Package ‘mapSpain’

October 14, 2021

Type Package
Title Administrative Boundaries of Spain
Version 0.4.0
Description Administrative Boundaries of Spain at several levels (Autonomous Communities, Provinces, Municipalities) based on the 'GISCO' 'Eurostat' database <https://ec.europa.eu/eurostat/web/gisco> and 'CartoBase SIANE' from 'Instituto Geografico Nacional' <https://www.ign.es/>. It also provides a 'leaflet' plugin and the ability of downloading and processing static tiles.
License GPL-3
BugReports https://github.com/rOpenSpain/mapSpain/issues
Depends R (>= 3.6.0)
Imports countrycode (>= 1.2.0), giscoR (>= 0.2.4), rappdirs (>= 0.3.0), sf (>= 0.9.0), utils
Suggests ggplot2 (>= 3.0.0), ggspatial, knitr, leaflet (>= 2.0.0), png (>= 0.1-5), raster (>= 3.0-2), rmarkdown, slippymath (>= 0.3.1), terra (>= 1.1-4), testthat (>= 3.0.0)
VignetteBuilder knitr
Config/testthat/edition 3
Config/testthat/parallel true
Copyright © EuroGeographics for the administrative boundaries. Atlas Nacional de España (ANE) CC BY 4.0 <http://www.ign.es>
Encoding UTF-8
LazyData true
RoxygenNote 7.1.2
X-schema.org-applicationCategory cartography
X-schema.org-isPartOf https://ropenspain.es/
X-schema.org-keywords  rOpenSpain, tiles, r, maps, spatial, rstats, 
r-package, municipalities, Spain, gisco, provinces, ign, 
administrative-boundaries, ccaa, static-tiles

NeedsCompilation  no

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

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

Repository  CRAN

Date/Publication  2021-10-14 11:10:07 UTC

R topics documented:

mapSpain-package  .................................................. 3
addProviderEspTiles ............................................. 4
esp_check_access ................................................. 6
esp_clear_cache ................................................... 6
esp_codelist ......................................................... 7
esp_dict_region_code ............................................. 9
esp_getTiles ........................................................ 10
esp_get_can_box ................................................... 13
esp_get_capimun ................................................... 15
esp_get_ccaa ......................................................... 18
esp_get_country .................................................... 21
esp_get_gridmap .................................................... 23
esp_get_hydrobasin ............................................... 25
esp_get_hypsobath ............................................... 27
esp_get_munic ....................................................... 30
esp_get_nuts ......................................................... 33
esp_get_prov ........................................................ 37
esp_get_railway ..................................................... 40
esp_get_rivers ...................................................... 42
esp_get_roads ....................................................... 44
esp_munic.sf ....................................................... 46
esp_nuts.sf ........................................................ 47
esp_set_cache_dir .................................................. 49
layer_spatraster ................................................... 50
leaflet.providersESP.df .......................................... 51
pobmun19 .......................................................... 54

Index 55
Description

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

Details

<table>
<thead>
<tr>
<th>Package</th>
<th>mapSpain</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type</td>
<td>Package</td>
</tr>
<tr>
<td>Version</td>
<td>0.4.0</td>
</tr>
<tr>
<td>Date</td>
<td>2021</td>
</tr>
<tr>
<td>License</td>
<td>GPL-3</td>
</tr>
<tr>
<td>LazyLoad</td>
<td>yes</td>
</tr>
</tbody>
</table>

COPYRIGHT NOTICE (IGN)

This package uses data from CartoBase SIANE, provided by Instituto Geográfico Nacional (IGN):
Atlas Nacional de España (ANE) CC BY 4.0 ign.es

COPYRIGHT NOTICE (GISCO)

When data downloaded from GISCO is used in any printed or electronic publication, in addition to any other provisions applicable to the whole Eurostat website, data source will have to be acknowledged in the legend of the map and in the introductory page of the publication with the following copyright notice:

- EN: (C) EuroGeographics for the administrative boundaries
- FR: (C) EuroGeographics pour les limites administratives
- DE: (C) EuroGeographics bezüglich der Verwaltungsgrenzen

For publications in languages other than English, French or German, the translation of the copyright notice in the language of the publication shall be used.
If you intend to use the data commercially, please contact EuroGeographics for information regarding their license agreements.

Source

GISCO webpage
addProviderEspTiles

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

Description

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

Usage

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

providerEspTileOptions(...)

