Package ‘mapSpain’

July 12, 2023

Type   Package
Title   Administrative Boundaries of Spain
Version 0.8.0

Description Administrative Boundaries of Spain at several levels
(Autonomous Communities, 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), rappdirs (>=
0.3.0), sf (>= 0.9.0), utils
Suggests ggplot2 (>= 3.0.0), knitr, leaflet (>= 2.0.0), png (>=
0.1-5), rmarkdown, slippymath (>= 0.3.1), terra (>= 1.1-4),
testthat (>= 3.0.0), tidyterra

VignetteBuilder knitr

Conif/Needs/coverage covr

Config/Needs/website ragg, reactable, maturalearth, tidyverse,
ropenspain/rostemplate, ropensci/maturalearthhires,
IamKDO/GADMTools

Config/testthat/edition 3

Config/testthat/parallel true

Copyright © EuroGeographics for the administrative boundaries. Atlas
Nacional de España (ANE) CC BY 4.0 <http://www.ign.es>. INE

Encoding UTF-8

1
**LazyData** true

**RoxygenNote** 7.2.3

**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)

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

**Repository** CRAN

**Date/Publication** 2023-07-12 18:30:08 UTC

---

**R topics documented:**

addProviderEspTiles .................................................. 3
esp_check_access ....................................................... 4
esp_clear_cache ......................................................... 5
esp_codelist ............................................................. 6
esp_dict_region_code ................................................... 7
esp_getTiles ............................................................. 9
esp_get_can_box ......................................................... 12
esp_get_capimun ........................................................ 15
esp_get_ccaa ............................................................. 18
esp_get_comarca ........................................................ 21
esp_get_country ........................................................ 23
esp_get_gridmap ........................................................ 25
esp_get_grid_BDN ......................................................... 27
esp_get_grid_EEA ........................................................ 29
esp_get_grid_ESDAC ...................................................... 30
esp_get_grid_MTN ........................................................ 32
esp_get_hydrobasin ...................................................... 34
esp_get_hypsobath ....................................................... 36
esp_get_munic ............................................................. 39
esp_get_nuts ............................................................. 42
esp_get_prov .............................................................. 45
esp_get_railway .......................................................... 49
esp_get_rivers ............................................................ 51
esp_get_roads ............................................................. 53
esp_get_simpl_prov ...................................................... 55
esp_make_provider ....................................................... 57
esp_munic.sf ............................................................. 58
esp_nuts.sf ............................................................... 59
esp_set_cache_dir ....................................................... 61
Description

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

Usage

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

```r
providerEspTileOptions(...)
```

Arguments

- **map**: A map widget created from `leaflet::leaflet()`.
- **provider**: Name of the provider, see esp_tiles_providers 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()`.

Details

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

Value

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

Source

See Also

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

Other imagery utilities: `esp_getTiles()`, `esp_make_provider()`, `esp_tiles_providers`

Examples

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

PuertadelSol
```

---

### esp_check_access

**Check access to SIANE data**

**Description**

Check if R has access to resources at [https://github.com/rOpenSpain/mapSpain/tree/sianedata](https://github.com/rOpenSpain/mapSpain/tree/sianedata).

**Usage**

```r
esp_check_access()
```

**Value**

A logical.

**See Also**

- `giscoR::gisco_check_access()`

**Examples**

```r
esp_check_access()
```
**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**

