Package 'mapSpain'

August 26, 2024

Type Package

Title Administrative Boundaries of Spain

Version 0.9.2

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

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 curl, 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/Needs/check curl

Config/Needs/coverage covr

Config/Needs/website ragg, reactable, rnaturalearth, tidyverse, devtools, remotes, ropenspain/rostemplate, ropensci/rnaturalearthhires, rspatial/geodata, rgeoboundaries, sessioninfo, sfheaders, rapidjsonr, jsonify, geometries

Config/testthat/edition 3

Config/testthat/parallel true

Encoding UTF-8

LazyData true

RoxygenNote 7.3.2

X-schema.org-applicationCategory cartography

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

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

NeedsCompilation no

Author Diego Hernangómez [aut, cre, cph] (<https://orcid.org/0000-0001-8457-4658>, rOpenSpain)

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

Repository CRAN

Date/Publication 2024-08-26 12:50:01 UTC

Contents

addProviderEspTiles . . . 3
esp_check_access . . . 4
esp_clear_cache . . . 5
esp_codelist . . . 6
esp_detect_cache_dir . . . 8
esp_dict_region_code . . . 9
esp_getTiles . . . 10
esp_get_can_box . . . 14
esp_get_capimun . . . 16
esp_get_ccaa . . . 19
esp_get_comarca . . . 22
esp_get_country . . . 25
esp_get_gridmap . . . 27
esp_get_grid_BDN . . . 29
esp_get_grid_EEA . . . 30
esp_get_grid_ESDAC . . . 32
esp_get_grid_MTN . . . 33
esp_get_hydrobasin . . . 36
esp_get_hypsobath . . . 37
esp_get_munic . . . 41
esp_get_nuts . . . 44
esp_get_prov . . . 47
esp_get_railway . . . 50
esp_get_rivers . . . 52
esp_get_roads . . . 54
esp_get_simpl_prov . . . 56
esp_make_provider . . . 58

esp_move_can 59
esp_munic.sf 61
esp_nuts.sf 62
esp_set_cache_dir 64
esp_tiles_providers 65
pobmun19 69

Index **71**

`addProviderEspTiles` *Include base tiles of Spanish public administrations on a* R*hrefhttps://CRAN.R-project.org/package=leaflet***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`	the map to add the tile layer to
`provider`	Name of the provider, see esp_tiles_providers for values available.
`layerId`	the layer id to assign
`group`	the name of the group the newly created layers should belong to (for `clearGroup` and `addLayersControl` purposes). Human-friendly group names are permitted–they need not be short, identifier-style names.
`options`	tile options
`...`	Arguments passed on to `leaflet::providerTileOptions()`.

Details

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

Value

A modified `leaflet::leaflet()` map object.

Source

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

See Also

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

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

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

Examples

```
library(leaflet)
leafmap <- leaflet(width = "100%", height = "60vh") %>%
  setView(lat = 40.4166, lng = -3.7038400, zoom = 10) %>%
  addTiles(group = "Default (OSM)") %>%
  addProviderEspTiles(
    provider = "IDErioja.Claro",
    group = "IDErioja Claro"
  ) %>%
  addProviderEspTiles(
    provider = "RedTransporte.Carreteras",
    group = "Carreteras"
  ) %>%
  addLayersControl(
    baseGroups = c("IDErioja Claro", "Default (OSM)"),
    overlayGroups = "Carreteras",
    options = layersControlOptions(collapsed = FALSE)
  )

leafmap
```

---

esp_check_access *Check access to SIANE data*

Description

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

Usage

```
esp_check_access()
```

Value

a logical.

See Also

giscoR::gisco_check_access()

Other helper: esp_move_can()

Examples

```
esp_check_access()
```

---

esp_clear_cache *Clear your* R*hrefhttps://CRAN.R-project.org/package=mapSpain***mapSpain** *cache dir*

Description

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

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

Usage

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

Arguments

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

Details

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

Value

Invisible. This function is called for its side effects.

See Also

Other cache utilities: esp_detect_cache_dir(), esp_set_cache_dir()

