Package: cstidy (via r-universe)

July 2, 2026

Title Helpful Functions for Cleaning Surveillance Data

Version 2026.7.1

Description Helpful functions for the cleaning and manipulation of surveillance data, especially with regards to the creation and validation of panel data from individual level surveillance data.

Depends R (>= 3.5.0)

License MIT + file LICENSE

URL https://niphr.github.io/cstidy/, https://github.com/niphr/cstidy

BugReports https://github.com/niphr/cstidy/issues

Encoding UTF-8

LazyData true

Imports data.table, magrittr, ggplot2, csdata, cstime, crayon, digest, stringr, methods

Suggests testthat, knitr, rmarkdown, rstudioapi, glue, gt, dplyr, purrr

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.2

VignetteBuilder knitr

Config/Needs/website niphr/cstemplate

Config/testthat/edition 3

Config/pak/sysreqs libicu-dev

Repository https://niphr.r-universe.dev

Date/Publication 2026-07-02 08:44:06 UTC

RemoteUrl https://github.com/niphr/cstidy

RemoteRef HEAD

RemoteSha 5e37fd790db8317860c106042371143cd4338bea

Contents

`expand_time_to` *Expand time to*

Description

Attempts to expand the dataset to include more time

A time series is defined as a unique combination of:

- granularity_time
- granularity_geo
- country_iso3
- location_code
- border
- age
- sex
- *_id
- *_tag

Usage

```
expand_time_to(
  x,
  max_isoyear = NULL,
  max_isoyearweek = NULL,
  max_date = NULL,
  ...
)
```

Arguments

x	An object of type csfmt_rts_data_v2
max_isoyear	Maximum isoyear to expand each isoyear time series up to.
max_isoyearweek	Maximum isoyearweek to expand each isoyearweek time series up to.
max_date	Maximum date to expand each daily time series up to.
...	Not used.

Value

csfmt_rts_data_v2, a larger dataset that includes more rows corresponding to more time.

See Also

Other csfmt_rts_data: identify_data_structure(), remove_class_csfmt_rts_data(), set_csfmt_rts_data_v1(), set_csfmt_rts_data_v2(), set_csfmt_rts_data_v3(), unique_time_series()

Examples

```
x <- cstidy::generate_test_data() %>%
  cstidy::set_csfmt_rts_data_v2()
cstidy::expand_time_to(x, max_isoyearweek = "2022-10")
```

---

generate_test_data *Generate test data*

Description

Generates some test data

Usage

```
generate_test_data(fmt = "csfmt_rts_data_v2")
```

Arguments

fmt	Data format (csfmt_rts_data_v2)

Value

csfmt_rts_data_v2, a dataset containing fake data.

Examples

```
cstidy::generate_test_data("csfmt_rts_data_v2")
```

heal_time_csfmt_rts_data_v1

Provides corresponding healed times (deprecated)

Description

Looks up the time columns (such as isoyear, isoweek, season, and date) that correspond to a vector of dates, isoyearweeks, or isoyears, returning them as a data.table restricted to the requested columns.

Usage

```
heal_time_csfmt_rts_data_v1(x, cols, granularity_time = "date")
```

Arguments

x	A vector containing either dates, isoyearweek, or isoyear.
cols	Columns to restrict the output to.
granularity_time	date, isoyearweek, or isoyear, depending on the values contained in x.

Value

data.table, a dataset with time columns corresponding to the values given in x.

Examples

```
cstidy::heal_time_csfmt_rts_data_v1(
  as.Date(c("2022-01-01", "2022-06-15")),
  cols = c("isoyear", "isoyearweek", "date"),
  granularity_time = "date"
)
```

heal_time_csfmt_rts_data_v2

Provides corresponding healed times

Description

Looks up the time columns (such as isoyear, isoweek, isoquarter, season, and date) that correspond to a vector of dates, isoyearweeks, seasons, or isoyears, returning them as a data.table restricted to the requested columns.

Usage

```
heal_time_csfmt_rts_data_v2(x, cols, granularity_time = "date")
```

Arguments

x	A vector containing dates, isoyearweek, season, or isoyear.
cols	Columns to restrict the output to.
granularity_time	One of "date", "isoyearweek", "season", or "isoyear", matching the values contained in x.