Arguments

map A map widget created from leaflet::leaflet().
provider Name of the provider, see leaflet.providersESP.df for values available.
layerId the layer id
group The name of the group the newly created layers should belong to Human-friendly
group names are permitted—they need not be short, identifier-style names. Any
number of layers and even different types of layers (e.g. markers and polygons)
can share the same group name. See leaflet::addTiles().
options a list of extra options for tile layers, popups, paths (circles, rectangles, polygons, ...
...), or other map elements
errorTileUrl the tile layer options; see https://leafletjs.com/reference-1.3.4.html#tilelayer
**addProviderEspTiles**

- **noWrap** the tile layer options; see [https://leafletjs.com/reference-1.3.4.html#tilelayer](https://leafletjs.com/reference-1.3.4.html#tilelayer)
- **opacity** the tile layer options; see [https://leafletjs.com/reference-1.3.4.html#tilelayer](https://leafletjs.com/reference-1.3.4.html#tilelayer)
- **zIndex** the tile layer options; see [https://leafletjs.com/reference-1.3.4.html#tilelayer](https://leafletjs.com/reference-1.3.4.html#tilelayer)
- **updateWhenIdle** the tile layer options; see [https://leafletjs.com/reference-1.3.4.html#tilelayer](https://leafletjs.com/reference-1.3.4.html#tilelayer)
- **detectRetina** the tile layer options; see [https://leafletjs.com/reference-1.3.4.html#tilelayer](https://leafletjs.com/reference-1.3.4.html#tilelayer)

**Details**

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

**Value**

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

**Source**


**See Also**

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

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

**Examples**

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

PuertadelSol
```
esp_check_access  Check access to SIANE data

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

Usage
esp_check_access()

Value
a logical.

See Also
giscoR::gisco_check_access()

Examples
esp_check_access()

.esp_clear_cache  Clear your mapSpain cache dir

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

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

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

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

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

Value

Invisible. This function is called for its side effects.

See Also

Other cache utilities: `esp_set_cache_dir()`

Examples

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

## End(Not run)
Sys.getenv("MAPSPAIN_CACHE_DIR")
```

---

<table>
<thead>
<tr>
<th>esp_code</th>
<th>Spanish Code Translation Data Frame</th>
</tr>
</thead>
</table>

Description

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

Format

A data frame with 59 rows codes as columns

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

• **cpro**: INE code of each province.

• **prov.shortname.+**: Short (common) name of each province. Several languages available.

**Note**

Languages available are:

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

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

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

**Source**

- **INE**: Instituto Nacional de Estadística: [https://www.ine.es/](https://www.ine.es/)
- **Eurostat (NUTS)**: [https://ec.europa.eu/eurostat/web/nuts/background](https://ec.europa.eu/eurostat/web/nuts/background)

**See Also**

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

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

Other dictionary: `esp_dict_region_code()`

**Examples**

```r
data("esp_codelist")```
**Description**

Converts long subdivision names into different coding schemes and languages.

**Usage**

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

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

**Arguments**

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

**Details**

If no match is found for any value, the function displays a warning and returns NA for those values. Note that mixing names of different administrative levels (e.g. "Catalonia" and "Barcelona") may return empty values, depending on the destination values.

**Value**

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

**See Also**

Other dictionary: `esp_codelist`
Other dictionary: `esp_codelist`
Examples

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

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

# From ISO2 to another codes

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

# Mixing levels
valsmix <- c("Centro", "Andalucia", "Seville", "Menorca")
esp_dict_region_code(valsmix, destination = "nuts")
## Not run:
# Warning
esp_dict_region_code(valsmix, destination = "codauto")
esp_dict_region_code(valsmix, destination = "iso2")
## End(Not run)

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

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

---

esp_getTiles

*Get static tiles from public administrations of Spanish.*
esp_getTiles

Description
Get static map tiles based on a spatial object. Maps can be fetched from various open map servers. This function is a implementation of the javascript plugin leaflet-providersESP v1.2.0.

Usage

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

Arguments

- `x` An sf or sfc object.
- `type` Name of the provider. See leaflet.providersESP.df.
- `zoom` Zoom level. If NULL, it is determined automatically. If set, it overrides zoommin. Only valid for WMTS tiles. On a single point it applies a buffer to the point and on zoom = NULL the function set a zoom level of 18. See Details.
- `zoommin` Delta on default zoom. The default value is designed to download fewer tiles than you probably want. Use 1 or 2 to increase the resolution.
- `crop` TRUE if results should be cropped to the specified x extent, FALSE otherwise. If x is an sf object with one POINT, crop is set to FALSE.
- `res` Resolution (in pixels) of the final tile. Only valid for WMS.
- `bbox_expand` A numeric value that indicates the expansion percentage of the bounding box of x.
- `transparent` Logical. Provides transparent background, if supported. Depends on the selected provider on type.
- `mask` TRUE if the result should be masked to x.
- `update_cache` A logical whether to update cache. Default is FALSE. When set to TRUE it would force a fresh download of the source file.
- `cache_dir` A path to a cache directory. See About caching.
- `verbose` Logical, displays information. Useful for debugging, default is FALSE.

Details
Zoom levels are described on the OpenStreetMap wiki:
For a complete list of providers see `leaflet.providersESP.df`.

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

\[
x <- \text{sf::st_transform}(x, 3857)
\]

**Value**

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

**About caching**

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

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

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

**Source**


**See Also**

`terra::rast()`.

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

**Examples**

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

Murcia <- esp_get_ccaa_siane("Murcia", epsg = 3857)
Tile <- esp_getTiles(Murcia)
```
library(ggplot2)

