

# Package ‘csdata’

March 31, 2026

**Title** Structural Data for Norway

**Version** 2026.3.30

**Description** Datasets relating to population in municipalities, municipality/county matching, and how different municipalities have merged/redistricted over time from 2006 to 2024.

**URL** <https://niphr.github.io/csdata/>, <https://github.com/niphr/csdata>

**BugReports** <https://github.com/niphr/csdata/issues>

**Depends** R (>= 3.5.0)

**License** MIT + file LICENSE

**Encoding** UTF-8

**Imports** data.table, stats, utils

**Suggests** testthat, broom, PxWebApiData, crayon, dplyr, forcats, fs, geojsonio, ggplot2, glue, gt, knitr, lubridate, magrittr, mapproj, methods, ncd4, purrr, readxl, reshape2, rmarkdown, rmapshaper, rstudioapi, stringr, sp, sf, tidyr, zoo

**RoxygenNote** 7.2.3

**VignetteBuilder** knitr

**Date/Publication** 2026-03-31 09:10:02 UTC

**NeedsCompilation** no

**Author** Richard Aubrey White [aut, cre] (ORCID: <https://orcid.org/0000-0002-6747-1726>),  
Chi Zhang [aut] (ORCID: <https://orcid.org/0000-0003-0501-5909>)

**Maintainer** Richard Aubrey White <hello@rwhite.no>

**Repository** CRAN

## Contents

add_granularity_geo_to_data_set . . . . .	2
add_iso3_to_data_set . . . . .	3
config . . . . .	3
location_code_to_granularity_geo . . . . .	4

location_code_to_iso3 . . . . .	4
nb . . . . .	5
nor_locations_hierarchy_from_to . . . . .	5
nor_locations_names . . . . .	6
nor_locations_redistricting . . . . .	7
nor_population_by_age_cats . . . . .	8
se . . . . .	9
set_config . . . . .	10

<b>Index</b>	<b>11</b>
--------------	-----------

---

add\_granularity\_geo\_to\_data\_set  
*Adds granularity\_geo to a given data set*

---

## Description

Adds granularity\_geo to a given data set

## Usage

```
add_granularity_geo_to_data_set(x, location_reference = NULL)
```

## Arguments

x                    A data.table containing a column called "location\_code".  
location\_reference                    A location reference data.table.

## Value

A data.table containing an extra column called "granularity\_geo".

## Examples

```
library(data.table)
data <- data.table(location_code = c("norge", "county03", "blah"))
csdata::add_granularity_geo_to_data_set(data)
print(data)

library(data.table)
data <- data.table(location_code = c("norge", "county03", "blah"))
csdata::add_granularity_geo_to_data_set(data, location_reference = csdata::nor_locations_names())
print(data)
```

---

add\_iso3\_to\_data\_set *Adds iso3 to a given data set*

---

**Description**

Adds iso3 to a given data set

**Usage**

```
add_iso3_to_data_set(x)
```

**Arguments**

x                    A data.table containing a column called "location\_code".

**Value**

A data.table containing an extra column called "iso3".

**Examples**

```
library(data.table)
data <- data.table(location_code = c("norge", "county03", "blah"))
csdata::add_iso3_to_data_set(data)
print(data)
```

---

config                    *An environment containing configuration variables*

---

**Description**

Available configuration variables:

- border\_nor (default 2024): The year in which Norwegian geographical boundaries were designated. Valid values: 2020, 2024.

**Usage**

```
config
```

**Format**

An object of class environment of length 1.

**Examples**

```
print(ls(csdata::config))
for(i in names(csdata::config)){
  cat(i, ":", csdata::config[[i]], "\n")
}
```

---

location\_code\_to\_granularity\_geo  
*Convert location\_code to granularity\_geo*

---

**Description**

Convert location\_code to granularity\_geo

**Usage**

```
location_code_to_granularity_geo(x, location_reference = NULL)
```

**Arguments**

`x` Either a vector, or a data.frame/data.table containing a column called "location\_code".

`location_reference` A location reference data.table.

**Value**

Character vector the same length as `x`, containing the corresponding `granularity_geo`.

**Examples**

```
csdata::location_code_to_granularity_geo(c("nation_nor", "county_nor03"))
```

---

location\_code\_to\_iso3 *Convert location\_code to iso3*

---

**Description**

Convert location\_code to iso3

**Usage**

```
location_code_to_iso3(x)
```

**Arguments**

x Either a vector, or a data.frame/data.table containing a column called "location\_code".

**Value**

Character vector the same length as x, containing the corresponding iso3.

**Examples**

```
csdata::location_code_to_iso3(c("nation_nor", "county_nor03"))
```

---

nb	<i>Norwegian characters in unicode</i>
----	--

---

**Description**

Norwegian characters in unicode

**Usage**

```
nb
```

**Format**

An object of class list of length 6.

**Examples**

```
print(csdata::nb)
```

---

nor_locations_hierarchy_from_to	<i>Location hierarchies in Norway</i>
---------------------------------	---------------------------------------

---

**Description**

Calculates the relationship between different locations in Norway, according to geographic granularity. For example, which municipalities are inside which counties.