Examples

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

## End(Not run)

Sys.getenv("MAPSPAIN_CACHE_DIR")
```

`esp_codelist` *Database with codes and names of spanish regions*

Description

A `data.frame` object 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 and columns:

nuts1.code NUTS 1 code

nuts1.name NUTS 1 name

nuts1.name.alt NUTS 1 alternative name

nuts1.shortname.es NUTS1 1 short (common) name (Spanish)

codauto INE code of the autonomous community

iso2.ccaa.code ISO2 code of the autonomous community

nuts2.code NUTS 2 Code

ine.ccaa.name INE name of the autonomous community

iso2.ccaa.name.es ISO2 name of the autonomous community (Spanish)

iso2.ccaa.name.ca ISO2 name of the autonomous community (Catalan)

iso2.ccaa.name.gl ISO2 name of the autonomous community (Galician)

iso2.ccaa.name.eu ISO2 name of the autonomous community (Basque)

nuts2.name NUTS 2 name

cldr.ccaa.name.en CLDR name of the autonomous community (English)

cldr.ccaa.name.es CLDR name of the autonomous community (Spanish)

cldr.ccaa.name.ca CLDR name of the autonomous community (Catalan)

cldr.ccaa.name.ga CLDR name of the autonomous community (Galician)

cldr.ccaa.name.eu CLDR name of the autonomous community (Basque)

ccaa.shortname.en Short (common) name of the autonomous community (English)

ccaa.shortname.es Short (common) name of the autonomous community (Spanish)

ccaa.shortname.ca Short (common) name of the autonomous community (Catalan)

ccaa.shortname.ga Short (common) name of the autonomous community (Galician)

ccaa.shortname.eu Short (common) name of the autonomous community (Basque)

cpro INE code of the province

iso2.prov.code ISO2 code of the province

nuts.prov.code NUTS code of the province

ine.prov.name INE name of the province

iso2.prov.name.es ISO2 name of the province (Spanish)

iso2.prov.name.ca ISO2 name of the province (Catalan)

iso2.prov.name.ga ISO2 name of the province (Galician)

iso2.prov.name.eu ISO2 name of the province (Basque)

cldr.prov.name.en CLDR name of the province (English)

cldr.prov.name.es CLDR name of the province (Spanish)

cldr.prov.name.ca CLDR name of the province (Catalan)

cldr.prov.name.ga CLDR name of the province (Galician)

cldr.prov.name.eu CLDR name of the province (Basque)

prov.shortname.en Short (common) name of the province (English)

prov.shortname.es Short (common) name of the province (Spanish)

prov.shortname.ca Short (common) name of the province (Catalan)

prov.shortname.ga Short (common) name of the province (Galician)

prov.shortname.eu Short (common) name of the province (Basque)

nuts3.code NUTS 3 code

nuts3.name NUTS 3 name

nuts3.shortname.es NUTS 3 short (common) name

Note

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 autonomous communities with a single province (as Madrid, Asturias or Murcia) on this database.

Source

- **INE**: Instituto Nacional de Estadistica: https://www.ine.es/
- **Eurostat (NUTS)**: https://ec.europa.eu/eurostat/web/nuts/overview
- **ISO**: https://www.iso.org/home.html
- **CLDR**: https://unicode-org.github.io/cldr-staging/charts/38/index.html

See Also

Other datasets: esp_munic.sf, esp_nuts.sf, esp_tiles_providers, 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_detect_cache_dir *Detect cache dir for* R*hrefhttps://CRAN.R-project.org/package=mapSpain***mapSpain**

Description

Helper function to detect the current cache folder. See esp_set_cache_dir().

Usage

```
esp_detect_cache_dir(x = NULL)
```

Arguments

x	Ignored.

Value

A character with the path to your cache_dir.

See Also

Other cache utilities: esp_clear_cache(), esp_set_cache_dir()

Examples

```
esp_detect_cache_dir()
```

`esp_dict_region_code` *Convert and translate subdivision names*

Description

Converts long subdivision names into different coding schemes and languages.

Usage

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

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

Arguments

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

Details

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

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

Value

`esp_dict_region_code()` returns a vector of characters.

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

See Also

Other dictionary: `esp_codelist`

Other dictionary: `esp_codelist`

Examples

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

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

# From ISO2 to another codes

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

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

# Warning

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

## End(Not run)


vals <- c("La Rioja", "Sevilla", "Madrid", "Jaen", "Orense", "Baleares")

esp_dict_translate(vals)
esp_dict_translate(vals, lang = "es")
esp_dict_translate(vals, lang = "ca")
esp_dict_translate(vals, lang = "eu")
esp_dict_translate(vals, lang = "ga")

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

`esp_getTiles` *Get static tiles from public administrations of Spain*

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 = "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	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.
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:

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

For a complete list of providers see esp_tiles_providers.

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

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

Value

A `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

```
## 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, maxcell = Inf) +
  geom_sf(fill = NA)

# Another provider

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

ggplot(segovia) +
  geom_spatraster_rgb(data = tile2, maxcell = Inf) +
  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"
)

custom_wms_tile <- esp_getTiles(segovia, custom_wms)

autoplot(custom_wms_tile, maxcell = Inf) +
  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 = "OI.OrthoimageCoverage"
)

custom_wmts_tile <- esp_getTiles(segovia, custom_wmts)

autoplot(custom_wmts_tile, maxcell = Inf) +
  geom_sf(data = segovia, fill = NA, color = "white", linewidth = 2)
```