ggplot(Murcia) +
  layer_spatraster(Tile) +
  geom_sf(fill = NA)

## End(Not run)

### esp_get_can_box

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

**Description**

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

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

**Usage**

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

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

**Arguments**

- **style**
  
  Style of line around Canary Islands. Four options available: "left", "right", "box" or "poly".

- **moveCAN**

  A logical `TRUE/FALSE` or a vector of coordinates `c(lat,lon)`. It places the Canary Islands close to Spain’s mainland. Initial position can be adjusted using the vector of coordinates. See [Displacing the Canary Islands](#).

- **epsg**

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

**Value**

A sf polygon or line depending of `style` parameter.

`esp_get_can_provinces` returns a LINESTRING object.
Displacing the Canary Islands

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

Source

esp_get_can_provinces extracted from CartoBase ANE, se89_mult_admin_provcan_1.shp file.

See Also

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

Examples

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

# Plot
library(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 sf points of the municipalities of Spain

Description

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

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

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

Arguments

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

Details

- `year` could be passed as a single year ("YYYY" format, as end of year) or as a specific date ("YYYY-MM-DD" format). Historical information starts as of 2005.
- When using `region` you can use and mix names and NUTS codes (levels 1, 2 or 3), ISO codes (corresponding to level 2 or 3) or "cpro". See esp_codelist
- When calling a superior level (Province, Autonomous Community or NUTS1), all the municipalities of that level would be added.
Value

A sf point object.

About caching

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

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

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

Displacing the Canary Islands

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

Source

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

See Also

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

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

Examples

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

library(sf)

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

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

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

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

tile <- esp_getTiles(area, zoommin = 2)

# Join both point geometries

pontes <- rbind(
  centroid[, "type"],
  capimun[, "type"]
)

# Check on plot

library(ggplot2)

library(ggplot2)

library(ggplot2)

ggplot(points) +
  layer_spatraster(tile) +
  geom_sf(data = area, fill = NA, color = "blue") +
  geom_sf(data = points, aes(fill = type), size = 5, shape = 21) +
  scale_fill_manual(values = c("green", "red")) +
  theme_void() +
  labs(title = "Centroid vs. capimun")

## End(Not run)

### esp_get_ccaa

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

#### Description

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

- `esp_get_ccaa()` uses GISCO (Eurostat) as source. Please use `giscoR::gisco_attributions()`

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

#### Usage

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

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

```r
resolution = "3",
moveCAN = TRUE,
rawcols = FALSE
)
```

**Arguments**

- `ccaa`: A vector of names and/or codes for autonomous communities or NULL to get all the autonomous communities. See Details.
- `...`: Arguments passed on to `esp_get_nuts`
- `spatialtype`: Type of geometry to be returned:
  - "LB": Labels - point object.
  - "RG": Regions - polygon object.
- `year`: Release year. See `esp_get_nuts()` for `esp_get_ccaa()` and Details for `esp_get_ccaa_siane()`.
- `epsg`: projection of the map: 4-digit EPSG code. One of:
  - "4258": ETRS89
  - "4326": WGS84
  - "3035": ETRS89 / ETRS-LAEA
  - "3857": Pseudo-Mercator
- `cache`: A logical whether to do caching. Default is TRUE. See About caching.
- `update_cache`: A logical whether to update cache. Default is FALSE. When set to TRUE it would force a fresh download of the source file.
- `cache_dir`: A path to a cache directory. See About caching.
- `verbose`: Logical, displays information. Useful for debugging, default is FALSE.
- `resolution`: Resolution of the polygon. Values available are "3", "6.5" or "10".
- `moveCAN`: A logical TRUE/FALSE or a vector of coordinates c(lat,lon). It places the Canary Islands close to Spain’s mainland. Initial position can be adjusted using the vector of coordinates. See Displacing the Canary Islands.
- `rawcols`: Logical. Setting this to TRUE would add the raw columns of the dataset provided by IGN.

