

# Package ‘placematchr’

February 11, 2026

**Title** Normalize and Match City Names to NUTS Regions

**Version** 0.2.3

**Date** 2026-02-08

## Description

Normalizes city names for Germany (DE) and Switzerland (CH) and matches them to NUTS 3 regions using provided crosswalks. Features include comprehensive normalization rules, cascading matching logic (Exact NUTS -> Exact LAU -> Fuzzy), and single-source data synthesis. The package implements the NUTS classification as described in the NUTS methodology (Eurostat (2021) <<https://ec.europa.eu/eurostat/web/nuts>>).

**License** MIT + file LICENSE

**Encoding** UTF-8

**LazyData** true

**Imports** dplyr, stringr, stringdist, data.table, tidyr, rlang

**Suggests** testthat, readxl

**RoxygenNote** 7.3.2

**NeedsCompilation** no

**Author** Giulian Etingin-Frati [aut, cre]

**Maintainer** Giulian Etingin-Frati <etingin-frati@kof.ethz.ch>

**Depends** R (>= 3.5.0)

**Repository** CRAN

**Date/Publication** 2026-02-11 19:50:02 UTC

## Contents

generate_fake_cities . . . . .	2
lau_data . . . . .	2
match_city . . . . .	5
normalize_city . . . . .	6
nuts_data . . . . .	6

<b>Index</b>	<b>10</b>
--------------	-----------

---

`generate_fake_cities` *Generate Fake City Data*

---

### Description

Generates a vector of fake city names for testing, including common variations and noise.

### Usage

```
generate_fake_cities(n = 10, country = "DE")
```

### Arguments

`n` Integer, matching number of cities to generate.  
`country` "DE" or "CH".

### Value

Character vector of city names.

### Examples

```
# Generate 5 fake German cities
generate_fake_cities(5, country = "DE")

# Generate 3 fake Swiss cities
generate_fake_cities(3, country = "CH")
```

---

`lau_data` *Local Administrative Units (LAU) Crosswalks*

---

### Description

Datasets containing mappings from city names to LAU codes and NUTS 3 regions for various countries. The data handles string normalization and matches cities to their respective statistical regions.

### Usage

```
lau_at
lau_be
lau_bg
lau_ch
```

lau\_cy

lau\_cz

lau\_de

lau\_dk

lau\_ee

lau\_el

lau\_es

lau\_fi

lau\_fr

lau\_hr

lau\_hu

lau\_ie

lau\_it

lau\_li

lau\_lt

lau\_lu

lau\_lv

lau\_mk

lau\_mt

lau\_nl

lau\_no

lau\_pl

lau\_pt

lau\_ro

lau\_se

lau\_si

lau\_sk

lau\_tr

### **Format**

Data frames with varying columns depending on the country, typically including:

**lau\_id** Local Administrative Unit code

**lau\_name** Name of the Local Administrative Unit

**nuts\_3\_id** NUTS 3 region code

**population** Population (if available)

An object of class data.frame with 2093 rows and 5 columns.

An object of class data.frame with 571 rows and 5 columns.

An object of class data.frame with 265 rows and 5 columns.

An object of class data.frame with 2135 rows and 5 columns.

An object of class data.frame with 617 rows and 5 columns.

An object of class data.frame with 6258 rows and 5 columns.

An object of class data.frame with 10972 rows and 5 columns.

An object of class data.frame with 99 rows and 5 columns.

An object of class data.frame with 79 rows and 5 columns.

An object of class data.frame with 6142 rows and 5 columns.

An object of class data.frame with 8132 rows and 5 columns.

An object of class data.frame with 309 rows and 5 columns.

An object of class data.frame with 32774 rows and 5 columns.

An object of class data.frame with 556 rows and 5 columns.

An object of class data.frame with 3155 rows and 5 columns.

An object of class data.frame with 166 rows and 5 columns.

An object of class data.frame with 7900 rows and 5 columns.

An object of class data.frame with 11 rows and 5 columns.

An object of class data.frame with 60 rows and 5 columns.

An object of class data.frame with 100 rows and 5 columns.

An object of class data.frame with 43 rows and 5 columns.