```
# Example from https://leaflet-extras.github.io/leaflet-providers/preview/
cartodb_voyager <- list(
  id = "CartoDB_Voyager",
  q = "https://a.basemaps.cartocdn.com/rastertiles/voyager/{z}/{x}/{y}.png"
)
cartodb <- esp_getTiles(segovia, cartodb_voyager, zoommin = 1)

autoplot(cartodb, maxcell = Inf) +
  geom_sf(data = segovia, fill = NA, color = "black", linewidth = 1)

## End(Not run)
```

`esp_get_can_box` *Get* `sf` *lines and polygons for insetting the Canary Islands*

Description

When plotting Spain, it is usual to represent the Canary Islands as an inset (see `moveCAN` on `esp_get_nuts()`). These functions provides complementary lines and polygons to be used when the Canary Islands are displayed as an inset.

- `esp_get_can_box()` is used to draw lines around the displaced Canary Islands.
- `esp_get_can_provinces()` is used to draw a separator line between the two provinces of the Canary Islands.

See also `esp_move_can()` to displace stand-alone objects on 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 LINESTRING 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. See also esp_move_can() for displacing stand-alone sf objects.

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()

Other Canary Islands: esp_move_can()

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 POINT *of the municipalities of Spain*

Description

Get a sf POINT with the location of the political powers for each municipality (possibly the center of the municipality).

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

Usage

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

Arguments

year	Release year. See **Details** for years available.
epsg	projection of the map: 4-digit EPSG code. One of: • "4258": ETRS89. • "4326": WGS84. • "3035": ETRS89 / ETRS-LAEA. • "3857": Pseudo-Mercator.
cache	A logical whether to do caching. Default is TRUE. See **About caching**.
update_cache	A logical whether to update cache. Default is FALSE. When set to TRUE it would force a fresh download of the source file.
cache_dir	A path to a cache directory. See **About caching**.
verbose	Logical, displays information. Useful for debugging, default is FALSE.
region	A vector of names and/or codes for provinces or NULL to get all the municipalities. See **Details**.
munic	A name or regex expression with the names of the required municipalities. NULL would return all municipalities.
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 resulting object as 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 higher level (province, CCAA 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. See also esp_move_can() for displacing stand-alone sf objects.

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, "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, maxcell = Inf) +
  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() +
  labs(title = "Centroid vs. capimun")

## End(Not run)
```

---

esp_get_ccaa *Get Autonomous Communities of Spain as* sf POLYGON *or* POINT

Description

Returns Autonomous Communities of Spain as sf POLYGON or POINT at a specified scale.

- esp_get_ccaa() uses GISCO (Eurostat) as source. Please use giscoR::gisco_attributions()
- esp_get_ccaa_siane() uses CartoBase ANE as source, provided by Instituto Geografico Nacional (IGN), http://www.ign.es/web/ign/portal. Years available are 2005 up to today.

Usage

```
esp_get_ccaa(ccaa = NULL, 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().
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 resulting object as 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. See also esp_move_can() for displacing stand-alone sf objects.

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",
  "ES52",
  "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 POLYGON

Description

Returns 'comarcas' of Spain as `sf POLYGON` objects, as provided by the **INE** (Instituto Nacional de Estadistica).

Usage

```
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 return all the possible comarcas.
`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.
`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 higher 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.

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 `esp_move_can()` for displacing stand-alone `sf` objects.

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* `sf` `POLYGON` *representing Spain*

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` Release year of the file. One of `"2003"`, `"2006"`, `"2010"`, `"2013"`, `"2016"` or `"2021"`. `epsg` projection of the map: 4-digit EPSG code. One of: • `"4258"`: ETRS89. • `"4326"`: WGS84. • `"3035"`: ETRS89 / ETRS-LAEA. • `"3857"`: Pseudo-Mercator. `cache` A logical whether to do caching. Default is `TRUE`. See **About caching**. `update_cache` A logical whether to update cache. Default is `FALSE`. When set to `TRUE` it would force a fresh download of the source file. `cache_dir` A path to a cache directory. See **About caching**. `verbose` Logical, displays information. Useful for debugging, default is `FALSE`. `resolution` Resolution of the geospatial data. One of

- "60": 1:60million
- "20": 1:20million
- "10": 1:10million
- "03": 1:3million
- "01": 1:1million

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 esp_move_can() for displacing stand-alone sf objects.

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 Andalusia).

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

```
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 POLYGON *with the national geographic grids from 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

```
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).