**Details**

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

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

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

**Value**

A sf object specified by `spatialtype`. 
About caching

You can set your cache_dir with esp_set_cache_dir().

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

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

Displacing the Canary Islands

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

Source

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

See Also

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

Examples

ccaa <- esp_get_ccaa()

library(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 resolution

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

# Transform to same CRS

europe <- st_transform(europe, 3035)
ccaa <- st_transform(ccaa, 3035)

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

---

esp_get_country

Get the borders of Spain as a sf polygon

Description

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

Usage

esp_get_country(...)  

Arguments

Arguments passed on to esp_get_nuts

or "2021".

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

cache A logical whether to do caching. Default is TRUE. See About caching.

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

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

verbose Logical, displays information. Useful for debugging, default is FALSE.

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

moveCAN A logical TRUE/FALSE or a vector of coordinates c(lat,lon). It places the Canary Islands close to Spain’s mainland. Initial position can be adjusted using the vector of coordinates. See Displacing the Canary Islands.

Value

A sf polygon object.

About caching

You can set your cache_dir with esp_set_cache_dir().

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

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

Displacing the Canary Islands

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

See Also

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

```r
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

```r
esp_get_hex_prov(prov = NULL)

esp_get_hex_ccaa(ccaa = NULL)

esp_get_grid_prov(prov = NULL)

esp_get_grid_ccaa(ccaa = NULL)
```

### Arguments

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

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

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

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

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

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

Value

A sf POLYGON object.

See Also

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

Examples

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

library(ggplot2)

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

hexprov <- esp_get_hex_prov()

ggplot(hexprov) +
  geom_sf(data = esp) +
  geom_sf(aes(fill = codauto),
    alpha = 0.3,
    show.legend = FALSE
  ) +
```

esp_get_hydrobasin

Get sf polygons of the drainage basin demarcations of Spain

Description

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

Usage

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

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()
esp_get_hypsobath

Examples

```r
ydroland <- esp_get_hydrobasin(domain = "land")
ydrolandsea <- 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 polygons and lines with the hypsometry and bathymetry of Spain*

**Description**

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

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

**Usage**

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

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

cache   A logical whether to do caching. Default is TRUE. See **About caching**.

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

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

verbose   Logical, displays information. Useful for debugging, default is FALSE.

resolution  Resolution of the shape. Values available are "3" or "6.5".

spatialtype  Spatial type of the output. Use "area" for polygons or "line" for lines.

Details

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

Value

A sf polygon or line object.

About caching

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

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

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

Source

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

See Also

Other natural: `esp_get_hydrobasin()`, `esp_get_rivers()`

Examples

```r
# This code would produce a nice plot - It will take a few seconds to run
library(ggplot2)
```
hypsobath <- esp_get_hypsobath()

# Error on the data provided - There is an empty shape
# Remove:

hypsobath <- hypsobath[!sf::st_is_empty(hypsobath), ]

# Tints from Wikipedia

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

hyps_tints <- colorRampPalette(
  rev(
    c(
      "#F5F4F2", "#E0DED8", "#CAC3B8", "#BAAE9A",
      "#AC9A7C", "#AA8753", "#B9985A", "#C3A76B",
      "#CAB982", "#D3CA9D", "#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(
```r
# 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",
    keyheight = .8
  ))
```

---

**esp_get_munic**  
*Get municipalities of Spain as sf polygons*

---

**Description**

Returns municipalities of Spain as polygons at a specified scale.

- `esp_get_munic()` uses GISCO (Eurostat) as source. Please use `giscoR::gisco_attributions()`

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

**Usage**

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

esp_get_munic

munic = NULL,
moveCAN = TRUE
)

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

Arguments

year Release year. See Details for years available.
epsg projection of the map: 4-digit EPSG code. One of:
- "4258": ETRS89
- "4326": WGS84
- "3035": ETRS89 / ETRS-LAEA
- "3857": Pseudo-Mercator

cache A logical whether to do caching. Default is TRUE. See About caching.
update_cache A logical whether to update cache. Default is FALSE. When set to TRUE it would force a fresh download of the source file.
cache_dir A path to a cache directory. See About caching.
verbose Logical, displays information. Useful for debugging, default is FALSE.
region A vector of names and/or codes for provinces or NULL to get all the municipalities. See Details.
munic A name or regex expression with the names of the required municipalities. NULL would not produce any filtering.
moveCAN A logical TRUE/FALSE or a vector of coordinates c(lat,lon). It places the Canary Islands close to Spain’s mainland. Initial position can be adjusted using the vector of coordinates. See Displacing the Canary Islands.
resolution Resolution of the polygon. Values available are "3", "6.5" or "10".
rawcols Logical. Setting this to TRUE would add the raw columns of the dataset provided by IGN.

Details

The years available are:
esp_get_munic


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

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

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

**Value**

A sf polygon

**About caching**

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

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

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

**Displacing the Canary Islands**

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

**Source**

GISCO API

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

**See Also**

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

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

Other municipalities: `esp_get_capimun()`, `esp_munic.sf`
Examples

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

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

# Load population data
data("pobmun19")

# 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()
```

