Package ‘mapSpain’

October 13, 2022

Type Package
Title Administrative Boundaries of Spain
Version 0.6.2
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
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
Config/testthat/edition 3
Config/testthat/parallel true


Encoding UTF-8
LazyData true
RoxygenNote 7.2.1
X-schema.org-applicationCategory cartography
X-schema.org-isPartOf https://ropenspain.es/
**R topics documented:**

- addProviderEspTiles
- esp_check_access
- esp_clear_cache
- esp_codelist
- esp_dict_region_code
- esp_getTiles
- esp_get_can_box
- esp_get_capimun
- esp_get_ccaa
- esp_get_comarca
- esp_get_country
- esp_get_gridmap
- esp_get_grid_BDN
- esp_get_grid_EEA
- esp_get_grid_ESDAC
- esp_get_grid_MTN
- esp_get_hydrobasin
- esp_get_hypsobath
- esp_get_munic
- esp_get_nuts
- esp_getProv
- esp_get_railway
- esp_get_rivers
- esp_get_roads
- esp_get_simpl_prov
- esp_munic.sf
- esp_nuts.sf
- esp_set_cache_dir
- leaflet.providersESP.df
- pobmun19

**Index**

63
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().

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()`, `leaflet.providersESP.df`

Examples

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

---

**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()
```
**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")
```
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+.name**: 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.
- **cpbo**: 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.
esp_dict_region_code

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, leaflet.providersESP.df, pobmun19
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")

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

```r
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 v1.3.0.

### Usage

```r
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,
  options = NULL
)
```

### Arguments

- **x**: An sf or sfc 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.
**esp_getTiles**

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 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</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 leaflet.providersESP.df.

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


See Also

terra::rast().

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

Examples

```r
## 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(ggplot2)
library(tidyterra)

ggplot(Murcia) +
  geom_spatraster_rgb(data = Tile) +
  geom_sf(fill = NA)

## 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 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

```r
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_1.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

```r
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

```r
ggplot(Countries) +
  geom_sf(fill = "#DFDFDF") +
  geom_sf(data = CANbox, fill = "#C7E7FB", lwd = 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.

**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.
esp_get_capimun

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_simpl_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(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

labs(title = "Centroid vs. capimun")

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

#### Usage

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

```r
esp_get_ccaa_siane(
 ccxaa = 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.

- **year**
  
  Release year. See `esp_get_nuts()` for `esp_get_ccaa()` and Details for `esp_get_ccaa_siane()`.

- **spatialtype**
  
  Type of geometry to be returned:

  - "LB": Labels - point object.
  - "RG": Regions - polygon object.

### Examples

```r
## Not run

labs(title = "Centroid vs. capimun")

## End(Not run)
```
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 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)
ggplot(europe) +
  geom_sf(fill = "#DFDFDF", color = "#656565") +
  geom_sf(data = ccaa, fill = "#FDFBEA", 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 Estadistica).

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()
# 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**

`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_simpl_prov()

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:4287, use sf::st_transform() to change the projection.
**esp_get_gridmap**

**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
```
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
)
```

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

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).


See Also

esp_get_ccaa()

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

Examples

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

library(ggplot2)

ggplot(grid) + geom_sf() + theme_light() +
esp_get_grid_EEA

Get sf polygons of the national geographic grids provided by EEA

Description

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

Usage

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_ESDAC

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)
```

esp_get_grid_ESDAC

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

Examples
```r
# Not run:
grid <- esp_get_grid_ESDAC()
esp <- esp_get_country(moveCAN = FALSE)
library(ggplot2)

ggplot(grid) +
  geom_sf() +
  geom_sf(data = esp, color = "grey50", fill = NA) +
```
esp_get_grid_MTN

```r
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:**

- 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
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

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")

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
**esp_get_hydrobasin**

- "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**

```r
hydroland <- esp_get_hydrobasin(domain = "land")
hydrolandsea <- esp_get_hydrobasin(domain = "landsea")
library(ggplot2)

ggplot(hydroland) +
  geom_sf(data = hydrolandsea, fill = "skyblue4", alpha = .4) +
```
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 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.
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(
      "#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", "#D366A3", "#E81B66",
      "#FEBBC0", "#E14B5", "#D17AB", "#BDCC96",
      "#A8C68F", "#94BF8B", "#ACD0A5"
    )
  )
)

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
    ) +
  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,
```
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_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_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