See original metadata and source on https://www.miteco.gob.es/es/biodiversidad/servicios/banco-datos-naturaleza/informacion-disponible/bdn-cart-aux-descargas-ccaa.html

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() +
  labs(title = "BDN Grid for Spain")
```

---

esp_get_grid_EEA *Get sf POLYGON of the national geographic grids from 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.

Source

EEA reference grid.

See Also

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

Examples

```
## 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` `POLYGON` *of the national geographic grids from ESDAC*

Description

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

Usage

```
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

- Panagos P., Van Liedekerke M., Jones A., Montanarella L., "European Soil Data Centre: Response to European policy support and public data requirements"; (2012) *Land Use Policy*, 29 (2), pp. 329-338. doi:10.1016/j.landusepol.2011.07.003
- 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

```
## 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) +
  theme_light() +
  labs(title = "ESDAC Grid for Spain")

## End(Not run)
```

---

esp_get_grid_MTN *Get* `sf` `POLYGON` *of the national geographic grids from IGN*

Description

Loads a `sf` `POLYGON` with the geographic grids of Spain.

Usage

```
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.

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.

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 *Get sf POLYGON of the drainage basin demarcations of Spain*

Description

Loads a sf POLYGON object containing areas with the required hydrographic elements of Spain.

Usage

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

Arguments

epsg	projection of the map: 4-digit EPSG code. One of: • "4258": ETRS89. • "4326": WGS84. • "3035": ETRS89 / ETRS-LAEA. • "3857": Pseudo-Mercator.
cache	A logical whether to do caching. Default is TRUE. See **About caching**.
update_cache	A logical whether to update cache. Default is FALSE. When set to TRUE it would force a fresh download of the source file.
cache_dir	A path to a cache directory. See **About caching**.
verbose	Logical, displays information. Useful for debugging, default is FALSE.
resolution	Resolution of the POLYGON. Values available are "3", "6.5" or "10".
domain	Possible values are "land", that includes only the ground part or the ground or "landsea", that includes both the ground and the related sea waters of the basin.

Details

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

Value

A sf POLYGON object.

About caching

You can set your cache_dir with esp_set_cache_dir().

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

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

Source

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

See Also

Other natural: esp_get_hypsobath(), esp_get_rivers()

Examples

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

library(ggplot2)


ggplot(hydroland) +
  geom_sf(data = hydrolandsea, fill = "skyblue4", alpha = .4) +
  geom_sf(fill = "skyblue", alpha = .5) +
  geom_sf_text(aes(label = rotulo),
    size = 3, check_overlap = TRUE,
    fontface = "bold",
    family = "serif"
  ) +
  coord_sf(
    xlim = c(-9.5, 4.5),
    ylim = c(35, 44)
  ) +
  theme_void()
```

---

esp_get_hypsobath *Get* sf POLYGON *or* LINESTRING *with hypsometry and bathymetry of Spain*

Description

Loads a sf POLYGON or LINESTRING object representing the hypsometry and bathymetry of Spain.

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

Usage

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

Arguments

epsg	projection of the map: 4-digit EPSG code. One of: • "4258": ETRS89. • "4326": WGS84. • "3035": ETRS89 / ETRS-LAEA. • "3857": Pseudo-Mercator.
cache	A logical whether to do caching. Default is TRUE. See **About caching**.
update_cache	A logical whether to update cache. Default is FALSE. When set to TRUE it would force a fresh download of the source file.
cache_dir	A path to a cache directory. See **About caching**.
verbose	Logical, displays information. Useful for debugging, default is FALSE.
resolution	Resolution of the shape. Values available are "3" or "6.5".
spatialtype	Spatial type of the output. Use "area" for POLYGON or "line" for LINESTRING.

Details

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

Value

A sf POLYGON or LINESTRING 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
# https://en.wikipedia.org/wiki/Wikipedia:WikiProject_Maps/Conventions/
# Topographic_maps

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

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

```
      "#A8C68F", "#94BF8B", "#ACD0A5"
    )
  )
)

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

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

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 POLYGON

Description

Returns municipalities of Spain `sf` POLYGON‘ at a specified scale.

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

Usage

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

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

Arguments

year	Release year. See **Details** for years available.
epsg	projection of the map: 4-digit EPSG code. One of: • `"4258"`: ETRS89. • `"4326"`: WGS84.

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

cache	A logical whether to do caching. Default is TRUE. See **About caching**.
update_cache	A logical whether to update cache. Default is FALSE. When set to TRUE it would force a fresh download of the source file.
cache_dir	A path to a cache directory. See **About caching**.
verbose	Logical, displays information. Useful for debugging, default is FALSE.
region	A vector of names and/or codes for provinces or NULL to get all the municipalities. See **Details**.
munic	A name or regex expression with the names of the required municipalities. NULL would return all municipalities.
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 resulting object as 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 higher 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. See also esp_move_can() for displacing stand-alone sf objects.