**Usage**

```
nor_locations_hierarchy_from_to(
  from,
  to,
  include_to_name = FALSE,
  border = csdata::config$border_nor
)
```

**Arguments**

from	wardoslo, wardbergen, wardtrondheim, wardstavanger, municip, baregion, county, georegion, mtregion, notmainlandmunicip, notmainlandcounty, missingmunicip, missingcounty
to	wardoslo, wardbergen, wardtrondheim, wardstavanger, municip, baregion, county, georegion, mtregion, notmainlandmunicip, notmainlandcounty, missingmunicip, missingcounty
include_to_name	Do you want to include the name of the 'to' location?
border	The year in which Norwegian geographical boundaries were designated (2020, 2024).

**Value**

Data.table containing the columns:

- from\_code
- to\_code
- to\_name (if include\_to\_name==TRUE)

**Examples**

```
csdata::nor_locations_hierarchy_from_to(from="wardoslo", to="county")
csdata::nor_locations_hierarchy_from_to(from="municip", to="baregion")
```

---

nor\_locations\_names    *All names in Norway*

---

**Description**

All names in Norway

**Usage**

```
nor_locations_names(border = csdata::config$border_nor)
```

**Arguments**

border	The year in which Norwegian geographical boundaries were designated (2020, 2024).
--------	---

**Value**

**location\_code** Location code.

**location\_name** Location name.

**location\_name\_short** 3 letter location name for nation and county. A shorter location name for wardoslo and extrawardoslo.

**location\_name\_description\_nb** Location name with additional description.

**location\_name\_file\_nb\_utf** Location name that should be used in file names, with Norwegian characters.

**location\_name\_file\_nb\_ascii** Location name that should be used in file names, without Norwegian characters.

**location\_order** The preferred presentation order.

**granularity\_geo** nation, county, municip, wardoslo, wardbergen, wardstavanger, wardtrondheim, baregion, lab.

**Source**

[https://no.wikipedia.org/wiki/Liste\\_over\\_norske\\_kommunenummer](https://no.wikipedia.org/wiki/Liste_over_norske_kommunenummer)

**Examples**

```
nor_locations_names()
```

---

```
nor_locations_redistricting
```

*All redistricting in Norway*

---

**Description**

This function returns a dataset that is used to transfer "original" datasets to the 2020 or 2024 borders.

**Usage**

```
nor_locations_redistricting(border = csdata::config$border_nor)
```

**Arguments**

**border** The year in which Norwegian geographical boundaries were designated (2020, 2024).

**Value**

**location\_code\_current** The location code per today.

**location\_code\_original** The location code as of "calyear".

**calyear** The year corresponding to "county\_code\_original".

**weighting** The weighting that needs to be applied.

**granularity\_geo** nation, county, municip, wardbergen, wardoslo, wardstavanger, wardtrondheim, missingwardbergen, missingwardoslo, missingwardstavanger, missingwardtrondheim, notmainlandcounty, notmainlandmunicip, missingcounty

**Examples**

```
csdata::nor_locations_redistricting()
```

---

```
nor_population_by_age_cats
```

*Population in Norway by categories*

---

**Description**

A function that easily categorizes the Norwegian population into different age categories.

**Usage**

```
nor_population_by_age_cats(
  cats = NULL,
  include_total = TRUE,
  include_9999 = FALSE,
  border = csdata::config$border_nor
)
```

**Arguments**

<b>cats</b>	A list containing vectors that you want to categorize.
<b>include_total</b>	Boolean. Should 'total' be included as an age cat?
<b>include_9999</b>	Boolean. Should the current year is duplicated and added as "calyear==9999". This is in accordance with the cstydy principles regarding granularity_time=="event_*".
<b>border</b>	The year in which Norwegian geographical boundaries were designated (2020, 2024).

**Value**

A data.table containing the following columns:

- granularity\_geo
- location\_code
- age (as specified in the argument "cats")
- sex ("total")
- calyear
- pop\_jan1\_n
- imputed

**Examples**

```
## Not run:  
nor_population_by_age_cats(cats = list(c(1:10), c(11:20)))  
  
## End(Not run)
```

---

se

*Swedish characters in unicode*

---

**Description**

Swedish characters in unicode

**Usage**

```
se
```

**Format**

An object of class list of length 4.

**Examples**

```
print(csdata::se)
```

---

set_config	<i>Set options in the package config</i>
------------	--

---

**Description**

Set options in the package config

**Usage**

```
set_config(border_nor = NULL)
```

**Arguments**

border_nor	The year in which Norwegian geographical boundaries were designated. Valid values: 2020, 2024.
------------	--

**Value**

Nothing. Side effect of setting the config environment.

# Index

## \* datasets

config, [3](#)

nb, [5](#)

se, [9](#)

add\_granularity\_geo\_to\_data\_set, [2](#)

add\_iso3\_to\_data\_set, [3](#)

config, [3](#)

location\_code\_to\_granularity\_geo, [4](#)

location\_code\_to\_iso3, [4](#)

nb, [5](#)

nor\_locations\_hierarchy\_from\_to, [5](#)

nor\_locations\_names, [6](#)

nor\_locations\_redistricting, [7](#)

nor\_population\_by\_age\_cats, [8](#)

se, [9](#)

set\_config, [10](#)