```r
esp_clear_cache(config = FALSE, 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 if 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**

```r
# 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  

**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": Cataln
- "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 Estadística: https://www.ine.es/
- ISO: https://www.iso.org/home.html

See Also

Other datasets: esp_munic.sf, esp_nuts.sf, esp_tiles_providers, pobmum19
Other political: esp_get_can_box(), esp_get_capimun(), esp_get_ccaa(), esp_get_comarca(), esp_get_country(), esp_get_gridmap, esp_get_munic(), esp_get_nuts(), esp_get_prov(), esp_get_simpl_prov()
Other dictionary: esp_dict_region_code()

Examples

data(“esp_codelist”)
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 IS02 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)

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 v1.3.3.

Usage
esp_getTiles(
  x,
  type = "IDErionda",
  zoom = NULL,
  zoommin = 0,
  crop = TRUE,
  res = 512,
  bbox_expand = 0.05,
  transparent = TRUE,
  mask = FALSE,
  update_cache = FALSE,
  cache_dir = NULL,
  verbose = FALSE,
  options = NULL
)

Arguments
x An sf or sfc object.
type This parameter could be either:
  - The name of one of the pre-defined providers (see esp_tiles_providers()).
  - A list with two named elements id and q with your own parameters. See esp_make_provider() and examples.
esp_getTiles

- **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.

- **options**: A named list containing additional options to pass to the query.

**Details**

Zoom levels are described on the OpenStreetMap wiki:

<table>
<thead>
<tr>
<th>Zoom level</th>
<th>Area to represent</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>whole world</td>
</tr>
<tr>
<td>3</td>
<td>large country</td>
</tr>
<tr>
<td>5</td>
<td>state</td>
</tr>
<tr>
<td>8</td>
<td>county</td>
</tr>
<tr>
<td>10</td>
<td>metropolitan area</td>
</tr>
<tr>
<td>11</td>
<td>city</td>
</tr>
<tr>
<td>13</td>
<td>village or suburb</td>
</tr>
<tr>
<td>16</td>
<td>streets</td>
</tr>
<tr>
<td>18</td>
<td>some buildings, trees</td>
</tr>
</tbody>
</table>

For a complete list of providers see `esp_tiles_providers`.

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

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

**Value**

A SpatRaster is returned, with 3 (RGB) or 4 (RGBA) layers, depending on the provider. See `terra::rast()`.
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, v1.3.3.

See Also

terra::rast().
Other imagery utilities: addProviderEspTiles(), esp_make_provider(), esp_tiles_providers

Examples

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

segovia <- esp_get_prov_siane("segovia", epsg = 3857)
tile <- esp_getTiles(segovia, "IGNBase.Todo")

library(ggplot2)
library(tidyterra)

ggplot(segovia) +
  geom_spatraster_rgb(data = tile) +
  geom_sf(fill = NA)

# Another provider

tile2 <- esp_getTiles(segovia, type = "MDT")

ggplot(segovia) +
  geom_spatraster_rgb(data = tile2) +
  geom_sf(fill = NA)

# A custom WMS provided

custom_wms <- esp_make_provider(
  id = "an_id_for_caching",
  q = "https://idecyl.jcyl.es/geoserver/ge/wms?",
  service = "WMS",
  version = "1.3.0",
  format = "image/png",
  layers = "geolog_cyl_litologia"
)
\begin{verbatim}

custom_wms_tile <- esp_getTiles(segovia, custom_wms)
autoplot(custom_wms_tile) +
  geom_sf(data = segovia, fill = NA, color = "red")

# A custom WMTS provider

custom_wmts <- esp_make_provider(
  id = "cyl_wmts",
  q = "https://www.ign.es/wmts/pnoa-ma?",
  service = "WMTS",
  layer = "01.OrthoimageCoverage"
)
custom_wmts_tile <- esp_getTiles(segovia, custom_wmts)
autoplot(custom_wmts_tile) +
  geom_sf(data = segovia, fill = NA, color = "white", linewidth = 2)

# Example from https://leaflet-extras.github.io/leaflet-providers/preview/
stamen_water <- list(
  id = "Stamen_Water",
  q = "https://stamen-tiles-b.a.ssl.fastly.net/watercolor/{z}/{x}/{y}.jpg"
)
stamen <- esp_getTiles(segovia, stamen_water, zoommin = 1)
autoplot(stamen) +
  geom_sf(data = segovia, fill = NA, color = "white", linewidth = 1)

## End(Not run)
\end{verbatim}

---

**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 provide 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_comarca(), esp_get_country(), esp_get_gridmap, esp_get_munic(), esp_get_nuts(), esp_get_prov(), esp_get_simpl_prov()

Examples

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

# Plot
library(ggplot2)
ggplot(Provs) +  
  geom_sf() +  
  geom_sf(data = Box) +  
  geom_sf(data = Line) +  
  theme_linedraw()

# Displacing Canary
# By same factor

displace <- c(15, 0)

Provs_D <- esp_get_prov(moveCAN = displace)

Box_D <- esp_get_can_box(style = "left", moveCAN = displace)

Line_D <- esp_get_can_provinces(moveCAN = displace)

ggplot(Provs_D) +  
  geom_sf() +  
  geom_sf(data = Box_D) +  
  geom_sf(data = Line_D) +  
  theme_linedraw()

# 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

ggplot(Countries) +
geom_sf(fill = "#DFDFDF") +
geom_sf(data = CANbox, fill = "#C7E7FB", linewidth = 1) +
geom_sf(data = CCAA, fill = "#FDFBEA") +
coord_sf(
  xlim = c(-10, 4.3),
  ylim = c(34.6, 44)
) +
theme(
  panel.background = element_rect(fill = "#C7E7FB"),
  panel.grid = element_blank()
)

---

**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**

```r
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
esp_get_capimun

- "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.

**Value**
A sf point object.

**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.

**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_comarca(), esp_get_country(), esp_get_gridmap, esp_get_munic(), esp_get_nuts(), esp_get_prov(), esp_get_simplProv()

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, "IGNBase.Todo", zoommin = 2)

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

# Check on plot
library(ggplot2)
library(tidyterra)
ggplot(points) +
geom_spatraster_rgb(data = tile) +
geom_sf(data = area, fill = NA, color = "blue") +
geom_sf(data = points, aes(fill = type), size = 5, shape = 21) +
scale_fill_manual(values = c("green", "red")) +
theme_void() +
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](http://www.ign.es/web/ign/portal). Years available are 2005 up to today.

**Usage**

```r
esp_get_ccaa(ccaa = NULL, moveCAN = TRUE, ...)
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**.
- `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**.
- `...` 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()`
esp_get_ccaa

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".

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_comarca()`, `esp_get_country()`, `esp_get_gridmap`, `esp_get_munic()`, `esp_get_nuts()`, `esp_get_prov()`, `esp_get_simpl_prov()`

Examples

ccaa <- esp_get_ccaa()

library(ggplot2)

ggplot(ccaa) +
  geom_sf()

  # Random CCAA
Random <- esp_get_ccaa(ccaa = c(
    "Euskadi",
    "Catalunya",
    "ES-EX",
    "Canarias",
    "ESS2",
    "01"
  ))

ggplot(Random) +
  geom_sf(aes(fill = codauto), show.legend = FALSE) +
  geom_sf_label(aes(label = codauto), alpha = 0.3)

  # All CCAA of a Zone plus an addition
Mix <-
  esp_get_ccaa(ccaa = c("La Rioja", "Noroeste"))

ggplot(Mix) +
  geom_sf()

  # Combine with giscoR to get countries

library(giscoR)
library(sf)

res <- 20  # Set same resolution

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)
ggplot(europe) +
  geom_sf(fill = "#DFDFDF", color = "#656565") +
  geom_sf(data = ccaa, fill = "#DFBEA", color = "#656565") +
  coord_sf(
    xlim = c(23, 74) * 10e4,
    ylim = c(14, 55) * 10e4
  ) +
  theme(panel.background = element_rect(fill = "#C7E7FB"))

---

**esp_get_comarca**

*Get 'comarcas' of Spain as sf polygons*

**Description**

Returns 'comarcas' of Spain as polygons, as provided by the **INE** (Instituto Nacional de Estadística).

**Usage**

