Package: csdata (via r-universe)

July 3, 2026

Title Structural Data for Norway

Version 2026.7.2

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, ncdf4, purrr, readxl, reshape2, rmarkdown, rmapshaper, rstudioapi, stringr, sp, sf, tidyr, zoo

VignetteBuilder knitr

Config/Needs/website niphr/cstemplate

Config/roxygen2/version 8.0.0

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 https://cran.r-universe.dev

Date/Publication 2026-07-02 22:40:02 UTC

RemoteUrl https://github.com/cran/csdata

RemoteRef HEAD

RemoteSha e56b0c2aa16dc7c141a79efcab01d33cacb00192

Contents

`add_granularity_geo_to_data_set` *Add a granularity_geo column to a data set*

Description

Derives the geographic granularity label from the `location_code` column and adds it as a new `granularity_geo` column, modifying `x` in place. When `location_reference` is `NULL` the granularity is inferred from the location code prefix; when a reference table is supplied, it is looked up directly.

Usage

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

Arguments

`x`	A data.table containing a column named `location_code`.
`location_reference`	A data.table with columns `location_code` and `granularity_geo` to use for lookup. When `NULL` (default), granularity is derived from the location code prefix (e.g. `"county_nor03"` -> `"county"`).

Value

`x`, invisibly, with the `granularity_geo` column added or updated.

Examples

```
library(data.table)
data <- data.table(location_code = c("nation_nor", "county_nor03", "blah"))
csdata::add_granularity_geo_to_data_set(data)
print(data)

library(data.table)
data <- data.table(location_code = c("nation_nor", "county_nor03", "blah"))
csdata::add_granularity_geo_to_data_set(data, location_reference = csdata::nor_locations_names())
print(data)
```

---

add_iso3_to_data_set *Add an iso3 column to a data set*

Description

Derives the ISO 3166-1 alpha-3 country code from the `location_code` column and adds it as a new column, modifying `x` in place. Currently all Norwegian location codes map to `"nor"`.

Usage

```
add_iso3_to_data_set(x)
```

Arguments

`x`	A data.table containing a column named `location_code`.

Value

`x`, invisibly, with a new `granularity_geo` column containing the ISO 3166-1 alpha-3 country code (always `"nor"` for Norwegian locations).

Examples

```
library(data.table)
data <- data.table(location_code = c("nation_nor", "county_nor03", "blah"))
csdata::add_iso3_to_data_set(data)
print(data)
```

config *Package configuration environment*

Description

An environment that stores package-level configuration variables. Modify via `set_config()`; read directly (e.g. `config$border_nor`).

Usage

```
config
```

Details

Available variables:

- `border_nor` (default `2024`): the border year used when selecting Norwegian geographic datasets. Valid values: `2024`.

Examples

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

location_code_to_granularity_geo *Convert location codes to granularity_geo*

Description

Extracts the geographic granularity label from one or more location codes. When `location_reference` is `NULL`, the granularity is derived from the lowercase alphabetic prefix of the location code (e.g. `"county_nor03"` -> `"county"`); the special prefix `"norge"` is mapped to `"nation"`. When a reference table is supplied, the granularity is looked up directly.

Usage

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

Arguments

`x`	A character vector of location codes, or a data.table / data.frame containing a column named `location_code`.
`location_reference`	A data.table with columns `location_code` and `granularity_geo` to use for lookup. When `NULL` (default), granularity is inferred from the location code prefix.

Value

A character vector the same length as `x` (or with as many elements as there are rows in `x` when `x` is a data.table), containing the corresponding `granularity_geo` values.

Examples

```
csdata::location_code_to_granularity_geo(c("nation_nor", "county_nor03", "municip_nor0301"))

library(data.table)
dt <- data.table(location_code = c("nation_nor", "county_nor03"))
csdata::location_code_to_granularity_geo(dt)
```

`location_code_to_iso3` *Convert location codes to ISO 3166-1 alpha-3 country codes*

Description

Returns the ISO 3166-1 alpha-3 country code for each location code. Currently all Norwegian location codes map to `"nor"`.

Usage

```
location_code_to_iso3(x)
```

Arguments

`x`	A character vector of location codes, or a data.table / data.frame containing a column named `location_code`.

Value

A character vector the same length as `x` (or with as many elements as there are rows in `x` when `x` is a data.table), containing the corresponding ISO 3166-1 alpha-3 country code (always `"nor"`).