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")

# 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)

# Plot
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` `POLYGON` *or* `POINT`

Description

Returns NUTS regions of Spain as `POLYGON` or `POINT` 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 (CCAA).
- **"3"**: Roughly matches the provinces, but providing specific individual objects for each major island.

Usage

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

Arguments

`year`	Release year of the file. One of `"2003"`, `"2006"`, `"2010"`, `"2013"`, `"2016"` or `"2021"`.
`epsg`	projection of the map: 4-digit EPSG code. One of:

- "4258": ETRS89.
- "4326": WGS84.
- "3035": ETRS89 / ETRS-LAEA.
- "3857": Pseudo-Mercator.

cache	A logical whether to do caching. Default is TRUE. See **About caching**.
update_cache	A logical whether to update cache. Default is FALSE. When set to TRUE it would force a fresh download of the source file.
cache_dir	A path to a cache directory. See **About caching**.
verbose	Logical, displays information. Useful for debugging, default is FALSE.
resolution	Resolution of the geospatial data. One of • "60": 1:60million • "20": 1:20million • "10": 1:10million • "03": 1:3million • "01": 1:1million
spatialtype	Type of geometry to be returned: • "LB": Labels - POINT object. • "RG": Regions - POLYGON object.
region	Optional. A vector of region names, NUTS or ISO codes (see esp_dict_region_code()).
nuts_level	NUTS level. One of "0" (Country-level), "1", "2" or "3". See **Description**.
moveCAN	A logical TRUE/FALSE or a vector of coordinates c(lat, lon). It places the Canary Islands close to Spain's mainland. Initial position can be adjusted using the vector of coordinates. See **Displacing the Canary Islands**.

Value

A sf object specified by spatialtype.

About caching

You can set your cache_dir with esp_set_cache_dir().

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

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

Displacing the Canary Islands

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

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) +
  geom_sf() +
  labs(
    title = "NUTS1: Displacing Canary Islands",
    subtitle = "to the right",
    caption = giscoR::gisco_attributions()
  )


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

ggplot(NUTS1_orig) +
  geom_sf() +
  labs(
    title = "NUTS1",
    subtitle = "Canary Islands on the true location",
    caption = giscoR::gisco_attributions()
  )
```

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


ggplot(AndOriental) +
  geom_sf()



RandomRegions <- esp_get_nuts(region = c("ES1", "ES300", "ES51"))

ggplot(RandomRegions) +
  geom_sf() +
  labs(title = "Random Regions")


MixingCodes <- esp_get_nuts(region = c("ES4", "ES-PV", "Valencia"))


ggplot(MixingCodes) +
  geom_sf() +
  labs(title = "Mixing Codes")
```

---

esp_get_prov *Get Provinces of Spain as* sf POLYGON *or* POINT

Description

Returns provinces of Spain as POLYGON or POINT at a specified scale.

- esp_get_prov() uses GISCO (Eurostat) as source. Please use giscoR::gisco_attributions()
- esp_get_prov_siane() uses CartoBase ANE as source, provided by Instituto Geografico Nacional (IGN), http://www.ign.es/web/ign/portal. Years available are 2005 up to today.

Usage

```
esp_get_prov(prov = NULL, 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 resulting object as 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 higher 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. See also esp_move_can() for displacing stand-alone sf objects.

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

```
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()
```

---

esp_get_railway *Get* sf LINESTRING *or* POINT *with the railways of Spain*

Description

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

Usage

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

Arguments

year	Release year.
epsg	projection of the map: 4-digit EPSG code. One of: • "4258": ETRS89. • "4326": WGS84. • "3035": ETRS89 / ETRS-LAEA. • "3857": Pseudo-Mercator.
cache	A logical whether to do caching. Default is TRUE. See **About caching**.
update_cache	A logical whether to update cache. Default is FALSE. When set to TRUE it would force a fresh download of the source file.
cache_dir	A path to a cache directory. See **About caching**.
verbose	Logical, displays information. Useful for debugging, default is FALSE.
spatialtype	Spatial type of the output. Use "line" for extracting the railway as lines and "point" for extracting stations.

Value

A sf LINESTRING or POINT object.

About caching

You can set your cache_dir with esp_set_cache_dir().

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

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

Source

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

See Also

Other infrastructure: esp_get_roads()

Examples

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

# Railways
rails <- esp_get_railway()
```

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

# Map

library(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` *or* `LINESTRING` *of rivers, channels and other wetlands of Spain*

Description

Loads a `sf` `POLYGON` or `LINESTRING` object representing rivers, channels, reservoirs and other wetlands of Spain.