```r
esp_get_comarca(
  region = NULL,
  comarca = NULL,
  moveCAN = TRUE,
  epsg = "4258",
  update_cache = FALSE,
  cache_dir = NULL,
  verbose = FALSE
)
```

**Arguments**

- **region**: A vector of names and/or codes for provinces or NULL to get all the comarcas. See Details.
- **comarca**: A name or **regex** expression with the names of the required comarcas. 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.
- **epsg**: projection of the map: 4-digit **EPSG code**. One of:
  - "4258": ETRS89
  - "4326": WGS84
  - "3855": ETRS89 / ETRS-LAEA
  - "3857": Pseudo-Mercator
- **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

'Comarcas' (English equivalent: district, county, area or zone) does not always have a formal legal status. They correspond mainly to natural areas (valleys, river basins etc.) or even to historical regions or ancient kingdoms.

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 comarcas of that level would be added.

Legal Notice:
The use of the information contained on the INE website may be carried out by users or re-use agents, at their own risk, and they will be the sole liable parties in the case of having to answer to third parties due to damages arising from such use.

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

INE: PC_Axis files.

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(), esp_get_prov(), esp_get_simpl_prov()

Examples

comarcas <- esp_get_comarca(moveCAN = FALSE)

library(ggplot2)

ggplot(comarcas) +
  geom_sf()
# Comarcas of Castille and Leon
comarcas_cyl <- esp_get_comarca("Castilla y Leon")

ggplot(comarcas_cyl) +
  geom_sf(aes(fill = ine.prov.name)) +
  labs(fill = "Province")

# Comarcas with Mountains or Alt(o,a) in the name
comarcas_alto <- esp_get_comarca(
  comarca = "Montaña|Monte|Sierra|Alt",
  epsg = 3857
)

ggplot(comarcas_alto) +
  geom_sf(aes(fill = ine.ccaa.name)) +
  geom_sf_text(aes(label = name), check_overlap = TRUE) +
  labs(fill = "CCAA")

---

## 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

```r
esp_get_country(moveCAN = TRUE, ...)
```

### Arguments

- **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](#).

- **...**
  Arguments passed on to `esp_get_nuts`

- **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.
resolution Resolution of the geospatial data. One of
- "60": 1:60million
- "20": 1:20million
- "10": 1:10million
- "03": 1:3million
- "01": 1:1million

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_comarca(), esp_get_gridmap, esp_get_munic(), esp_get_nuts(), esp_get_prov(), esp_get_simplProv()

Examples

OriginalCan <- esp_get_country(moveCAN = FALSE)

# One row only
nrow(OriginalCan)
library(ggplot2)
ggplot(OriginalCan) +
  geom_sf(fill = "grey70")

# Less resolution

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

library(ggplot2)

ggplot(MovedCan) +
  geom_sf(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.

See Also

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

Examples

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

library(ggplot2)

ggplot(hexccaa) +
  geom_sf(data = esp) +
  geom_sf(aes(fill = codauto),
          alpha = 0.3,
          show.legend = FALSE
     ) +
  geom_sf_text(aes(label = label), check_overlap = TRUE) +
  theme_void() +
  labs(title = "Hexbin: CCAA")

hexprov <- esp_get_hex_prov()

ggplot(hexprov) +
  geom_sf(data = esp) +
  geom_sf(aes(fill = codauto),
          alpha = 0.3,
          show.legend = FALSE
     ) +
  geom_sf_text(aes(label = label), check_overlap = TRUE) +
  theme_void() +
  labs(title = "Hexbin: Provinces")

gridccaa <- esp_get_grid_ccaa()

ggplot(gridccaa) +
  geom_sf(data = esp) +
  geom_sf(aes(fill = codauto),
          alpha = 0.3,
          show.legend = FALSE
     ) +
  geom_sf_text(aes(label = label), check_overlap = TRUE) +
```

theme_void() +
labs(title = "Grid: CCAA")

gridprov <- esp_get_grid_prov()

ggplot(gridprov) +
  geom_sf(data = esp) +
  geom_sf(aes(fill = codauto),
          alpha = 0.3,
          show.legend = FALSE
  ) +
  geom_sf_text(aes(label = label), check_overlap = TRUE) +
  theme_void() +
  labs(title = "Grid: Provinces")

---

**esp_get_grid_BDN**

*Get sf polygons of the national geographic grids provided by BDN*

**Description**

Loads a *sf* polygon with the geographic grids of Spain as provided on the Banco de Datos de la Naturaleza (Nature Data Bank), by the Ministry of Environment (MITECO):

- `esp_get_grid_BDN()` extracts country-wide grids with resolutions 5x5 or 10x10 kms.
- `esp_get_grid_BDN_ccaa()` extracts grids by Autonomous Community with resolution 1x1 km.

**Usage**

```r
esp_get_grid_BDN(
  resolution = 10,
  type = "main",
  update_cache = FALSE,
  cache_dir = NULL,
  verbose = FALSE
)

esp_get_grid_BDN_ccaa(
  ccaa,
  update_cache = FALSE,
  cache_dir = NULL,
  verbose = FALSE
)
```
### Arguments

- **resolution**
  Resolution of the grid in kms. Could be 5 or 10.

- **type**
  The scope of the grid. It could be mainland Spain ("main") or the Canary Islands ("canary").

- **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.

- **ccaa**
  A vector of names and/or codes for autonomous communities. See **Details** on `esp_get_ccaa()`.

### 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.

### Source

BDN data via a custom CDN (see [https://github.com/rOpenSpain/mapSpain/tree/sianedata/MTN](https://github.com/rOpenSpain/mapSpain/tree/sianedata/MTN)).