Examples

```
csdata::location_code_to_iso3(c("nation_nor", "county_nor03", "municip_nor0301"))

library(data.table)
dt <- data.table(location_code = c("nation_nor", "county_nor03"))
csdata::location_code_to_iso3(dt)
```

nb *Norwegian characters in unicode*

Description

A named list of Norwegian special characters as unicode strings, for use where literal non-ASCII characters are inconvenient. Elements: AA/aa (Aa/aa), OE/oe (Oe/oe), AE/ae (Ae/ae).

Usage

```
nb
```

Examples

```
print(csdata::nb)
csdata::nb$AA  # uppercase Aa
```

nor_locations_hierarchy_from_to *Location hierarchies in Norway*

Description

Returns a mapping between two geographic levels in Norway. Both from and to accept character vectors, in which case all requested combinations are returned combined into a single data.table.

Usage

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

Arguments

from	Character vector. The source geographic granularity. One or more of: "wardoslo", "extrawardoslo", "wardbergen", "wardtrondheim", "wardstavanger", "missingwardoslo", "missingwardbergen", "missingwardtrondheim", "missingwardstavanger", "municip", "baregion", "county", "georegion", "mtregion", "notmainlandmunicip", "notmainlandcounty", "missingmunicip", "missingcounty".
to	Character vector. The target geographic granularity. Same valid values as from.
include_to_name	Logical. If TRUE, include the name of each to location as a third column to_name. Default FALSE.
border	Integer. The geographic border year. Valid values: 2024. Defaults to csdata::config$border_nor.

Value

A data.table with columns:

from_code Location code at the `from` granularity level.

to_code Location code at the `to` granularity level.

to_name Name of the `to` location (only present when `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")
csdata::nor_locations_hierarchy_from_to(
  from = c("municip", "county"),
  to   = "georegion",
  include_to_name = TRUE
)
```

---

nor_locations_names *Location codes and names for Norwegian geographic units*

Description

Returns a data.table of all Norwegian geographic units with their location codes, display names, and presentation metadata. Coverage includes nation, counties, municipalities, city districts (Oslo, Bergen, Stavanger, Trondheim), BA-regions, and lab regions.

Usage

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

Arguments

`border`	Integer. The geographic border year determining which administrative boundaries are used. Valid values: `2024`. Defaults to `csdata::config$border_nor`.

Value

A data.table with one row per geographic unit and the columns:

location_code Location code (e.g. `"nation_nor"`, `"county_nor03"`).

location_name Full location name.

location_name_short Abbreviated name: 1-letter for nation and county, shorter display name for Oslo and Bergen city districts.

location_name_description_nb Location name with a parenthetical description of geographic level (Norwegian Bokmal).

location_name_file_nb_utf Name suitable for use in file names, retaining Norwegian characters.

location_name_file_nb_ascii Name suitable for use in file names, with Norwegian characters replaced by ASCII equivalents.

location_order Integer giving the preferred presentation order.

granularity_geo Geographic granularity: one of `"nation"`, `"georegion"`, `"mtregion"`, `"county"`, `"municip"`, `"baregion"`, `"wardoslo"`, `"wardbergen"`, `"wardstavanger"`, `"wardtrondheim"`, `"extrawardoslo"`, `"lab"`.

Source

https://no.wikipedia.org/wiki/Liste_over_norske_kommunenummer

Examples

```
d <- nor_locations_names()
head(d)
d[granularity_geo == "county"]
```

---

nor_locations_redistricting

Redistricting weights for Norwegian geographic units

Description

Returns a data.table of weighting factors used to convert historical data recorded under old administrative boundaries to the 2024 borders. Each row maps an original location code (as of a given calendar year) to the current location code, with a proportional weighting.

Usage

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

Arguments

`border`	Integer. The target geographic border year. Valid values: `2024`. Defaults to `csdata::config$border_nor`.

Value

A data.table with columns:

location_code_current Location code under the target border year.

location_code_original Location code as it existed in `calyear`.

calyear The calendar year to which `location_code_original` applies.

weighting Proportional weight to apply when aggregating from the original location to the current location (values sum to 1 within each `location_code_original` / `calyear` group).

granularity_geo Geographic granularity: one of `"nation"`, `"county"`, `"municip"`, `"wardbergen"`, `"wardoslo"`, `"wardstavanger"`, `"wardtrondheim"`, `"missingwardbergen"`, `"missingwardoslo"`, `"missingwardstavanger"`, `"missingwardtrondheim"`, `"notmainlandcounty"`, `"notmainlandmunicip"`, `"missingcounty"`.

Source

Statistics Norway (SSB) municipal reform documentation.

Examples

