About caching**.
`verbose`	Logical, displays 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. See **Displacing the Canary Islands**.
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" (see esp_codelist).

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 sf object specified by spatialtype.

About caching

You can set your cache_dir with esp_set_cache_dir().

Sometimes cached files may be corrupt. On that case, try re-downloading the data setting update_cache = TRUE.

If you experience any problem on download, try to download the corresponding .geojson file by any other method and save it on your cache_dir. Use the option verbose = TRUE for debugging the API query.

Displacing the Canary Islands

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

Source

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

See Also

Other political: esp_codelist, esp_get_can_box(), esp_get_capimun(), esp_get_ccaa(), esp_get_country(), esp_get_gridmap, esp_get_munic(), esp_get_nuts()

Examples

```
prov <- esp_get_prov()

library(tmap)

qtm(prov)

# Random Provinces

Random <-
  esp_get_prov(prov = c(
    "Zamora",
    "Palencia",
    "ES-GR",
    "ES521",
    "01"
  ))

tm_shape(Random) +
  tm_polygons(col = "codauto", legend.show = FALSE, palette = "Spectral")



# All Provinces of a Zone plus an addition

Mix <- esp_get_prov(prov = c(
  "Noroeste",
  "Castilla y Leon", "La Rioja"
))

qtm(Mix)

# ISO codes available

allprovs <- esp_get_prov()


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

---

esp_get_railway *Get* sf *lines and points with the railways of Spain*

Description

Loads a sf lines or point object representing the nodes and railway lines of Spain.

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. See **About caching**.
update_cache	A logical whether to update cache. Default is FALSE. When set to TRUE it would force a fresh download of the source file.
cache_dir	A path to a cache directory. See **About caching**.
verbose	Logical, displays information. Useful for debugging, default is FALSE.
spatialtype	Spatial type of the output. Use "line" for extracting the railway as lines and "point" for extracting stations.

Value

A sf line or point object.

About caching

You can set your cache_dir with esp_set_cache_dir().

Sometimes cached files may be corrupt. On that case, try re-downloading the data setting update_cache = TRUE.

If you experience any problem on download, try to download the corresponding .geojson file by any other method and save it on your cache_dir. Use the option verbose = TRUE for debugging the API query.

Source

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

See Also

Other infrastructure: 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 = "grey99", border.col = "grey50") +
  tm_shape(ccaa) +
  tm_borders("black") +
  tm_shape(rails) +
  tm_lines("tipo",
    legend.col.show = FALSE, lwd = 3,
    palette = "viridis"
  ) +
  tm_shape(stations) +
  tm_symbols("red", size = .3, alpha = 0.5, shape = 19)
```

---

esp_get_rivers *Get* sf *polygon and lines of rivers, channels and other wetlands of Spain*

Description

Loads a sf polygon or line object representing rivers, channels, reservoirs and other wetlands 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. See **About caching**.
update_cache	A logical whether to update cache. Default is FALSE. When set to TRUE it would force a fresh download of the source file.
cache_dir	A path to a cache directory. See **About caching**.
verbose	Logical, displays 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 lines.
name	Optional. A character or regex expression with the name of the element(s) to be extracted.

Details

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

Value

A sf polygon or line object.

Source

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

See Also

Other natural: esp_get_hydrobasin(), esp_get_hypsobath()

Examples

```
# Use of regex

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


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

# See the diference

# Rivers in Spain
```

```
shapeEsp <- esp_get_country(moveCAN = FALSE)

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


library(tmap)

tm_shape(shapeEsp, bbox = MainRivers) +
  tm_borders() +
  tm_shape(MainRivers) +
  tm_lines(col = "skyblue", lwd = 3)


# Wetlands in South-West Andalucia
and <- esp_get_prov(c("Huelva", "Sevilla", "Cadiz"))
Wetlands <- esp_get_rivers(spatialtype = "area")


tm_shape(and) +
  tm_polygons() +
  tm_shape(Wetlands) +
  tm_polygons(col = "skyblue", alpha = 0.5, border.col = "skyblue", lwd = 2)
```

`esp_get_roads` *Get* `sf` *lines of the roads of Spain*

Description

Loads a `sf` line object representing the main 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. See **About caching**.
update_cache	A logical whether to update cache. Default is FALSE. When set to TRUE it would force a fresh download of the source file.
cache_dir	A path to a cache directory. See **About caching**.
verbose	Logical, displays 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. See **Displacing the Canary Islands**.

Details

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

Value

A sf line object.

About caching

You can set your cache_dir with esp_set_cache_dir().

Sometimes cached files may be corrupt. On that case, try re-downloading the data setting update_cache = TRUE.

If you experience any problem on download, try to download the corresponding .geojson file by any other method and save it on your cache_dir. Use the option verbose = TRUE for debugging the API query.