### See Also

- `esp_get_ccaa()`

Other grids: `esp_get_grid_EEA()`, `esp_get_grid_ESDAC()`, `esp_get_grid_MTN()`

### Examples

```r
grid <- esp_get_grid_BDN(resolution = "10", type = "main")

library(ggplot2)

ggplot(grid) +
  geom_sf() +
  theme_light()
```
**Description**

Loads a **sf** polygon with the geographic grids of Spain as provided by the European Environment Agency (EEA).

**Usage**

```r
esp_get_grid_EEA(
  resolution = 100,
  type = "main",
  update_cache = FALSE,
  cache_dir = NULL,
  verbose = FALSE
)
```

**Arguments**

- **resolution**: Resolution of the grid in kms. Could be 1, 10 or 100.
- **type**: The scope of the grid. It could be mainland Spain ("main") or the Canary Islands ("canary").
- **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.

**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.
esp_get_grid_EEA

Get sf polygons of the national geographic grids provided by ESDAC

Description

Loads a sf polygon with the geographic grids of Spain as provided by the European Soil Data Centre (ESDAC).

Usage

```r
esp_get_grid_EEA(
  resolution = 10,
  update_cache = FALSE,
  cache_dir = NULL,
  verbose = FALSE
)
```

Source

EEA reference grid.

See Also

Other grids: esp_get_grid_BDN(), esp_get_grid_ESDAC(), esp_get_grid_MTN()

Examples

```r
## Not run:

grid <- esp_get_grid_EEA(type = "main", resolution = 100)
grid_can <- esp_get_grid_EEA(type = "canary", resolution = 100)
esp <- esp_get_country(moveCAN = FALSE)

library(ggplot2)

ggplot(grid) +
  geom_sf() +
  geom_sf(data = grid_can) +
  geom_sf(data = esp, fill = NA) +
  theme_light() +
  labs(title = "EEA Grid for Spain")

## End(Not run)
```
Arguments

resolution: Resolution of the grid in kms. Could be 1 or 10.
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.

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.

Source

EEA reference grid.

References

• European Soil Data Centre (ESDAC), esdac.jrc.ec.europa.eu, European Commission, Joint Research Centre.

See Also

Other grids: esp_get_grid_BDN(), esp_get_grid_EEA(), esp_get_grid_MTN()
theme_light() +
labs(title = "ESDAC Grid for Spain")

## End(Not run)

---

**esp_get_grid_MTN**  
*Get sf polygons of the national geographic grids provided by IGN*

**Description**

 Loads a sf polygon with the geographic grids of Spain.

**Usage**

```r
esp_get_grid_MTN(
  grid = "MTN25_ETRS89_Peninsula_Baleares_Canarias",
  update_cache = FALSE,
  cache_dir = NULL,
  verbose = FALSE
)
```

**Arguments**

- **grid**  
  Name of the grid to be loaded. See Details.

- **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**

Metadata available on [https://github.com/rOpenSpain/mapSpain/tree/sianedata/MTN](https://github.com/rOpenSpain/mapSpain/tree/sianedata/MTN).

Possible values of grid are:

- **grid_name**
  - MTN25_ED50_Peninsula_Baleares
  - MTN25_ETRS89_ceuta_melilla_alboran
  - MTN25_ETRS89_Peninsula_Baleares_Canarias
  - MTN25_RegCan95_Canarias
  - MTN50_ED50_Peninsula_Baleares
  - MTN50_ETRS89_Peninsula_Baleares_Canarias
  - MTN50_RegCan95_Canarias

**MTN Grids:**
A description of the MTN (Mapa Topografico Nacional) grids available:

**MTN25_ED50_Peninsula_Baleares**
MTN25 grid corresponding to the Peninsula and Balearic Islands, in ED50 and geographical coordinates (longitude, latitude) This is the real MTN25 grid, that is, the one that divides the current printed series of the map, taking into account special sheets and irregularities.

**MTN50_ED50_Peninsula_Baleares**
MTN50 grid corresponding to the Peninsula and Balearic Islands, in ED50 and geographical coordinates (longitude, latitude) This is the real MTN50 grid, that is, the one that divides the current printed series of the map, taking into account special sheets and irregularities.

**MTN25_ETRS89_ceuta_melilla_alboran**
MTN25 grid corresponding to Ceuta, Melilla, Alboran and Spanish territories in North Africa, adjusted to the new official geodetic reference system ETRS89, in geographical coordinates (longitude, latitude).

**MTN25_ETRS89_Peninsula_Baleares_Canarias**
MTN25 real grid corresponding to the Peninsula, the Balearic Islands and the Canary Islands, adjusted to the new ETRS89 official reference geodetic system, in geographical coordinates (longitude, latitude).

**MTN50_ETRS89_Peninsula_Baleares_Canarias**
MTN50 real grid corresponding to the Peninsula, the Balearic Islands and the Canary Islands, adjusted to the new ETRS89 official reference geodetic system, in geographical coordinates (longitude, latitude).

**MTN25_RegCan95_Canarias**
MTN25 grid corresponding to the Canary Islands, in REGCAN95 (WGS84 compatible) and geographic coordinates (longitude, latitude). It is the real MTN25 grid, that is, the one that divides the current printed series of the map, taking into account the special distribution of the Canary Islands sheets.

**MTN50_RegCan95_Canarias**
MTN50 grid corresponding to the Canary Islands, in REGCAN95 (WGS84 compatible) and geographic coordinates (longitude, latitude). This is the real grid of the MTN50, that is, the one that divides the current printed series of the map, taking into account the special distribution of the Canary Islands sheets.