```r
# 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
)
```
br <- sort(c(0, 50, 100, 200, 500, 1000, 5000, 50000, 100000, Inf))

Base_pop$cuts <- cut(Base_pop$pob19, br, dig.lab = 20)

library(ggplot2)

ggplot(Base_pop) +
  geom_sf(aes(fill = cuts), color = NA) +
  geom_sf(data = provs, fill = NA, color = "grey70") +
  scale_fill_manual(values = hcl.colors(length(br), "cividis")) +
  labs(
    title = "Population in Castilla y Leon",
    subtitle = "INE, 2019",
    fill = "Persons"
  ) +
  theme_void()

---

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",
movesCAN = TRUE
)

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 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.
movesCAN 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

```r
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) +
```

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

• esp_get_prov_siane() uses CartoBase ANE as source, provided by Instituto Geográfico 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_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_getProv_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).

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).

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, lwd = 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/](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 difference

# 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", lwd = 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).

See Also

Other infrastructure: `esp_get_railway()`

Examples

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

library(ggplot2)

```r
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()
```
esp_get_simpl_prov

```r
labs(color = "Road type") +
theme(legend.position = "bottom")
```

---

**esp_get_simpl_prov**  
*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_simplProv(
  prov = NULL,
  update_cache = FALSE,
  cache_dir = NULL,
  verbose = FALSE
)
esp_get_simpl_ccaa(
  ccaa = NULL,
  update_cache = FALSE,
  cache_dir = NULL,
  verbose = FALSE
)
```

**Arguments**

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>prov</td>
<td>A vector of names and/or codes for provinces or NULL to get all the provinces. See Details.</td>
</tr>
<tr>
<td>update_cache</td>
<td>A logical whether to update cache. Default is FALSE. When set to TRUE it would force a fresh download of the source file.</td>
</tr>
<tr>
<td>cache_dir</td>
<td>A path to a cache directory. See About caching.</td>
</tr>
<tr>
<td>verbose</td>
<td>Logical, displays information. Useful for debugging, default is FALSE.</td>
</tr>
<tr>
<td>ccaa</td>
<td>A vector of names and/or codes for autonomous communities or NULL to get all the autonomous communities. See Details.</td>
</tr>
</tbody>
</table>

**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).
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)
```
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, 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(ggplot2)

ggplot(teruel_sf) +
  geom_sf(fill = "#FDFBEA") +
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, leaflet.providersESP.df, pobmun19`
Other nuts: `esp_get_nuts()`
Examples

```r
data("esp_nuts.sf")
nuts <- esp_nuts.sf

# Select NUTS 3
code <- esp_nuts.sf[esp_nuts.sf$LEVEL_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)
levels(nuts$URBN_TYPE_cat) <- c("Urban", "Intermediate", "Rural")
ggplot(nuts) + geom_sf(aes(fill = URBN_TYPE_cat), lwd = .1) + scale_fill_manual(values = c("grey80", "#FFC183", "#68AC20")] + labs(
  title = "NUTS3 levels of Spain",
  fill = "Urban topology"
) + theme_linedraw() + theme(
  legend.position = c(.8, .2)
)
```

---

**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

```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`.