```
d <- csdata::nor_locations_redistricting()
head(d)
d[calyear == 2019 & granularity_geo == "municip"]
```

`nor_population_by_age_cats`

Norwegian population aggregated into custom age categories

Description

Aggregates the bundled Norwegian population dataset (from Statistics Norway) into caller-defined age bands. The underlying data covers every integer age from 0 to 105 at national, county, municipality, and city-district level.

Usage

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

Arguments

`cats`	A named or unnamed list of integer vectors specifying the age values to include in each category. Each vector element defines one age band. If a list element is named, that name is used as the age label; otherwise the label is auto-generated as `"LLL_HHH"` (zero-padded lower and upper bounds). Defaults to `NULL` (no custom bands; only `"total"` is returned when `include_total = TRUE`).
`include_total`	Logical. If `TRUE` (default), an additional row group with `age = "total"` covering all ages is appended.
`include_9999`	Logical. If `TRUE`, the most recent calendar year is duplicated and added with `calyear = 9999`, following the cstidy convention for `granularity_time == "event_*"`. Default `FALSE`.
`border`	Integer. The geographic border year. Valid values: `2024`. Defaults to `csdata::config$border_nor`.

Value

A data.table with columns:

granularity_geo Geographic granularity level.

location_code Location code.

age Age category label, as derived from `cats` names or auto-generated, plus `"total"` if `include_total = TRUE`.

sex Sex. Always `"total"` in the current dataset.

calyear Calendar year.

pop_jan1_n Population count as of 1 January of `calyear`.

imputed Logical. `TRUE` if the value was imputed.

Examples

```
## Not run:
# Not run: aggregates the full population dataset, which exceeds CRAN's
# example time limit.

# Default: return total population only
d <- nor_population_by_age_cats()
head(d[granularity_geo == "nation"])

# Named age bands
d2 <- nor_population_by_age_cats(
  cats = list("children" = 0:17, "working_age" = 18:66, "elderly" = 67:105),
  include_total = TRUE
)
d2[granularity_geo == "nation" & calyear == 2024]

## End(Not run)
```

---

nor_population_by_sex_age_cats

Population in Norway by sex and age categories

Description

A function that categorizes the Norwegian population into custom age categories, split by sex (male/female) and optionally the total of both sexes.

Usage

```
nor_population_by_sex_age_cats(
  cats = NULL,
  include_total_age = TRUE,
  include_total_sex = TRUE,
  include_9999 = FALSE,
  border = csdata::config$border_nor
)
```

Arguments

cats	A list containing vectors that you want to categorize.
include_total_age	Boolean. Should 'total' be included as an age cat?
include_total_sex	Boolean. Should 'total' (both sexes combined) be included as a sex, alongside 'male' and 'female'?
include_9999	Boolean. Should the current year is duplicated and added as "calyear==9999". This is in accordance with the cstidy principles regarding granularity_time=="event_*".
border	The year in which Norwegian geographical boundaries were designated (2024).

Details

For locations where Statistics Norway provides no sex breakdown (Svalbard, Jan Mayen, unknown), `male` and `female` are returned as `NA` while `total` holds the real count.

Note that when `include_total_sex = TRUE` the output holds `male`, `female` and `total` in long format, so summing `pop_jan1_n` across all sex values double-counts. Filter to a single sex, or sum only the `male`/`female` components.

Value

A data.table containing the following columns:

- granularity_geo
- location_code
- age (as specified in the argument "cats")
- sex ("male", "female", and "total" if include_total_sex)
- calyear
- pop_jan1_n
- imputed

Examples

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

## End(Not run)
```

se *Swedish characters in unicode*

Description

A named list of Swedish special characters as unicode strings, for use where literal non-ASCII characters are inconvenient. Elements: `OE`/`oe` (Oe/oe), `AE`/`ae` (Ae/ae).

Usage

```
se
```

Examples

```
print(csdata::se)
csdata::se$oe  # lowercase oe
```

set_config *Set package configuration options*

Description

Updates one or more variables in the config environment. Call this at the start of a script to change the default border year used by all data-returning functions.

Usage

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

Arguments

`border_nor`	Integer. The Norwegian geographic border year to use as the default. Valid values: `2024`. Pass `NULL` to leave unchanged.

Value

`NULL`, invisibly. Called for the side effect of updating config.

Examples

```
old <- csdata::config$border_nor
csdata::set_config(border_nor = 2024)
csdata::config$border_nor  # 2024
csdata::set_config(border_nor = old)  # restore
```

Index