**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.
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

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

See Also
Other grids: esp_get_grid_BDN(), esp_get_grid_EEA(), esp_get_grid_ESDAC()

Examples
```
grid <- esp_get_grid_MTN(grid = "MTN50_ETRS89_Peninsula_Baleares_Canarias")
library(ggplot2)

ggplot(grid) +
  geom_sf() +
  theme_light() +
  labs(title = "MTN50 Grid for Spain")
```
esp_get_hydrobasin

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

```r
# 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

```r
hydroland <- esp_get_hydrobasin(domain = "land")
ydrolandsea <- esp_get_hydrobasin(domain = "landsea")

library(ggplot2)

ggplot(hydroland) +
  geom_sf(data = hydrolandsea, fill = "skyblue4", alpha = .4) +
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
```r
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.
esp_get_hypsobath

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(ggplot2)

hypsobath <- esp_get_hypsobath()

# Error on the data provided - There is an empty shape
# Remove:
hypsobath <- hypsobath[!sf::st_is_empty(hypsobath), ]

# Tints from Wikipedia
bath_tints <- colorRampPalette(
  rev(
    c(}
```r
hypsobath <- colorRampPalette(
  rev(
    c(
      "#D8F2FE", "#C6ECFF", "#B9E3FF",
      "#ACDBFB", "#A1D2F7", "#96C9F0",
      "#8DC1EA", "#84B9E3", "#79B2DE",
      "#71ABD8"
    )
  )
)

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

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

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

# Plot Canary Islands
ggplot(hypsobath) +
  geom_sf(aes(fill = as.factor(val_inf)),
          color = NA) +
  coord_sf(
    xlim = c(-18.6, -13),
    ylim = c(27, 29.5)
  ) +
  scale_fill_manual(values = pal) +
  guides(fill = guide_legend(
    title = "Elevation",
    direction = "horizontal",
    label.position = "bottom",
    title.position = "top",
    nrow = 1
  )) +
  theme(legend.position = "bottom")

# Plot Mainland
ggplot(hypsobath) +
  geom_sf(aes(fill = as.factor(val_inf)),
          color = NA)
```
```r
get_municipalities() +
coord_sf(
  xlim = c(-9.5, 4.4),
  ylim = c(35.8, 44)
) +
scale_fill_manual(values = pal) +
guides(fill = guide_legend(
  title = "Elevation",
  reverse = TRUE,
  keyheight = .8
))
```

---

**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](http://www.ign.es/web/ign/portal). Years available are 2005 up to today.

**Usage**

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

```r
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

Arguments

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 municipali-
  ties. 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 municipali-
  ties 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_comarca(), esp_get_country(), esp_get_gridmap, esp_get_nuts(), esp_get_prov(), esp_get_simpl_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")

data("pobmun19")

# Arrange and create breaks
Base_pop <- merge(Base, pobmun19,
by = c("cpro", "cmun"),
all.x = TRUE
)
esp_get_nuts

Get NUTS of Spain as \texttt{sf} polygons and points

\textbf{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 on 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

\textbf{Usage}

```r
esp_get_nuts(
  year = "2016",
  epsg = "4258",
  cache = TRUE,
  update_cache = FALSE,
  cache_dir = NULL,
)```
esp_get_nuts

verbose = FALSE,
resolution = "01",
spatialtype = "RG",
region = NULL,
nuts_level = "all",
moveCAN = TRUE
)

Arguments

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.

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_comarca(), esp_get_country(), esp_get_gridmap, esp_get_munic(), esp_get_prov(), esp_get_simpl_prov()

Other nuts: esp_nuts.sf

Examples

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

library(ggplot2)

ggplot(NUTS1) +
geom_sf() +
  labs(
    title = "NUTS1: Displacing Canary Islands",
    caption = giscoR::gisco_attributions()
  )

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

ggplot(NUTS1_alt) +
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

```r
esp_get_prov(prov = NULL, moveCAN = TRUE, ...)

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.
- **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.
- ... 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".
- **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](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_comarca()`, `esp_get_country()`, `esp_get_gridmap`, `esp_get_munic()`, `esp_get_nuts()`, `esp_get_simpl_prov()`

Examples

```r
prov <- esp_get_prov()
library(ggplot2)

ggplot(prov) +
  geom_sf() +
  theme_void()
```
# Random Provinces
Random <- esp_get_prov(prov = c("Zamora", "Palencia", "ES-GR", "ES521", "01"))

ggplot(Random) + geom_sf(aes(fill = codauto), show.legend = FALSE, alpha = 0.5) + scale_fill_manual(values = hcl.colors(nrow(Random), "Spectral") + theme_minimal()

# All Provinces of a Zone plus an addition
Mix <- esp_get_prov(prov = c("Noroeste", "Castilla y Leon", "La Rioja"))
Mix$CCAA <- esp_dict_region_code(Mix$codauto, origin = "codauto")

ggplot(Mix) + geom_sf(aes(fill = CCAA), alpha = 0.5) + scale_fill_discrete(type = hcl.colors(5, "Temps")) + theme_classic()

# ISO codes available
allprovs <- esp_get_prov()

ggplot(allprovs) + geom_sf(fill = NA) + geom_sf_text(aes(label = iso2.prov.code), check_overlap = TRUE, fontface = "bold") + theme_void()
Description

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

Usage

```r
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](https://github.com/rOpenSpain/mapSpain/tree/sianedata)).

**See Also**

Other infrastructure: `esp_get_roads()`

**Examples**

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

# Railways
rails <- esp_get_railway()

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

# Map
library(ggplot2)

ggplot(provs) +
  geom_sf(fill = "grey99", color = "grey50") +
  geom_sf(data = ccaa, fill = NA) +
  geom_sf(
    data = rails, aes(color = tipo),
    show.legend = FALSE, linewidth = 1.5
  ) +
  geom_sf(
    data = stations,
    color = "red", alpha = 0.5
  ) +
  coord_sf(
    xlim = c(-7.5, -2.5),
    ylim = c(38, 41)
  ) +
  scale_color_manual(values = hcl.colors(
    length(unique(rails$tipo)), "viridis") +
  theme_minimal()
```
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**

```r
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