Value

data.table, a dataset with time columns corresponding to the values given in x.

Examples

```
cstidy::heal_time_csfmt_rts_data_v2(
  c("2022-01", "2022-02"),
  cols = c("isoyear", "isoweek", "season", "date"),
  granularity_time = "isoyearweek"
)
```

---

identify_data_structure

Hash the data structure of a dataset for a given column

Description

Summarises the data structure of a single column inside a dataset. For each combination of granularity_time, granularity_geo, age, and sex it records whether the column is structurally missing, only NA, only data, or a mix of data and NA. The result can be passed to `plot()` for a visual overview.

Usage

```
identify_data_structure(x, col, ...)

## S3 method for class 'csfmt_rts_data_v2'
identify_data_structure(x, col, ...)

## S3 method for class '`tbl_Microsoft SQL Server`'
identify_data_structure(x, col, ...)
```

Arguments

x	An object of type csfmt_rts_data_v2.
col	Column name (character) whose data structure is summarised.
...	Arguments passed to or from other methods.

Value

csfmt_rts_data_structure_hash_v2, a summary object that can be plotted.

See Also

Other csfmt_rts_data: `expand_time_to()`, `remove_class_csfmt_rts_data()`, `set_csfmt_rts_data_v1()`, `set_csfmt_rts_data_v2()`, `set_csfmt_rts_data_v3()`, `unique_time_series()`

Examples

```
cstidy::generate_test_data() %>%
  cstidy::set_csfmt_rts_data_v2() %>%
  cstidy::identify_data_structure("deaths_n") %>%
  plot()
```

---

nor_covid19_cases_by_time_location_csfmt_rts_v1

Covid-19 data for PCR-confirmed cases in Norway (nation and county)

Description

This data comes from the Norwegian Surveillance System for Communicable Diseases (MSIS). The date corresponds to when the PCR-test was taken.

Usage

```
nor_covid19_cases_by_time_location_csfmt_rts_v1
```

Format

A csfmt_rts_data_v1 with 11028 rows and 18 variables:

granularity_time day/isoweek

granularity_geo nation, county

country_iso3 nor

location_code norge, 11 counties

border 2020

age total

isoyear Isoyear of event

isoweek Isoweek of event

isoyearweek Isoyearweek of event

season Season of event

seasonweek Seasonweek of event

calyear Calyear of event

calmonth Calmonth of event

calyearmonth Calyearmonth of event

date Date of event

covid19_cases_testdate_n Number of confirmed covid19 cases

covid19_cases_testdate_pr100000 Number of confirmed covid19 cases per 100.000 population

Details

The raw number of cases and cases per 100.000 population are recorded.

This data was extracted on 2022-05-04.

Source

https://github.com/folkehelseinstituttet/surveillance_data/blob/master/covid19/_DOCUMENTATION_data_covid19_msis_by_time_location.txt

Examples

```
head(cstidy::nor_covid19_cases_by_time_location_csfmt_rts_v1)
```

`nor_covid19_icu_and_hospitalization_csfmt_rts_v1`
Norwegian Covid-19 data for ICU and hospitalization

Description

This data was extracted on 2022-05-04.

Usage

```
nor_covid19_icu_and_hospitalization_csfmt_rts_v1
```

Format

A csfmt_rts_data_v1 with 919 rows and 18 variables:

granularity_time day/isoweek

granularity_geo nation

country_iso3 nor

location_code norge

border 2020

age total

isoyear Isoyear of event

isoweek Isoweek of event

isoyearweek Isoyearweek of event

season Season of event

seasonweek Seasonweek of event

calyear Calyear of event

calmonth Calmonth of event

calyearmonth Calyearmonth of event

date Date of event

icu_with_positive_pcr_n Number of new admissions to the ICU with a positive PCR test

hospitalization_with_covid19_as_primary_cause_n Number of new hospitalizations with Covid-19 as the primary cause

Source

https://github.com/folkehelseinstituttet/surveillance_data/blob/master/covid19/_DOCUMENTATION_data_covid19_hospital_by_time.txt

Examples

```
head(cstidy::nor_covid19_icu_and_hospitalization_csfmt_rts_v1)
```

---

remove_class_csfmt_rts_data

*Remove class csfmt_rts_data_**

Description

Remove class csfmt_rts_data_*

Usage

```
remove_class_csfmt_rts_data(x)
```

Arguments

x	data.table

Value

No return value, called for the side effect of removing the csfmt_rts_data class from x.