Displacing the Canary Islands

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

Source

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

See Also

Other infrastructure: esp_get_railway()

Examples

```
#'
country <- esp_get_country()
Roads <- esp_get_roads()


library(tmap)

tm_shape(country) +
  tm_fill(col = "grey90") +
  tm_shape(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 (2019)*

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 with 8,131 rows and fields:

- **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. This is a combination of **cpro** and **cmun**, aligned with INE coding scheme.
- **geometry**: geometry field.

Source

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

See Also

esp_get_munic().

Other datasets: esp_codelist, esp_nuts.sf, leaflet.providersESP.df, pobmun19

Other municipalities: esp_get_capimun(), esp_get_munic()

Examples

```
data("esp_munic.sf")

teruel_cpro <- esp_dict_region_code("Teruel", destination = "cpro")

teruel_sf <- esp_munic.sf[esp_munic.sf$cpro == teruel_cpro, ]
teruel_city <- teruel_sf[teruel_sf$name == "Teruel", ]

# Plot

library(tmap)

tm_shape(teruel_sf) +
  tm_polygons("#FDFBEA") +
  tm_shape(teruel_city) +
  tm_fill(
    col = "name",
    palette = "#C12838",
    labels = "City of Teruel",
    title = ""
  ) +
  tm_graticules(lines = FALSE) +
  tm_layout(
    main.title = "Municipalities of Teruel",
    legend.position = c("left", "top")
  ) +
  tm_scale_bar() +
  tm_compass(
    type = "rose",
    size = 3,
    position = c("left", "bottom")
  )
```

---

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

Other datasets: esp_codelist, esp_munic.sf, leaflet.providersESP.df, pobmun19

Other nuts: esp_get_nuts()

Examples

```
data("esp_nuts.sf")

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)

# Plot Urban Type: See
# https://ec.europa.eu/eurostat/web/rural-development/methodology
tm_shape(nuts3) +
  tm_polygons(
    "URBN_TYPE",
    style = "cat",
    border.col = "black",
    border.alpha = 0.3,
    title = "Urban topology",
    labels = c("Urban", "Intermediate", "Rural"),
```

```
  palette = c("grey80", "#FFC183", "#68AC20")
) +
tm_graticules(lines = FALSE) +
tm_layout(
  main.title = "NUTS3 levels of Spain",
  legend.position = c("left", "center"),
  legend.title.size = 0.8
)
```

`esp_set_cache_dir` *Set your* **mapSpain** *cache dir*

Description

This function will store your `cache_dir` path on your local machine and would load it for future sessions. Type `Sys.getenv("MAPSPAIN_CACHE_DIR")` to find your cached path.

Alternatively, you can store the `cache_dir` manually with the following options:

- Run `Sys.setenv(MAPSPAIN_CACHE_DIR = "cache_dir")`. You would need to run this command on each session (Similar to `install = FALSE`).
- Set `options(mapSpain_cache_dir = "cache_dir")`. Similar to the previous option. This is **not recommended any more**, and it is provided for backwards compatibility purposes.
- Write this line on your .Renviron file: `MAPSPAIN_CACHE_DIR = "value_for_cache_dir"` (same behavior than `install = TRUE`). This would store your `cache_dir` permanently.

Usage

```
esp_set_cache_dir(
  cache_dir,
  overwrite = FALSE,
  install = FALSE,
  verbose = TRUE
)
```

Arguments

`cache_dir`	A path to a cache directory. On missing value the function would store the cached files on a temporary dir (See `base::tempdir()`).
`overwrite`	If this is set to `TRUE`, it will overwrite an existing `MAPSPAIN_CACHE_DIR` that you already have in local machine.
`install`	if `TRUE`, will install the key in your local machine for use in future sessions. Defaults to `FALSE`. If `cache_dir` is `FALSE` this parameter is set to `FALSE` automatically.
`verbose`	Logical, displays information. Useful for debugging, default is `FALSE`.

Value

An (invisible) character with the path to your `cache_dir`.

See Also

`rappdirs::user_config_dir()`

Other cache utilities: `esp_clear_cache()`

Examples

```
# Don't run this! It would modify your current state
## Not run:
esp_set_cache_dir(verbose = TRUE)

## End(Not run)

Sys.getenv("MAPSPAIN_CACHE_DIR")
```

`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` on `esp_getTiles()` are:

provider
'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'
'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'
'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

Other datasets: esp_codelist, esp_munic.sf, esp_nuts.sf, pobmun19

Other imagery utilities: addProviderEspTiles(), esp_getTiles()

Examples

```
data("leaflet.providersESP.df")

library(tibble)

as_tibble(leaflet.providersESP.df)
```

---

pobmun19 *Population by municipality (2019)*

Description

A data frame with 8,131 rows containing the population data by municipality in Spain (2019).

Source

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

See Also

Other datasets: esp_codelist, esp_munic.sf, esp_nuts.sf, leaflet.providersESP.df

Examples

```
data("pobmun19")

library(tibble)

as_tibble(pobmun19)
```

Index