Description

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

NUTS are provided at three different levels:
- "0": Country level
- "1": Groups of autonomous communities
- "2": Autonomous communities
- "3": Roughly matches the provinces, but providing specific individual objects for each major island

Usage

```r
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

- **epsg**: projection of the map: 4-digit **EPSG code**. One of:
  - "4258": ETRS89
  - "4326": WGS84
  - "3857": ETRS89 / ETRS-LAEA
  - "3035": 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.
esp_get_nuts

- "RG": Regions - polygon object.

region  Optional. A vector of region names, NUTS or ISO codes (see esp_dict_region_code()).

nuts_level  NUTS level. One of "0" (Country-level), "1", "2" or "3". See Description.

moveCAN  A logical TRUE/FALSE or a vector of coordinates c(lat,lon). It places the Canary Islands close to Spain’s mainland. Initial position can be adjusted using the vector of coordinates. See Displacing the Canary Islands.

Value

A sf object specified by spatialtype.

About caching

You can set your cache_dir with esp_set_cache_dir().

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

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

Displacing the Canary Islands

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

Note

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

Source

GISCO API

See Also

giscoR::gisco_get_nuts(), esp_dict_region_code().

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

Other nuts: esp_nuts.sf
Examples

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

library(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 polygons and points

---

**Description**

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

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

**Usage**

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

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

**Arguments**

- **prov**
  - A vector of names and/or codes for provinces or NULL to get all the provinces. See Details.
- **...**
  - Arguments passed on to `esp_get_nuts`
- **spatialtype**
  - Type of geometry to be returned:
    - "LB": Labels - point object.
    - "RG": Regions - polygon object.
- **year**
  - Release year. See `esp_get_nuts()` for `esp_get_prov()` and Details for `esp_get_prov_siane()`
esp_get_prov

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

cache  A logical whether to do caching. Default is TRUE. See About caching.

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

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

verbose  Logical, displays information. Useful for debugging, default is FALSE.

resolution  Resolution of the polygon. Values available are "3", "6.5" or "10".

moveCAN  A logical TRUE/FALSE or a vector of coordinates c(lat,lon). It places the Canary Islands close to Spain’s mainland. Initial position can be adjusted using the vector of coordinates. See Displacing the Canary Islands.

rawcols  Logical. Setting this to TRUE would add the raw columns of the dataset provided by IGN.

Details

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

Ceuta and Melilla are considered as provinces on this dataset.

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

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

Value

A sf object specified by spatialtype.

About caching

You can set your cache_dir with esp_set_cache_dir().

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

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

Displacing the Canary Islands

While moveCAN is useful for visualization, it would alter the actual geographic position of the Canary Islands. When using the output for spatial analysis or using tiles (e.g. with esp_getTiles() or addProviderEspTiles()) this option should be set to FALSE in order to get the actual coordinates, instead of the modified ones.
Source
IGN data via a custom CDN (see https://github.com/rOpenSpain/mapSpain/tree/sianedata).

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

Examples
prov <- esp_get_prov()

library(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) +
```r
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 lines and points with the railways of Spain

---

**Description**

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

**Usage**

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

**Arguments**

- `year`  
  Release year.

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

- `cache`  
  A logical whether to do caching. Default is TRUE. See About caching.

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

- **cache_dir**: A path to a cache directory. See **About caching**.
- **verbose**: Logical, displays information. Useful for debugging, default is FALSE.
- **spatialtype**: Spatial type of the output. Use "line" for extracting the railway as lines and "point" for extracting stations.

**Value**

A `sf` line or point object.