```r
# 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")

sf::st_bbox(MainRivers)
library(ggplot2)

ggplot(shapeEsp) +
  geom_sf() +
  geom_sf(data = MainRivers, color = "skyblue", linewidth = 2) +
  coord_sf(
    xlim = c(-7.5, 1),
    ylim = c(36.8, 43)
  ) +
  theme_void()

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

ggplot(and) +
  geom_sf() +
  geom_sf(
esp_get_roads

```r
data = Wetlands, fill = "skyblue",
       color = "skyblue", alpha = 0.5
) +
coord_sf(
  xlim = c(-7.5, -4.5),
  ylim = c(36, 38.5)
) +
theme_void()
```

---

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

**Description**

Loads a *sf* line object representing the main roads of Spain.

**Usage**

```r
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](https://github.com/rOpenSpain/mapSpain/tree/sianedata)).

See Also

Other infrastructure: `esp_get_railway()`

Examples

country <- esp_get_country()
Roads <- esp_get_roads()

library(ggplot2)

ggplot(country) +
  geom_sf(fill = "grey90") +
  geom_sf(data = Roads, aes(color = tipo), show.legend = "line") +
  scale_color_manual(
    values = c("#003399", "#003399", "#ff0000", "#ffff00")
  ) +
  guides(color = guide_legend(direction = "vertical")) +
  theme_minimal() +
Get a simplified map of provinces and autonomous communities of Spain

Description

Loads a simplified map (sf object) with the boundaries of the provinces or autonomous communities of Spain, as provided by the INE (Instituto Nacional de Estadística).

Usage

```r
esp_get_simpl_prov(
  prov = NULL,
  update_cache = FALSE,
  cache_dir = NULL,
  verbose = FALSE
)
```

```r
esp_get_simpl_ccaa(
  ccaa = NULL,
  update_cache = FALSE,
  cache_dir = NULL,
  verbose = FALSE
)
```

Arguments

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

Details

Results are provided without CRS, as provided on source.

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_simpl_prov("Andalucia")) all the corresponding units of that level are provided (in this case, all the provinces of Andalucia).
esp_get_simpl_prov

### 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

INE: PC_Axis files

### See Also

- `esp_get_hex_prov()`, `esp_get_hex_ccaa()`
- Other political: `esp_codelist`, `esp_get_can_box()`, `esp_get_capimun()`, `esp_get_ccaa()`, `esp_get_comarca()`, `esp_get_country()`, `esp_get_gridmap`, `esp_get_munic()`, `esp_get_nuts()`, `esp_get_prov()`

### Examples

```r
prov_simp <- esp_get_simpl_prov()
library(ggplot2)

ggplot(prov_simp) +
  geom_sf(aes(fill = ine.ccaa.name)) +
  labs(fill = "CCAA")

# Provs of Single CCAA

and_simple <- esp_get_simpl_prov("Andalucia")

ggplot(and_simple) +
  geom_sf()

# CCAAs

ccaa_simp <- esp_get_simpl_ccaa()

ggplot(ccaa_simp) +
  geom_sf() +
  geom_sf_text(aes(label = ine.ccaa.name), check_overlap = TRUE)
```
Create a custom tile provider

Description
Helper function for `esp_getTiles()` that helps to create a custom provider.

Usage
```r
esp_make_provider(id, q, service, layers, ...)
```

Arguments
- `id`: An identifier for the user. Would be used also for identifying cached tiles.
- `q`: The base url of the service
- `service`: The type of tile service, either "WMS" or "WMTS".
- `layers`: The name of the layer to retrieve
- `...`: Additional parameters to the query, like `version`, `format`, `crs/srs`, `style`, ... depending on the capabilities of the service.

Details
This function is meant to work with services provided as of the OGC Standard.

Note that:
- `mapSpain` would not provide advice on the parameter `q` to be provided.
- Currently, on WMTS requests only services with `tilematrixset=GoogleMapsCompatible` are supported.

Value
A named list with two elements `id` and `q`.

See Also
- `esp_getTiles()`.
- For a list of potential providers from Spain check IDEE Directory.
- Other imagery utilities: `addProviderEspTiles()`, `esp_getTiles()`, `esp_tiles_providers`
Examples

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

custom_wms <- esp_make_provider(
  id = "an_id_for_caching",
  q = "https://idecyl.jcyl.es/geoserver/ge/wms?",
  service = "WMS",
  layers = "geolog_cyl_litologia"
)

x <- esp_get_ccaa("Castilla y León", epsg = 3857)

mytile <- esp_getTiles(x, type = custom_wms)

tidyterra::autoplot(mytile) +
  ggplot2::geom_sf(data = x, fill = NA)
## End(Not run)
```

---

**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**

See Also

esp_get_munic()

Other datasets: esp_codelist, esp_nuts.sf, esp_tiles_providers, 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(ggplot2)

ggplot(teruel_sf) +
  geom_sf(fill = "#FDFBEA") +
  geom_sf(data = teruel_city, aes(fill = name)) +
  scale_fill_manual(
    values = "#C12838",
    labels = "City of Teruel"
  ) +
  labs(
    fill = "",
    title = "Municipalities of Teruel"
  ) +
  theme_minimal() +
  theme(
    text = element_text(face = "bold"),
    panel.background = element_rect(colour = "black"),
    panel.grid = element_blank(),
    legend.position = c(.2, .95)
  )