An object of class data.frame with 80 rows and 5 columns.

An object of class data.frame with 68 rows and 5 columns.

An object of class data.frame with 342 rows and 5 columns.  
An object of class data.frame with 378 rows and 5 columns.  
An object of class data.frame with 2477 rows and 5 columns.  
An object of class data.frame with 3092 rows and 5 columns.  
An object of class data.frame with 3181 rows and 5 columns.  
An object of class data.frame with 290 rows and 5 columns.  
An object of class data.frame with 211 rows and 5 columns.  
An object of class data.frame with 2927 rows and 5 columns.  
An object of class data.frame with 972 rows and 5 columns.

### Source

Eurostat and national statistical institutes.

---

match_city	<i>Match City Names to NUTS Regions</i>
------------	---

---

### Description

Matches a vector of city names to NUTS 3 regions using a cascading logic for any supported country.

### Usage

```
match_city(x, country = "DE", fuzzy = TRUE, threshold = 0.95)
```

### Arguments

x	Character vector of city names.
country	Character string of two-letter country code (e.g. "DE", "FR").
fuzzy	Logical, whether to perform fuzzy matching.
threshold	Numeric, similarity threshold for fuzzy matching (0-1).

### Value

A data frame with columns: original, city\_clean, nuts\_3\_id, lau\_name, match\_type, similarity.

### Examples

```
# Match German cities
cities <- c("Berlin", "Munich", "Hamburg")
match_city(cities, country = "DE")

# Match with exact matching only (no fuzzy)
match_city(c("Frankfurt am Main"), country = "DE", fuzzy = FALSE)
```

---

normalize_city	<i>Normalize City Names</i>
----------------	-----------------------------

---

**Description**

Normalizes city names for EEA countries using comprehensive rules tailored to each language/region.

**Usage**

```
normalize_city(x, country = "DE")
```

**Arguments**

x	Character vector of city names.
country	Character string of the ISO 2-character country code (e.g. "DE", "FR", "PL").

**Value**

Character vector of normalized names.

**Examples**

```
# Normalize German city names
# Normalize German city names
normalize_city(c("M\u00FCnchen", "K\u00F6ln", "Frankfurt a.M."), country = "DE")

# Normalize Swiss city names
normalize_city(c("Z\u00FCrich", "Gen\u00E8ve", "Basel-Stadt"), country = "CH")
```

---

nuts_data	<i>NUTS 3 Region Metadata</i>
-----------	-------------------------------

---

**Description**

Metadata for NUTS 3 regions for various countries, used for hierarchical matching.

**Usage**

```
nuts_at
nuts_be
nuts_bg
nuts_ch
```

nuts\_cy

nuts\_cz

nuts\_de

nuts\_dk

nuts\_ee

nuts\_el

nuts\_es

nuts\_fi

nuts\_fr

nuts\_hr

nuts\_hu

nuts\_ie

nuts\_it

nuts\_li

nuts\_lt

nuts\_lu

nuts\_lv

nuts\_mk

nuts\_mt

nuts\_nl

nuts\_no

nuts\_pl

nuts\_pt

nuts\_ro

nuts\_se

nuts\_si

nuts\_sk

nuts\_tr

### Format

Data frames with columns:

**nuts\_3\_id** NUTS 3 region code

**nuts\_3\_name** Name of the NUTS 3 region

An object of class data.frame with 35 rows and 4 columns.

An object of class data.frame with 43 rows and 4 columns.

An object of class data.frame with 28 rows and 4 columns.

An object of class data.frame with 26 rows and 4 columns.

An object of class data.frame with 1 rows and 4 columns.

An object of class data.frame with 14 rows and 4 columns.

An object of class data.frame with 401 rows and 4 columns.

An object of class data.frame with 11 rows and 4 columns.

An object of class data.frame with 5 rows and 4 columns.

An object of class data.frame with 53 rows and 4 columns.

An object of class data.frame with 59 rows and 4 columns.

An object of class data.frame with 19 rows and 4 columns.

An object of class data.frame with 96 rows and 4 columns.

An object of class data.frame with 21 rows and 4 columns.