Usage

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

Arguments

epsg	projection of the map: 4-digit EPSG code. One of: • "4258": ETRS89. • "4326": WGS84. • "3035": ETRS89 / ETRS-LAEA. • "3857": Pseudo-Mercator.
cache	A logical whether to do caching. Default is TRUE. See **About caching**.
update_cache	A logical whether to update cache. Default is FALSE. When set to TRUE it would force a fresh download of the source file.
cache_dir	A path to a cache directory. See **About caching**.
verbose	Logical, displays information. Useful for debugging, default is FALSE.
resolution	Resolution of the POLYGON. Values available are "3", "6.5" or "10".
spatialtype	Spatial type of the output. Use "area" for POLYGON or "line" for LINESTRING.
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 LINESTRING object.

Source

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

See Also

Other natural: esp_get_hydrobasin(), esp_get_hypsobath()

Examples

```
# Use of regex

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


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

# See the diference

# Rivers in Spain
```

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

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

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(
    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 LINESTRING *of the roads of Spain*

Description

Loads a sf LINESTRING object representing the main roads of Spain.

Usage

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

```
  cache_dir = NULL,
  verbose = FALSE,
  moveCAN = TRUE
)
```

Arguments

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

Details

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

Value

A sf LINESTRING 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 esp_move_can() for displacing stand-alone sf objects.

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)

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() +
  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 Estadistica).

Usage

```
esp_get_simpl_prov(
  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

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 Andalusia).

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

```
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)
```

---

esp_make_provider *Create a custom tile provider*

Description

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

Usage

```
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

```
## 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",
  version = "1.3.0",
  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_move_can *Displace a `sf` object located in the Canary Islands*

Description

Helper function to displace an external `sf` object (potentially representing a location in the Canary Islands) to align it with the objects provided by `sf` with the option `moveCAN = TRUE`.

Usage

```
esp_move_can(x, moveCAN = TRUE)
```

Arguments

x	An sf object. It may be sf or sfc object.
moveCAN	A logical TRUE/FALSE or a vector of coordinates c(lat, lon).

Details

This is a helper function that intends to ease the representation of objects located in the Canary Islands that have been obtained from other sources rather than the package **mapSpain**.

Value

A sf object of the same class and same CRS than x but displaced accordingly.

See Also

Other helper: esp_check_access()

Other Canary Islands: esp_get_can_box()

Examples

```
library(sf)
teide <- data.frame(
  name = "Teide Peak",
  lon = -16.6437593,
  lat = 28.2722883
)

teide_sf <- st_as_sf(teide, coords = c("lon", "lat"), crs = 4326)

# If we use any mapSpain produced object with moveCAN = TRUE...

esp <- esp_get_country(moveCAN = c(13, 0))

library(ggplot2)


ggplot(esp) +
  geom_sf() +
  geom_sf(data = teide_sf, color = "red") +
  labs(
    title = "Canary Islands displaced",
    subtitle = "But not the external Teide object"
  )



# But we can
```

```
teide_sf_disp <- esp_move_can(teide_sf, moveCAN = c(13, 0))

ggplot(esp) +
  geom_sf() +
  geom_sf(data = teide_sf_disp, color = "red") +
  labs(
    title = "Canary Islands displaced",
    subtitle = "And also the external Teide object"
  )
```

---

esp_munic.sf *sf object with all the municipalities of Spain (2019)*

Description

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

Format

A sf object (resolution: 1:1 million, EPSG:4258) object with 8,131 rows and columns:

codauto INE code of the autonomous community.

ine.ccaa.name INE name of the autonomous community.

cpro INE code of the province.

ine.prov.name INE name of the province.

cmun INE code of the municipality.

name Name of the municipality.

LAU_CODE LAU Code (GISCO) of the municipality. This is a combination of **cpro** and **cmun** fields, aligned with INE coding scheme.

geometry geometry field.

Source

https://ec.europa.eu/eurostat/web/gisco/geodata/statistical-units/local-administrative-units, LAU 2019 data.

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"
  ) +
  guides(fill = guide_legend(position = "inside")) +
  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.inside = c(.2, .95)
  )
```

---

esp_nuts.sf *sf object with all the NUTS levels of Spain (2016)*

Description

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

Format

A sf object (resolution: 1:1million, EPSG:4258) with 86 rows and columns:

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

NUTS_ID NUTS identifier

URBN_TYPE Urban Type, see Details

CNTR_CODE Eurostat Country code ES

NAME_LATN NUTS name on Latin characters

NUTS_NAME NUTS name on local alphabet

MOUNT_TYPE Mount Type, see Details

COAST_TYPE Coast Type, see Details

FID FID

geometry geometry field

Details

MOUNT_TYPE: Mountain typology:

- 1: More than 50 % of the surface is covered by topographic mountain areas.
- 2: More than 50 % of the regional population lives in topographic mountain areas.
- 3: More than 50 % of the surface is covered by topographic mountain areas and where more than 50 % of the regional population lives in these mountain areas.
- 4: Non-mountain region / other regions.
- 0: No classification provided