---

**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**


**See Also**

Other datasets: esp_codelist, esp_munic.sf, esp_tiles_providers, 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)

# Plot Urban Type: See
# https://ec.europa.eu/eurostat/web/rural-development/methodology

library(ggplot2)
nuts3$URBN_TYPE_cat <- as.factor(nuts3$URBN_TYPE)
levels(nuts3$URBN_TYPE_cat)
levels(nuts3$URBN_TYPE_cat) <- c("Urban", "Intermediate", "Rural")
ggplot(nuts3) +
  geom_sf(aes(fill = URBN_TYPE_cat), linewidth = .1) +
  scale_fill_manual(values = c("grey80", "#FFC183", "#68AC20")) +
  labs(
    title = "NUTS3 levels of Spain",
    fill = "Urban topology"
  )
```
**esp_set_cache_dir**

```r
tHEME LINEDRAW() +
thEME(
    legEND.POsiTION = c(.8, .2)
)
```

---

**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 `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**

```r
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.
esp_tiles_providers

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(verbos = TRUE)
## End(Not run)
Sys.getenv("MAPSPAIN_CACHE_DIR")

---

**esp_tiles_providers**  
*List with information of Public WMS and WMTS of Spain*

Description

A named list of length 102 containing the parameters of the url information of different public WMS and WMTS providers of Spain.

Implementation of javascript plugin leaflet-providersESP v1.3.3.

Format

A named list of the providers available with the following structure:

- Each item of the list is named with the provider alias.
- Each element of the list contains two nested named lists:
  - `static` with the parameters to get static tiles plus an additional item named `attribution`.
  - `leaflet` with additional parameters to be passed onto `addProviderEspTiles()`.

Details

Providers available to be passed to type on `esp_getTiles()` are:

- "IDErioja"
- "IDErioja.Base"
- "IDErioja.Relieve"
- "IDErioja.Claro"
- "IDErioja.Oscuro"
- "IGNBase"
- "IGNBase.Todo"
- "IGNBase.Gris"
• "IGNBase.TodoNoFondo"
• "IGNBase.Orto"
• "MDT"
• "MDT.Elevaciones"
• "MDT.Relieve"
• "MDT.CurvasNivel"
• "MDT.SpotElevation"
• "PNOA"
• "PNOA.MaximaActualidad"
• "PNOA.Mosaico"
• "OcupacionSuelo"
• "OcupacionSuelo.Ocupacion"
• "OcupacionSuelo.Usos"
• "LiDAR"
• "MTN"
• "Geofisica"
• "Geofisica.Terremotos10dias"
• "Geofisica.Terremotos30dias"
• "Geofisica.Terremotos365dias"
• "Geofisica.ObservedEvents"
• "Geofisica.HazardArea"
• "VigilanciaVolcanica"
• "VigilanciaVolcanica.ErupcionesHistoricas"
• "CaminoDeSantiago"
• "CaminoDeSantiago.CaminoFrances"
• "CaminoDeSantiago.CaminosFrancia"
• "CaminoDeSantiago.CaminosGalicia"
• "CaminoDeSantiago.CaminosDelNorte"
• "CaminoDeSantiago.CaminosAndaluces"
• "CaminoDeSantiago.CaminosCentro"
• "CaminoDeSantiago.CaminosEste"
• "CaminoDeSantiago.CaminosCatalanes"
• "CaminoDeSantiago.CaminosSureste"
• "CaminoDeSantiago.CaminosInsulares"
• "CaminoDeSantiago.CaminosPortugueses"
• "Catastro"
• "Catastro.Catastro"
• "Catastro.Parcela"
• "Catastro.CadastralParcel"
• "Catastro.CadastralZoning"
• "Catastro.Address"
• "Catastro.Building"
• "Catastro.BuildingPart"
• "Catastro.AdministrativeBoundary"
• "Catastro.AdministrativeUnit"
• "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.CECAFTM"
• "ADIF"
• "ADIF.Vias"
• "ADIF.Nodos"
• "ADIF.Estaciones"
• "LimitesMaritimos"
• "LimitesMaritimos.LimitesMaritimos"
• "LimitesMaritimos.LineasBase"
• "Copernicus"
• "Copernicus.Forest"
• "Copernicus.ForestLeaf"
• "Copernicus.WaterWet"
• "Copernicus.SoilSeal"
• "Copernicus.GrassLand"
• "Copernicus.RiparianGreen"
• "Copernicus.RiparianLandCover"
• "Copernicus.Natura2k"
• "Copernicus.UrbanAtlas"
• "ParquesNaturales"
• "ParquesNaturales.Limites"
• "ParquesNaturales.ZonasPerifericas"

Source

https://dieghernan.github.io/leaflet-providersESP/ leaflet plugin, v1.3.3.

See Also

Other datasets: esp_codelist, esp_munic.sf, esp_nuts.sf, pobmun19
Other imagery utilities: addProviderEspTiles(), esp_getTiles(), esp_make_provider()

Examples

data("esp_tiles_providers")
# Get a single provider

single <- esp_tiles_providers["IGNBase.Todo"]
single$static

single$leaflet
Description

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

Source

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

See Also

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

Examples

data("pobmun19")
Index

* cache utilities
  esp_clear_cache, 5
  esp_set_cache_dir, 61

* datasets
  esp_codelist, 6
  esp_munic.sf, 58
  esp_nuts.sf, 59
  esp_tiles_providers, 62
  pobmun19, 66

* dictionary
  esp_codelist, 6
  esp_dict_region_code, 7

* grids
  esp_get_grid_BDN, 27
  esp_get_grid_EEA, 29
  esp_get_grid_ESDAC, 30
  esp_get_grid_MTN, 32