**About caching**

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

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

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

**Source**

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

**See Also**

Other infrastructure: `esp_get_roads()`

**Examples**

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

# Railways
rails <- esp_get_railway()

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

# Map
library(ggplot2)

ggplot(provs) +
  geom_sf(fill = "grey99", color = "grey50") +
  geom_sf(data = ccaa, fill = NA) +
  geom_sf(
    data = rails,
    aes(color = tipo),
    show.legend = FALSE, lwd = 1.5
```
Get sf polygon and lines of rivers, channels and other wetlands of Spain

Description

Loads a sf polygon or line object representing rivers, channels, reservoirs and other wetlands of Spain

Usage

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

Arguments

- **epsg**
  - projection of the map: 4-digit EPSG code. One of:
    - "4258": ETRS89
    - "4326": WGS84
    - "3035": ETRS89 / ETRS-LAEA
    - "3857": Pseudo-Mercator
- **cache**
  - A logical whether to do caching. Default is TRUE. See About caching.
esp_get_rivers

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

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

verbose Logical, displays information. Useful for debugging, default is FALSE.

resolution Resolution of the polygon. Values available are "3", "6.5" or "10".

spatialtype Spatial type of the output. Use "area" for polygons or "line" for lines.

name Optional. A character or regex expression with the name of the element(s) to be extracted.

Details

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

Value

A sf polygon or line object.

Source

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

See Also

Other natural: esp_get_hydrobasin(), esp_get_hypsobath()

Examples

# Use of regex

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

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

# See the difference

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

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

sf::st_bbox(MainRivers)
library(ggplot2)

ggplot(shapeEsp) +
# 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 lines of the roads of Spain*

**Description**

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

**Usage**

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

**Arguments**

- `year` Release year. See Details for years available.
esp_get_roads

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

cache  
A logical whether to do caching. Default is TRUE. See About caching.

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

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

verbose  
Logical, displays information. Useful for debugging, default is FALSE.

moveCAN  
A logical TRUE/FALSE or a vector of coordinates c(lat,lon). It places the Canary Islands close to Spain’s mainland. Initial position can be adjusted using the vector of coordinates. See Displacing the Canary Islands.

Details

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

Value

A sf line object.

About caching

You can set your cache_dir with esp_set_cache_dir().

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

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

Displacing the Canary Islands

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

Source

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

See Also

Other infrastructure: esp_get_railway()
Examples

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

library(ggplot2)

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")

Description

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

Format

A POLYGON data frame (resolution: 1:1million, EPSG:4258) object with 8,131 rows and fields:

- **codauto**: INE code of each autonomous community.
- **ine.ccaa.name**: INE name of each autonomous community.
- **cpro**: INE code of each province.
- **ine.prov.name**: INE name of each province.
- **cmun**: INE code of each municipality.
- **name**: Name of the municipality.
- **LAU_CODE**: LAU Code (GISCO) of the municipality. This is a combination of **cpro** and **cmun**, aligned with INE coding scheme.
- **geometry**: geometry field.

Source

esp_nuts.sf

See Also

esp_get_munic().

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

Other municipalities: esp_get_capimun(), esp_get_munic()

Examples

data("esp_munic.sf")
teruel_cpro <- esp_dict_region_code("Teruel", destination = "cpro")
teruel_sf <- esp_munic.sf[esp_munic.sf$cpro == teruel_cpro, ]
teruel_city <- teruel_sf[teruel_sf$name == "Teruel", ]

# Plot
library(ggplot2)
library(ggspatial)

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

---

esp_nuts.sf  All NUTS POLYGON object of Spain

Description

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

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

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

Source


See Also

Other datasets: esp_codelist, esp_munic.sf, leaflet.providersESP.df, pobmun19
Other nuts: esp_get_nuts()

Examples

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), lwd = .1) +
Description

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

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

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

Usage

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

Arguments

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

[Experimental]

This is a wrapper of `ggspatial::layer_spatial.Raster()` that works with SpatRaster objects. This function is likely to be deprecated in the future when `ggspatial` (or any other package) provides native support to SpatRaster on `ggplot`. See also [https://github.com/paleolimbot/ggspatial/issues/91](https://github.com/paleolimbot/ggspatial/issues/91)

Other packages that supports natively SpatRaster:

- `tmap`
- `mapsf`
- `rasterVis`

### Usage

```r
layer_spatraster(data, ...)
```

### Arguments

- `data` A SpatRaster object created with `terra::rast()`.
- `...` Arguments passed on to `ggspatial::layer_spatial`

Mapping A mapping, created using `aes`.
Details

This function requires both **ggspatial** and **raster** packages.
You can install both running `install.packages("ggspatial", dependencies = TRUE)`

Value

A ggplot2 layer

See Also

`ggspatial::layer_spatial.Raster(), raster::stack()`.  
Other imagery utilities: `addProviderEspTiles(), esp_getTiles(), leaflet.providersESP.df`

Examples

```r
# Get a SpatRaster
x <- esp_get_ccaa("Galicia")
tile <- esp_getTiles(x, "PNOA")
class(tile)
library(ggplot2)
ggplot(x) +
  layer_spatraster(tile) +
  geom_sf(color = "yellow", fill = NA) +
  theme_minimal()
```

---

**leaflet.providersESP.df**

*Public WMS and WMTS of Spain*

Description

A data frame containing information of different public WMS and WMTS providers of Spain
This function is a implementation of the javascript plugin leaflet-providersESP v1.2.0.
Format

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

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

Details

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