See Also

- `rappdirs::user_config_dir()`
- Other cache utilities: `esp_clear_cache()`

Examples

```r
# 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")
```

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 v1.3.0`. 
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:

```r
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.CaminosAndalucia'
'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'
'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.POIs'
'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


See Also

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

Examples

data("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")
Index

* cache utilities
  esp_clear_cache, 5
  esp_set_cache_dir, 58
* datasets
  esp_codelist, 6
  esp_munic.sf, 56
  esp_nuts.sf, 57
  leaflet.providersESP.df, 59
  pobmun19, 62
* dictionary
  esp_codelist, 6
  esp_dict_region_code, 7
* grids
  esp_get_grid_BDN, 26
  esp_get_grid_EEA, 28
  esp_get_grid_ESDAC, 29
  esp_get_grid_MTN, 31
* helper
  esp_check_access, 4
* imagery utilities
  addProviderEspTiles, 3
  esp_getTiles, 9
  leaflet.providersESP.df, 59
* infrastructure
  esp_get_railway, 48
  esp_get_roads, 52
* municipalities
  esp_get_capimun, 14
  esp_get_munic, 38
  esp_munic.sf, 56
* natural
  esp_get_hydrobasin, 33
  esp_get_hypsobath, 35
  esp_get_rivers, 50
* nuts
  esp_get_nuts, 41
  esp_nuts.sf, 57
* political
  esp_codelist, 6
  esp_get_can_box, 11
  esp_get_capimun, 14
  esp_get_ccaa, 17
  esp_get_comarca, 20
  esp_get_country, 22
  esp_get_gridmap, 24
  esp_get_munic, 38
  esp_get_nuts, 41
  esp_get_prov, 44
  esp_get_simpl_prov, 54
  addProviderEspTiles, 3, 11, 62
  addProviderEspTiles(), 12, 15, 18, 21, 23, 40, 43, 46, 53
  base::regex(), 40
  base::tempdir(), 59
  esp_check_access, 4
  esp_clear_cache, 5, 59
  esp_codelist, 6, 8, 12, 15, 16, 18, 19, 21, 23–25, 39, 40, 43, 46, 54–57, 62
  esp_dict_region_code, 7, 7
  esp_dict_region_code(), 8, 42, 43
  esp_dict_translate
    (esp_dict_region_code), 7
  esp_dict_translate(), 8
  esp_get_can_box, 7, 11, 16, 19, 21, 23, 25, 40, 43, 46, 55
  esp_get_can_box(), 11
  esp_get_can_provinces
    (esp_get_can_box), 11
  esp_get_can_provinces(), 11
  esp_get_capimun, 7, 12, 14, 19, 21, 23, 25, 40, 43, 46, 55, 56
  esp_get_ccaa, 7, 12, 16, 17, 21, 23, 25, 40, 43, 46, 55
  esp_get_ccaa(), 17, 27
  esp_get_ccaa_siane (esp_get_ccaa), 17
  esp_get_ccaa_siane(), 17, 18

63
esp_get_comarca, 7, 12, 16, 19, 20, 23, 25, 40, 43, 46, 55
esp_get_country, 7, 12, 16, 19, 21, 22, 25, 40, 43, 46, 55
esp_get_grid_BDN, 26, 29, 30, 33
esp_get_grid_BDN(), 26
esp_get_grid_BDN_ccaa
  (esp_get_grid_BDN), 26
esp_get_grid_BDN_ccaa(), 26
esp_get_grid_EEA, 27, 28, 30, 33
esp_get_grid_ESDAC, 27, 29, 29, 33
esp_get_grid_MTN, 27, 29, 30, 31
esp_get_grid_prov (esp_get_gridmap), 24
esp_get_gridmap, 7, 12, 16, 19, 21, 23, 24, 40, 43, 46, 55
esp_get_hex_ccaa (esp_get_gridmap), 24
esp_get_hex_ccaa(), 55
esp_get_hex_prov (esp_get_gridmap), 24
esp_get_hex_prov(), 55
esp_get_hydrobasin, 33, 36, 51
esp_get_hypsobath, 34, 35, 51
esp_get_munic, 7, 12, 16, 19, 21, 23, 25, 38, 43, 46, 55, 56
esp_get_munic(), 14, 38, 39, 56
esp_get_munic_siane (esp_get_munic), 38
esp_get_munic_siane(), 38, 39
esp_get_nuts, 7, 12, 16, 17, 19, 21–23, 25, 40, 41, 45, 46, 55, 57
esp_get_nuts(), 11, 17, 45
esp_get_prov, 7, 12, 16, 19, 21, 23, 25, 40, 43, 44, 55
esp_get_prov(), 44, 45
esp_get_prov_siane (esp_get_prov), 44
esp_get_prov_siane(), 45, 46
esp_get_railway, 48, 53
esp_get_rivers, 34, 36, 50
esp_get_roads, 49, 52
esp_get_simpl_ccaa
  (esp_get_simpl_prov), 54
esp_get_simpl_prov, 7, 12, 16, 19, 21, 23, 25, 40, 43, 46, 54
esp_getTiles, 4, 9, 62
esp_getTiles(), 12, 15, 18, 21, 23, 40, 43, 46, 53, 60
esp_munic.sf, 7, 16, 40, 56, 57, 62
esp_nuts.sf, 7, 43, 56, 57, 62
giscoR::gisco_check_access(), 4
giscoR::gisco_get_communes(), 39
giscoR::gisco_get_lau(), 39, 40
giscoR::gisco_get_nuts(), 43
leaflet::addTiles(), 3, 4
leaflet::leaflet(), 3, 4
leaflet::providerTileOptions(), 3, 4
leaflet::tileOptions(), 4
pobmun19, 7, 56, 57, 62, 62
providerEspTileOptions
  (addProviderEspTiles), 3
providerEspTileOptions(), 3
rappdirs::user_config_dir(), 59
regex, 15, 20, 39, 50
sf::st_transform(), 24
terra::rast(), 10, 11