URBN_TYPE: Urban-rural typology:

- 1: Predominantly urban region.
- 2: Intermediate region.
- 3: Predominantly rural region.
- 0: No classification provided

COAST_TYPE: Coastal typology:

- 1: Coastal (on coast).
- 2: Coastal (less than 50% of population living within 50 km. of the coastline).
- 3: Non-coastal region.
- 0: No classification provided

Source

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

See Also

Other datasets: esp_codelist, esp_munic.sf, 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"
  ) +
  theme_linedraw()
```

`esp_set_cache_dir` *Set your* R*hrefhttps://CRAN.R-project.org/package=mapSpain***mapSpain** *cache dir*

Description

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

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

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

Usage

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

Arguments

cache_dir	A path to a cache directory. On missing value the function would store the cached files on a temporary dir (See base::tempdir()).
overwrite	Logical. If this is set to TRUE, it will overwrite an existing MAPSPAIN_CACHE_DIR that you already have in local machine.
install	Logical. 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(), esp_detect_cache_dir()

Examples

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

## End(Not run)

Sys.getenv("MAPSPAIN_CACHE_DIR")
```

---

esp_tiles_providers *Database 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 WMTSproviders 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.CECAF1M"
- "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
```

---

pobmun19 *Database with the population of Spain by municipality (2019)*

Description

Database with the population of Spain by municipality (2019)

Format

An example `data.frame` object with 8,131 rows containing the population data by municipality in Spain (2019).

cpro INE code of the province.

provincia name of the province.

cmun INE code of the municipality.

name Name of the municipality.

pob19 Overall population (2019)

men Men population (2019)

women Women population (2019)

Source

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

See Also

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

Examples