```r
provider
  'IDErioja'
  'IGNBase'
  'IGNBase.Todo'
  'IGNBase.Gris'
  'IGNBase.TodoNoFondo'
  'IGNBase.Orto'
  'MDT'
  'MDT.Elevaciones'
  'MDT.Relieve'
  'MDT.CurvasNivel'
  'PNOA'
  'PNOA.MaximaActualidad'
  'PNOA.Mosaico'
  'OcupacionSuelo'
  'OcupacionSuelo.Ocupacion'
  'OcupacionSuelo.Usos'
  'LiDAR'
  'MTN'
  'Geofisica'
  'Geofisica.Terremotos10dias'
  'Geofisica.Terremotos30dias'
  'Geofisica.Terremotos365dias'
  'Geofisica.VigilanciaVolcanica'
  'CaminoDeSantiago'
  'CaminoDeSantiago.CaminoFrances'
  'CaminoDeSantiago.CaminosTuronensis'
  'CaminoDeSantiago.CaminosGalicia'
  'CaminoDeSantiago.CaminosDelNorte'
  'CaminoDeSantiago.CaminosAndaluces'
  'CaminoDeSantiago.CaminosCentro'
  'CaminoDeSantiago.CaminosEste'
  'CaminoDeSantiago.CaminosCatalanes'
  'CaminoDeSantiago.CaminosSureste'
  'CaminoDeSantiago.CaminosInsulares'
  'CaminoDeSantiago.CaminosPiemonts'
  'CaminoDeSantiago.CaminosTolosana'
```
'CaminoDeSantiago.CaminosPortugueses'
'Catastro'
'Catastro.Catastro'
'Catastro.Parcela'
'Catastro.CadastralParcel'
'Catastro.CadastralZoning'
'Catastro.Address'
'Catastro.Building'
'RedTransporte'
'RedTransporte.Carreteras'
'RedTransporte.Ferroviario'
'RedTransporte.Aerodromo'
'RedTransporte.AreaServicio'
'RedTransporte.EstacionesFerroviario'
'RedTransporte.Puertos'
'Cartociudad'
'Cartociudad.CodigosPostales'
'Cartociudad.Direccion'
'NombresGeograficos'
'UnidadesAdm'
'UnidadesAdm.Limites'
'UnidadesAdm.Unidades'
'Hidrografía'
'Hidrografía.MasaAgua'
'Hidrografía.Cuencas'
'Hidrografía.Subcuencas'
'Hidrografía.POIs'
'Hidrografía.ManMade'
'Hidrografía.LineaCosta'
'Hidrografía.Rios'
'Hidrografía.Humedales'
'Militar'
'Militar.CEGET1M'
'Militar.CEGETM7814'
'Militar.CEGETM7815'
'Militar.CEGETM682'
'Militar.CECAF1M'
'ADIF'
'ADIF.Vias'
'ADIF.Nodos'
'ADIF.Estaciones'
'LimitesMaritimos'
'LimitesMaritimos.LimitesMaritimos'
'LimitesMaritimos.LineasBase'
'Copernicus'
'Copernicus.LandCover'
'Copernicus.Forest'
'Copernicus.ForestLeaf'
'Copernicus.WaterWet'
'Copernicus.SoilSeal'
'Copernicus.GrassLand'
'Copernicus.Local'
'Copernicus.RiparianGreen'
'Copernicus.RiparianLandCover'
'Copernicus.Natura2k'
'Copernicus.UrbanAtlas'
'ParquesNaturales'
'ParquesNaturales.Limites'
'ParquesNaturales.ZonasPerifericas'

Source


See Also

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

Examples

data("leaflet.providersESP.df")

<table>
<thead>
<tr>
<th>pobmun19</th>
<th>Population by municipality (2019)</th>
</tr>
</thead>
</table>

Description

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

Source

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

See Also

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

Examples

data("pobmun19")
Index

* cache utilities
  esp_clear_cache, 6
  esp_set_cache_dir, 49
* datasets
  esp_codelist, 7
  esp_munic.sf, 46
  esp_nuts.sf, 47
  leaflet.providersESP.df, 51
  pobmun19, 54
* dictionary
  esp_codelist, 7
  esp_dict_region_code, 9
* helper
  esp_check_access, 6
* imagery utilities
  addProviderEspTiles, 4
  esp_getTiles, 10