See Also

Other csfmt_rts_data: `expand_time_to()`, `identify_data_structure()`, `set_csfmt_rts_data_v1()`, `set_csfmt_rts_data_v2()`, `set_csfmt_rts_data_v3()`, `unique_time_series()`

Examples

```
x <- cstidy::generate_test_data() %>%
  cstidy::set_csfmt_rts_data_v2()
class(x)
cstidy::remove_class_csfmt_rts_data(x)
class(x)
```

`set_csfmt_rts_data_v1` *Convert data.table to csfmt_rts_data_v1 (deprecated)*

Description

`set_csfmt_rts_data_v1` converts a `data.table` to `csfmt_rts_data_v1` by reference. `csfmt_rts_data_v1` creates a new `csfmt_rts_data_v1` (not by reference) from either a `data.table` or `data.frame`.

Usage

```
set_csfmt_rts_data_v1(x, create_unified_columns = TRUE, heal = TRUE)

csfmt_rts_data_v1(x, create_unified_columns = TRUE, heal = TRUE)
```

Arguments

`x`	The data.table to be converted to csfmt_rts_data_v1
`create_unified_columns`	Do you want it to create unified columns?
`heal`	Do you want to impute missing values on creation?

Value

An extended `data.table`, which has been modified by reference and returned (invisibly).

No return value, called for side effect of replacing the current data.table with a csfmt_rts_data_v1 in place.

Returns a duplicated csfmt_rts_data_v1.

Smart assignment

`csfmt_rts_data_v1` contains the smart assignment feature for time and geography.

When the **variables in bold** are assigned using `:=`, the listed variables will be automatically imputed.

location_code:

- granularity_geo
- country_iso3

isoyear:

- granularity_time
- isoweek
- isoyearweek
- season
- seasonweek
- calyear
- calmonth
- calyearmonth
- date

isoyearweek:

- granularity_time
- isoyear
- isoweek
- season
- seasonweek
- calyear
- calmonth
- calyearmonth
- date

date:

- granularity_time
- isoyear
- isoweek
- isoyearweek
- season
- seasonweek
- calyear
- calmonth
- calyearmonth

Unified columns

`csfmt_rts_data_v1` contains 16 unified columns:

- granularity_time
- granularity_geo
- country_iso3
- location_code

- border
- age
- sex
- isoyear
- isoweek
- isoyearweek
- season
- seasonweek
- calyear
- calmonth
- calyearmonth
- date

See Also

Other csfmt_rts_data: `expand_time_to()`, `identify_data_structure()`, `remove_class_csfmt_rts_data()`, `set_csfmt_rts_data_v2()`, `set_csfmt_rts_data_v3()`, `unique_time_series()`

`set_csfmt_rts_data_v2` *Convert data.table to csfmt_rts_data_v2*

Description

`set_csfmt_rts_data_v2` converts a `data.table` to `csfmt_rts_data_v2` by reference. `csfmt_rts_data_v2` creates a new `csfmt_rts_data_v2` (not by reference) from either a `data.table` or `data.frame`.

Usage

```
set_csfmt_rts_data_v2(x, create_unified_columns = TRUE, heal = TRUE)

csfmt_rts_data_v2(x, create_unified_columns = TRUE, heal = TRUE)
```

Arguments

`x`	The data.table to be converted to csfmt_rts_data_v2
`create_unified_columns`	Do you want it to create unified columns?
`heal`	Do you want to impute missing values on creation?

Details

For more details see the vignette: `vignette("csfmt_rts_data_v2", package = "cstidy")`

Value

An extended `data.table`, which has been modified by reference and returned (invisibly).

No return value, called for side effect of replacing the current data.table with a csfmt_rts_data_v2 in place.

Returns a duplicated csfmt_rts_data_v2.

Smart assignment

`csfmt_rts_data_v2` contains the smart assignment feature for time and geography.