```
data("pobmun19")
```

Index

∗ **Canary Islands**
esp_get_can_box, 14
esp_move_can, 59
∗ **cache utilities**
esp_clear_cache, 5
esp_detect_cache_dir, 8
esp_set_cache_dir, 64
∗ **datasets**
esp_codelist, 6
esp_munic.sf, 61
esp_nuts.sf, 62
esp_tiles_providers, 65
pobmun19, 69
∗ **dictionary**
esp_codelist, 6
esp_dict_region_code, 9
∗ **grids**
esp_get_grid_BDN, 29
esp_get_grid_EEA, 30
esp_get_grid_ESDAC, 32
esp_get_grid_MTN, 33
∗ **helper**
esp_check_access, 4
esp_move_can, 59
∗ **imagery utilities**
addProviderEspTiles, 3
esp_getTiles, 10
esp_make_provider, 58
esp_tiles_providers, 65
∗ **infrastructure**
esp_get_railway, 50
esp_get_roads, 54
∗ **municipalities**
esp_get_capimun, 16
esp_get_munic, 41
esp_munic.sf, 61
∗ **natural**
esp_get_hydrobasin, 36
esp_get_hypsobath, 37
esp_get_rivers, 52
∗ **nuts**
esp_get_nuts, 44
esp_nuts.sf, 62
∗ **political**
esp_codelist, 6
esp_get_can_box, 14
esp_get_capimun, 16
esp_get_ccaa, 19
esp_get_comarca, 22
esp_get_country, 25
esp_get_gridmap, 27
esp_get_munic, 41
esp_get_nuts, 44
esp_get_prov, 47
esp_get_simpl_prov, 56

addLayersControl, *3*
addProviderEspTiles, 3, *12*, *59*, *69*
addProviderEspTiles(), *15*, *18*, *21*, *24*, *26*, *43*, *45*, *49*, *55*, *66*

base::regex(), *43*
base::tempdir(), *65*

clearGroup, *3*

data.frame, *6*

esp_check_access, 4, *60*
esp_clear_cache, 5, *8*, *65*
esp_codelist, 6, *9*, *15*, *18*, *21*, *23, 24*, *26, 27*, *42, 43*, *46*, *48, 49*, *57*, *61*, *63*, *69*
esp_detect_cache_dir, *5*, 8, *65*
esp_dict_region_code, *8*, 9
esp_dict_region_code(), *9*, *45, 46*
esp_dict_translate *(*esp_dict_region_code*)*, 9
esp_dict_translate(), *9*
esp_get_can_box, *8*, 14, *18*, *21*, *24*, *26, 27*, *43*, *46*, *49*, *57*, *60*

esp_get_can_box(), *14*
esp_get_can_provinces *(*esp_get_can_box*)*, 14
esp_get_can_provinces(), *14*
esp_get_capimun, *8*, *15*, 16, *21*, *24*, *26, 27*, *43*, *46*, *49*, *57*, *61*
esp_get_ccaa, *8*, *15*, *18*, 19, *24*, *26, 27*, *43*, *46*, *49*, *57*
esp_get_ccaa(), *19, 20*, *29, 30*
esp_get_ccaa_siane *(*esp_get_ccaa*)*, 19
esp_get_ccaa_siane(), *19–21*
esp_get_comarca, *8*, *15*, *18*, *21*, 22, *26, 27*, *43*, *46*, *49*, *57*
esp_get_country, *8*, *15*, *18*, *21*, *24*, 25, *27*, *43*, *46*, *49*, *57*
esp_get_grid_BDN, 29, *31*, *33*, *35*
esp_get_grid_BDN(), *29*
esp_get_grid_BDN_ccaa *(*esp_get_grid_BDN*)*, 29
esp_get_grid_BDN_ccaa(), *29*
esp_get_grid_ccaa *(*esp_get_gridmap*)*, 27
esp_get_grid_EEA, *30*, 30, *33*, *35*
esp_get_grid_ESDAC, *30, 31*, 32, *35*
esp_get_grid_MTN, *30, 31*, *33*, 33
esp_get_grid_prov *(*esp_get_gridmap*)*, 27
esp_get_gridmap, *8*, *15*, *18*, *21*, *24*, *26*, 27, *43*, *46*, *49*, *57*
esp_get_hex_ccaa *(*esp_get_gridmap*)*, 27
esp_get_hex_ccaa(), *57*
esp_get_hex_prov *(*esp_get_gridmap*)*, 27
esp_get_hex_prov(), *57*
esp_get_hydrobasin, 36, *39*, *53*
esp_get_hypsobath, *37*, 37, *53*
esp_get_munic, *8*, *15*, *18*, *21*, *24*, *26, 27*, 41, *46*, *49*, *57*, *61*
esp_get_munic(), *17*, *41, 42*, *61*
esp_get_munic_siane *(*esp_get_munic*)*, 41
esp_get_munic_siane(), *41, 42*
esp_get_nuts, *8*, *15*, *18*, *20, 21*, *24–27*, *43*, 44, *48, 49*, *57*, *63*
esp_get_nuts(), *14*, *20*, *48*
esp_get_prov, *8*, *15*, *18*, *21*, *24*, *26, 27*, *43*, *46*, 47, *57*
esp_get_prov(), *47, 48*
esp_get_prov_siane *(*esp_get_prov*)*, 47
esp_get_prov_siane(), *47, 48*
esp_get_railway, 50, *56*
esp_get_rivers, *37*, *39*, 52
esp_get_roads, *51*, 54
esp_get_simpl_ccaa *(*esp_get_simpl_prov*)*, 56
esp_get_simpl_prov, *8*, *15*, *18*, *21*, *24*, *26, 27*, *43*, *46*, *49*, 56
esp_getTiles, *4*, 10, *59*, *69*
esp_getTiles(), *15*, *18*, *21*, *24*, *26*, *43*, *45*, *49*, *55*, *58, 59*, *66*
esp_make_provider, *4*, *12*, 58, *69*
esp_make_provider(), *11*
esp_move_can, *5*, *15*, 59
esp_move_can(), *14, 15*, *18*, *21*, *24*, *26*, *43*, *45*, *49*, *55*
esp_munic.sf, *8*, *18*, *43*, 61, *63*, *69*
esp_nuts.sf, *8*, *46*, *61*, 62, *69*
esp_set_cache_dir, *5*, *8*, 64
esp_set_cache_dir(), *8*, *12*, *18*, *21*, *24*, *26*, *30–32*, *35*, *37*, *39*, *42*, *45*, *49*, *51*, *55*, *57*
esp_tiles_providers, *3, 4*, *8*, *12*, *59*, *61*, *63*, 65, *69*
esp_tiles_providers(), *11*

giscoR::gisco_attributions(), *19*, *41*, *46, 47*
giscoR::gisco_check_access(), *5*
giscoR::gisco_get_communes(), *42*
giscoR::gisco_get_lau(), *42, 43*
giscoR::gisco_get_nuts(), *46*

leaflet::addTiles(), *4*
leaflet::leaflet(), *3, 4*
leaflet::providerTileOptions(), *3, 4*
leaflet::tileOptions(), *4*
list, *65*

pobmun19, *8*, *61*, *63*, *69*, 69
providerEspTileOptions *(*addProviderEspTiles*)*, 3
providerEspTileOptions(), *3*

rappdirs::user_config_dir(), *65*
regex, *17*, *23*, *42*, *53*

sf, *11*, *14–19*, *21, 22*, *24–27*, *29–33*, *35–38*, *41–45*, *47–57*, *59–62*
sf::st_transform(), *27*
sfc, *11*

terra::rast(), *12*