An object of class data.frame with 20 rows and 4 columns.

An object of class data.frame with 8 rows and 4 columns.

An object of class data.frame with 107 rows and 4 columns.

An object of class data.frame with 1 rows and 4 columns.

An object of class data.frame with 10 rows and 4 columns.

An object of class data.frame with 1 rows and 4 columns.

An object of class data.frame with 5 rows and 4 columns.

An object of class data.frame with 8 rows and 4 columns.

An object of class data.frame with 2 rows and 4 columns.

An object of class data.frame with 40 rows and 4 columns.

An object of class data.frame with 17 rows and 4 columns.

An object of class data.frame with 73 rows and 4 columns.



An object of class data.frame with 26 rows and 4 columns.

An object of class data.frame with 42 rows and 4 columns.

An object of class data.frame with 21 rows and 4 columns.

An object of class data.frame with 12 rows and 4 columns.

An object of class data.frame with 8 rows and 4 columns.

An object of class data.frame with 81 rows and 4 columns.

**Source**

Eurostat

# Index

## \* datasets

lau\_data, 2  
nuts\_data, 6

generate\_fake\_cities, 2

lau\_at (lau\_data), 2  
lau\_be (lau\_data), 2  
lau\_bg (lau\_data), 2  
lau\_ch (lau\_data), 2  
lau\_cy (lau\_data), 2  
lau\_cz (lau\_data), 2  
lau\_data, 2  
lau\_de (lau\_data), 2  
lau\_dk (lau\_data), 2  
lau\_ee (lau\_data), 2  
lau\_el (lau\_data), 2  
lau\_es (lau\_data), 2  
lau\_fi (lau\_data), 2  
lau\_fr (lau\_data), 2  
lau\_hr (lau\_data), 2  
lau\_hu (lau\_data), 2  
lau\_ie (lau\_data), 2  
lau\_it (lau\_data), 2  
lau\_li (lau\_data), 2  
lau\_lt (lau\_data), 2  
lau\_lu (lau\_data), 2  
lau\_lv (lau\_data), 2  
lau\_mk (lau\_data), 2  
lau\_mt (lau\_data), 2  
lau\_nl (lau\_data), 2  
lau\_no (lau\_data), 2  
lau\_pl (lau\_data), 2  
lau\_pt (lau\_data), 2  
lau\_ro (lau\_data), 2  
lau\_se (lau\_data), 2  
lau\_si (lau\_data), 2  
lau\_sk (lau\_data), 2  
lau\_tr (lau\_data), 2

match\_city, 5

normalize\_city, 6  
nuts\_at (nuts\_data), 6  
nuts\_be (nuts\_data), 6  
nuts\_bg (nuts\_data), 6  
nuts\_ch (nuts\_data), 6  
nuts\_cy (nuts\_data), 6  
nuts\_cz (nuts\_data), 6  
nuts\_data, 6  
nuts\_de (nuts\_data), 6  
nuts\_dk (nuts\_data), 6  
nuts\_ee (nuts\_data), 6  
nuts\_el (nuts\_data), 6  
nuts\_es (nuts\_data), 6  
nuts\_fi (nuts\_data), 6  
nuts\_fr (nuts\_data), 6  
nuts\_hr (nuts\_data), 6  
nuts\_hu (nuts\_data), 6  
nuts\_ie (nuts\_data), 6  
nuts\_it (nuts\_data), 6  
nuts\_li (nuts\_data), 6  
nuts\_lt (nuts\_data), 6  
nuts\_lu (nuts\_data), 6  
nuts\_lv (nuts\_data), 6  
nuts\_mk (nuts\_data), 6  
nuts\_mt (nuts\_data), 6  
nuts\_nl (nuts\_data), 6  
nuts\_no (nuts\_data), 6  
nuts\_pl (nuts\_data), 6  
nuts\_pt (nuts\_data), 6  
nuts\_ro (nuts\_data), 6  
nuts\_se (nuts\_data), 6  
nuts\_si (nuts\_data), 6  
nuts\_sk (nuts\_data), 6  
nuts\_tr (nuts\_data), 6