When the **variables in bold** are assigned using `:=`, the listed variables will be automatically imputed.

location_code:

- granularity_geo
- country_iso3

isoyear:

- granularity_time
- isoweek
- isoyearweek
- isoquarter
- isoyearquarter
- season
- seasonweek
- calyear
- calmonth
- calyearmonth
- date

isoyearweek:

- granularity_time
- isoyear
- isoweek
- isoquarter
- isoyearquarter
- season
- seasonweek
- calyear
- calmonth
- calyearmonth

- date

season:

- granularity_time
- isoyear
- isoweek
- isoyearweek
- isoquarter
- isoyearquarter
- seasonweek
- calyear
- calmonth
- calyearmonth
- date

date:

- granularity_time
- isoyear
- isoweek
- isoyearweek
- isoquarter
- isoyearquarter
- season
- seasonweek
- calyear
- calmonth
- calyearmonth

Unified columns

`csfmt_rts_data_v2` contains 16 unified columns:

- granularity_time
- granularity_geo
- country_iso3
- location_code
- border
- age
- sex
- isoyear

- isoweek
- isoyearweek
- isoquarter
- isoyearquarter
- season
- seasonweek
- calyear
- calmonth
- calyearmonth
- date

See Also

Other csfmt_rts_data: `expand_time_to()`, `identify_data_structure()`, `remove_class_csfmt_rts_data()`, `set_csfmt_rts_data_v1()`, `set_csfmt_rts_data_v3()`, `unique_time_series()`

Examples

```
# Create some fake data as data.table
d <- cstidy::generate_test_data(fmt = "csfmt_rts_data_v2")
d <- d[1:5]

# convert to csfmt_rts_data_v2 by reference
cstidy::set_csfmt_rts_data_v2(d, create_unified_columns = TRUE)

#
d[1, isoyearweek := "2021-01"]
d
d[2, isoyear := 2019]
d
d[3, date := as.Date("2020-01-01")]
d
d[4, c("isoyear", "isoyearweek") := .(2021, "2021-01")]
d
d[5, c("location_code") := .("norge")]
d

# Investigating the data structure of one column inside a dataset
cstidy::generate_test_data() %>%
  cstidy::set_csfmt_rts_data_v2() %>%
  cstidy::identify_data_structure("deaths_n") %>%
  plot()
# Investigating the data structure via summary
cstidy::generate_test_data() %>%
  cstidy::set_csfmt_rts_data_v2() %>%
  summary()
```

set_csfmt_rts_data_v3 *Convert a data.table to csfmt_rts_data_v3 (clean csfmt; explicit healing)*

Description

Same unified columns as `set_csfmt_rts_data_v2`, but without the self-healing `[` override (healing is explicit) and with a content-hash `time_series_id`.

Usage

```
set_csfmt_rts_data_v3(x, create_unified_columns = TRUE, heal = TRUE)

csfmt_rts_data_v3(x, create_unified_columns = TRUE, heal = TRUE)
```

Arguments

`x`	The data.table to convert (by reference).
`create_unified_columns`	Create the unified columns?
`heal`	Impute missing time/geo columns on creation?

Value

x, modified by reference, invisibly.

A new csfmt_rts_data_v3 (not by reference).

See Also

Other csfmt_rts_data: `expand_time_to()`, `identify_data_structure()`, `remove_class_csfmt_rts_data()`, `set_csfmt_rts_data_v1()`, `set_csfmt_rts_data_v2()`, `unique_time_series()`

unique_time_series *Unique time series*

Description

Attempts to identify the unique time series that exist in this dataset.

A time series is defined as a unique combination of:

- granularity_time
- granularity_geo
- country_iso3
- location_code

- border
- age
- sex
- *_id
- *_tag

Usage

```
unique_time_series(x, set_time_series_id = FALSE, ...)
```

Arguments

x	An object of type csfmt_rts_data_v2
set_time_series_id	If TRUE, then x will have a new column called 'time_series_id'
...	Not used.

Value

data.table, a dataset that lists all the unique time series in x.

See Also

Other csfmt_rts_data: expand_time_to(), identify_data_structure(), remove_class_csfmt_rts_data(), set_csfmt_rts_data_v1(), set_csfmt_rts_data_v2(), set_csfmt_rts_data_v3()

Examples

```
x <- cstidy::generate_test_data() %>%
  cstidy::set_csfmt_rts_data_v2()
cstidy::unique_time_series(x)
```

Index