Package 'geonuts'

March 18, 2026

Type Package

Title Identification and Visualisation of European NUTS Regions from Geolocations

Version 1.0.1

Description Provides functions to identify European NUTS (Nomenclature of Territorial Units for Statistics) regions for geographic coordinates (latitude/longitude) using Eurostat geospatial boundaries. Includes map-based visualisation of the matched regions for validation and exploration. Designed for regional data analysis, reproducible workflows, and integration with common geospatial R packages.

License GPL (>= 3)

URL `https://github.com/aikatona/geonuts`

BugReports `https://github.com/aikatona/geonuts/issues`

Depends R (>= 4.1.0)

Imports eurostat, ggplot2, giscoR, units, grid, sf

Suggests knitr, rmarkdown, testthat (>= 3.0.0)

Config/testthat/edition 3

Encoding UTF-8

RoxygenNote 7.3.2

VignetteBuilder knitr

NeedsCompilation no

Author Attila I. Katona [aut, cre],
Marcell T. Kurbucz [aut]

Maintainer Attila I. Katona <`katona.attila@gtk.uni-pannon.hu`>

Repository CRAN

Date/Publication 2026-03-18 18:00:02 UTC

Contents

`get_nuts` *Identify NUTS Regions for Geolocations*

Description

Vectorised identification of NUTS regions for input coordinates using Eurostat geospatial layers. Supports a single level (0–3) or `"all"` to return all levels. Optional country pre-filter and "nearest" fallback make the function robust to points falling just outside polygon boundaries.

Usage

```
get_nuts(
  latitude,
  longitude,
  level = "all",
  year = 2021,
  resolution = 20,
  crs = 4326,
  country = NULL,
  match_strategy = c("within", "nearest"),
  nearest_max_km = Inf,
  verbose = TRUE
)
```

Arguments

`latitude`	(numeric, **mandatory**) Latitudes in decimal degrees (WGS84). Must be within `[-90, 90]`.
`longitude`	(numeric, **mandatory**) Longitudes in decimal degrees (WGS84). Must be within `[-180, 180]`. Length must match `latitude`.
`level`	(integer or character, optional) One of `0`, `1`, `2`, `3`, `"all"`. Default: `"all"`. When `"all"`, the result includes `nuts0..nuts3`.
`year`	(integer, optional) NUTS reference year for the Eurostat layer. Default: `2021`.
`resolution`	(integer, optional) Eurostat map resolution. Typical values: `1`, `3`, `10`, `20`, `60`. Default: `20`.
`crs`	(integer, optional) EPSG code of input coordinates. If not 4326, inputs are transformed to WGS84 (EPSG:4326). Default: `4326`.
`country`	(character, optional) Two-letter `CNTR_CODE` to pre-filter polygons (e.g., `"DE"`, `"IT"`). Reduces memory and speeds up queries.
`match_strategy`	(character, optional) Matching strategy: `"within"` (default) assigns a NUTS only when the point lies within a polygon; `"nearest"` uses the nearest NUTS polygon for unmatched points.
`nearest_max_km`	(numeric, optional) Maximum distance (km) for nearest fallback; use `Inf` to allow any distance. Default: `Inf`.
`verbose`	(logical, optional) Print informative messages. Default: `TRUE`.

Details

- Geometries are downloaded via `eurostat::get_eurostat_geospatial()` and cached per `(level, year, resolution)` to avoid repeated I/O.
- Polygons are pre-filtered using the points' bounding box to accelerate joins on continental layers (with safe fallback).
- Nearest distances are computed with units and converted to kilometers.

Value

If `level` is a single integer: a `data.frame` with columns `lat`, `lon`, `nuts`, `cntr_code`, `match_status`, `match_dist_km`, `level`, `year`, `resolution`. If `level = "all"`: a `data.frame` with `lat`, `lon`, `nuts0`, `nuts1`, `nuts2`, `nuts3`, `cntr_code`, `match_status`, `match_dist_km`, `year`, `resolution`. Here `match_status`/`match_dist_km` are computed using level 3 (most granular).

See Also

`map_nuts`

Examples

```
res <- get_nuts(52.52, 13.405, level = 3, year = 2021, resolution = 20)
head(res)
```

`map_nuts` *Plot NUTS Matches on a Map*

Description

Visualises the frequency of matched NUTS regions and (optionally) overlays the input points (matched vs. unmatched) for validation. Works with both single-level and multi-level (`level = "all"`) outputs from `get_nuts`.

Usage

```
map_nuts(
  nuts,
  map_level = 3,
  country = NULL,
  show_points = TRUE,
  border_col = "lightgrey",
  low_col = "lightgreen",
  high_col = "darkgreen",
  id_col = "black",
  uid_col = "red",
  verbose = TRUE
)
```

Arguments

`nuts`	(data.frame, **mandatory**) Output of `get_nuts`. Must include `lat`, `lon`, and either `nuts` (single-level) or any of `nuts0..nuts3`, plus `year`, `resolution`. If both single and multi-level columns exist, `map_level` selects which to display.
`map_level`	(integer, optional) NUTS level to display when multiple levels are present. One of `0, 1, 2, 3`. Default: `3`.
`country`	(character, optional) Two-letter `CNTR_CODE` to filter the map polygons (e.g., `"DE"`, `"FR"`). If `NULL`, uses all available regions. When supplied, the map is always zoomed to the full extent of the selected country.
`show_points`	(logical, optional) Overlay input points. Default: `TRUE`.
`border_col`	(character, optional) Polygon border colour. Default: `"lightgrey"`.
`low_col`	(character, optional) Fill colour for lower frequencies. Default: `"lightgreen"`.
`high_col`	(character, optional) Fill colour for higher frequencies. Default: `"darkgreen"`.
`id_col`	(character, optional) Point colour for matched/nearest inputs. Default: `"black"`.
`uid_col`	(character, optional) Point colour for unmatched inputs. Default: `"red"`.
`verbose`	(logical, optional) Print informative messages. Default: `TRUE`.

Value

A `ggplot2` object showing a choropleth of NUTS frequencies with optional point overlays.

See Also

`get_nuts`

Examples

```
res <- get_nuts(52.52, 13.405, level = 3, year = 2021, resolution = 20)
p <- map_nuts(res, map_level = 3)
print(p)
```

Index