  layer_spatraster, 50
  leaflet.providersESP.df, 51
* infrastructure
  esp_get_railway, 40
  esp_get_roads, 44
* municipalities
  esp_get_capimun, 15
  esp_get_munic, 30
  esp_munic.sf, 46
* natural
  esp_get_hydrobasin, 25
  esp_get_hypsobath, 27
  esp_get_rivers, 42
* nuts
  esp_get_nuts, 33
  esp_nuts.sf, 47
* package
  mapSpain-package, 3
* political
  esp_codelist, 7
  esp_get_can_box, 13
  esp_get_capimun, 15
  esp_get_ccaa, 18
  esp_get_country, 21
  esp_get_gridmap, 23
  esp_get_munic, 30
  esp_get_nuts, 33
  esp_get_prov, 37
addProviderEspTiles, 4, 12, 51, 54
addProviderEspTiles(), 14, 17, 20, 22, 32, 35, 38, 45
aes, 50
base::regex(), 32
base::tempdir(), 49
esp_check_access, 6
esp_clear_cache, 6, 50
esp_codelist, 7, 9, 14, 16, 17, 19, 20, 22, 24, 32, 35, 38, 39, 47, 48, 54
esp_dict_region_code, 8, 9
esp_dict_region_code(), 9, 35
esp_dict_translate
  (esp_dict_region_code), 9
esp_dict_translate(), 9
esp_get_can_box, 8, 13, 17, 20, 22, 24, 32, 35, 39
esp_get_can_box(), 13
esp_get_can_provinces
  (esp_get_can_box), 13
esp_get_can_provinces(), 13
esp_get_capimun, 8, 14, 15, 20, 22, 24, 32, 35, 39, 47
esp_get_ccaa, 8, 14, 17, 18, 22, 24, 32, 35, 39
esp_get_ccaa(), 18, 19
esp_get_ccaa_siane (esp_get_ccaa), 18
esp_get_ccaa_siane(), 18, 19
esp_get_country, 8, 14, 17, 20, 21, 24, 32, 35, 39
esp_get_grid_ccaa (esp_get_gridmap), 23
esp_get_grid_prov (esp_get_gridmap), 23
esp_get_gridmap, 8, 14, 17, 20, 22, 23, 32, 35, 39
esp_get_hex_ccaa (esp_get_gridmap), 23
esp_get_hex_prov (esp_get_gridmap), 23
esp_get_hydrobasin, 25, 28, 43
esp_get_hypsobath, 26, 27, 43
esp_get_munic, 8, 14, 17, 20, 22, 24, 30, 35, 39, 47
esp_get_munic (), 15, 30, 32, 47
esp_get_munic_siane (esp_get_munic), 30
esp_get_munic_siane (), 30, 32
esp_get_nuts, 8, 14, 17, 19–22, 24, 32, 33, 37, 39, 48
esp_get_nuts (), 13, 19, 37
esp_get_prov, 8, 14, 17, 20, 22, 24, 32, 35, 37
esp_get_prov (), 37
esp_get_prov_siane (esp_get_prov), 37
esp_get_prov_siane (), 37, 38
esp_get_railway, 40, 45
esp_get_rivers, 26, 28, 42
esp_get_roads, 41, 44
esp_getTiles, 5, 10, 51, 54
esp_getTiles (), 14, 17, 20, 22, 32, 35, 38, 45, 52
esp_munic.sf, 8, 17, 32, 46, 48, 54
esp_nuts.sf, 8, 35, 47, 47, 54
esp_set_cache_dir, 7, 49
esp_set_cache_dir (), 12, 17, 20, 22, 26, 28, 32, 35, 38, 41, 45

ggspatial::layer_spatial, 50
ggspatial::layer_spatial.Raster (), 50, 51
giscoR::gisco_attribution, 18, 30, 35, 37

giscoR::gisco_check_access (), 6
giscoR::gisco_get_communes (), 32
giscoR::gisco_get_lau (), 32
giscoR::gisco_get_nuts (), 35

layer_spatraster, 5, 12, 50, 54
leaflet.providersESP.df, 4, 5, 8, 11, 12, 47, 48, 51, 51, 54
leaflet::addTiles (), 4, 5
leaflet::leaflet (), 4, 5
leaflet::providerTileOptions, 4
leaflet::providerTileOptions (), 5
leaflet::tileOptions (), 5

mapSpain (mapSpain-package), 3
mapSpain-package, 3
pobmun19, 8, 47, 48, 54, 54
providerEspTileOptions
(providerEspTiles), 4
providerEspTileOptions (), 5
rappdirs::user_config_dir (), 50
raster::stack (), 51
regex, 16, 31, 43
sf::st_transform (), 24
terra::rast (), 12, 50