* helper
  esp_check_access, 4

* imagery utilities
  addProviderEspTiles, 3
  esp_getTiles, 9
  esp_make_provider, 57
  esp_tiles_providers, 62

* infrastructure
  esp_get_railway, 49
  esp_get_roads, 53

* municipalities
  esp_get_capimun, 15
  esp_get_munic, 39
  esp_munic.sf, 58

* natural
  esp_get_hydrobasin, 34
  esp_get_hypsobath, 36
  esp_get_rivers, 51

* nuts
  esp_get_nuts, 42
  esp_nuts.sf, 59

* political
  esp_codelist, 6
  esp_get_can_box, 12
  esp_get_capimun, 15
  esp_get_caa, 18
  esp_get_comarca, 21
  esp_get_country, 23
  esp_get_gridmap, 25
  esp_get_munic, 39
  esp_get_nuts, 42
  esp_get_prov, 45
  esp_get_simpl_prov, 55

addProviderEspTiles, 3, 11, 57, 65
addProviderEspTiles(), 13, 16, 19, 22, 24, 41, 44, 47, 54, 62
base::regex(), 41
base::tempdir(), 61

esp_check_access, 4
esp_clear_cache, 5, 62
esp_codelist, 6, 8, 13, 16, 17, 19, 20, 22, 24–26, 40, 41, 44, 47, 55, 56, 59, 60, 65, 66
esp_dict_region_code, 7, 7
esp_dict_region_code(), 8, 43, 44
esp_dict_translate
  (esp_dict_region_code), 7
esp_dict_translate(), 8
esp_get_can_box, 7, 12, 17, 20, 22, 24, 26, 41, 44, 47, 56
esp_get_can_box(), 12
esp_get_can_provinces
  (esp_get_can_box), 12
esp_get_can_provinces(), 12
esp_get_capimun, 7, 13, 15, 20, 22, 24, 26, 41, 44, 47, 56, 59
esp_get_caa, 7, 13, 17, 18, 22, 24, 26, 41, 44, 47, 56
esp_get_caa(), 18, 28
esp_get_ccaa_siane (esp_get_ccaa), 18
esp_get_ccaa_siane(), 18, 19
esp_get_comarca, 7, 13, 17, 20, 21, 24, 26, 41, 44, 47, 56
esp_get_country, 7, 13, 17, 20, 22, 23, 26, 41, 44, 47, 56
esp_get_grid_BDN, 27, 30, 31, 34
esp_get_grid_BDN(), 27
esp_get_grid_BDN_ccaa
    (esp_get_grid_BDN), 27
esp_get_grid_BDN_ccaa(), 27
esp_get_grid_ccaa (esp_get_gridmap), 25
esp_get_grid_EEA, 28, 29, 31, 34
esp_get_grid_ESDAC, 28, 30, 30, 34
esp_get_grid_MTN, 28, 30, 31, 32
esp_get_grid_prov (esp_get_gridmap), 25
esp_get_gridmap, 7, 13, 17, 20, 22, 24, 25, 41, 44, 47, 56
esp_get_hex_ccaa (esp_get_gridmap), 25
esp_get_hex_ccaa(), 56
esp_get_hex_prov (esp_get_gridmap), 25
esp_get_hex_prov(), 56
esp_get_hydrobasin, 34, 37, 52
esp_get_hypsobath, 35, 36, 52
esp_get_munic, 7, 13, 17, 20, 22, 24, 26, 39, 44, 47, 56
esp_get_munic(), 15, 39, 40, 59
esp_get_munic_siane (esp_get_munic), 39
esp_get_munic_siane(), 39, 40
esp_get_nuts, 7, 13, 17, 18, 20, 22–24, 26, 41, 42, 46, 47, 56, 60
esp_get_nuts(), 12, 18, 46
esp_get_prov, 7, 13, 17, 20, 22, 24, 26, 41, 44, 45, 56
esp_get_prov(), 45, 46
esp_get_prov_siane (esp_get_prov), 45
esp_get_prov_siane(), 46, 47
esp_get_railway, 49, 54
esp_get_rivers, 35, 37, 51
esp_get_roads, 50, 53
esp_get_simpl_ccaa
    (esp_get_simpl_prov), 55
esp_get_simpl_prov, 7, 13, 17, 20, 22, 24, 26, 41, 44, 47, 55
esp_getTiles, 4, 9, 57, 65
esp_getTiles(), 13, 16, 19, 22, 24, 41, 44, 47, 54, 57, 62
esp_make_provider, 4, 11, 57, 65
esp_make_provider(), 9
esp_munic.sf, 7, 17, 41, 58, 60, 65, 66
esp_nuts.sf, 7, 44, 59, 59, 65, 66
esp_set_cache_dir, 5, 61
esp_set_cache_dir(), 11, 16, 19, 22, 24, 28, 29, 31, 33, 35, 37, 41, 44, 47, 49, 54, 56
esp_tiles_providers, 3, 4, 7, 10, 11, 57, 59, 60, 62, 66
esp_tiles_providers(), 9
giscoR::gisco_attributions(), 18, 39, 44, 45
giscoR::gisco_check_access(), 4
giscoR::gisco_get_communes(), 40
giscoR::gisco_get_lau(), 40, 41
giscoR::gisco_get_nuts(), 44
leaflet::addTiles(), 3, 4
leaflet::leaflet(), 3, 4
leaflet::providerTileOptions(), 3, 4
leaflet::tileOptions(), 4
pobmun19, 7, 59, 60, 65, 66
providerEspTileOptions
    (addProviderEspTiles), 3
providerEspTileOptions(), 4
rappdirs::user_config_dir(), 62
regex, 16, 21, 40, 51
sf::st_transform(), 25
terra::rast(), 10, 11