Package 'csquares'

July 5, 2024

Title Concise Spatial Query and Representation System (c-Squares)

Version 0.0.5

Author Pepijn de Vries [aut, cre] (<https://orcid.org/0000-0002-7961-6646>)

Description Encode and decode c-squares, from and to simple feature (sf) or spatiotemporal arrays (stars) objects. Use c-squares codes to quickly join or query spatial data.

Imports dplyr, methods, purrr, rlang, sf, stars, stringr, tidyr, tidyselect, vctrs

Suggests curl, ggplot2, knitr, rmarkdown, testthat (>= 3.0.0)

License GPL (>= 3)

Encoding UTF-8

RoxygenNote 7.2.3

Depends R (>= 4.1.0)

LazyData true

Collate 'helpers.R' 'as_csquares.R' 'as_stars.R' 'csquares-package.R' 'csquares_methods.R' 'drop.R' 'expand.R' 'in.R' 'init.R' 'new_csquares.R' 'orca.R' 'st_as_sf.R' 'summarise.R' 'tidyverse.R' 'validate.R'

Config/testthat/edition 3

URL https://pepijn-devries.github.io/csquares/, https://github.com/pepijn-devries/csquares/

BugReports https://github.com/pepijn-devries/csquares/issues

VignetteBuilder knitr

NeedsCompilation no

Maintainer Pepijn de Vries <pepijn.devries@outlook.com>

Repository CRAN

Date/Publication 2024-07-05 11:20:02 UTC

Contents

`as_csquares` *Convert lon-lat coordinates into c-square codes*

Description

Takes WGS84 longitude and latitude coordinates and finds the closest matching c-squares for a given resolution.

Usage

```
as_csquares(x, resolution, csquares, ...)

## Default S3 method:
as_csquares(x, resolution, csquares, ...)

## S3 method for class 'character'
as_csquares(x, resolution, csquares, validate = TRUE, ...)

## S3 method for class 'numeric'
as_csquares(x, resolution = 1, csquares, ...)

## S3 method for class 'data.frame'
as_csquares(x, resolution = 1, csquares, ...)

## S3 method for class 'sf'
as_csquares(x, resolution = 1, csquares, ..., use_centroids = TRUE)

## S3 method for class 'stars'
as_csquares(x, resolution = 1, csquares, ...)
```

Arguments

x	An object to be coerced to a csquares object. x can be a vector of character strings representing c-squares code. It can also be a numeric matrix with two columns containing the x and y coordinates. x can also be a simple features object (sf) or a spatial arrays object (stars).
resolution	Resolution (in WGS84 degrees) to be used for creating c-squares codes. As per c-square specifications, the resolution should be 10 or less, yet greater than 0. It should be a tenfold of 1 or 5. Valid resolutions are therefore: 10, 5, 1, 0.5, 0.1, etc.
csquares	If x is not a vector of character strings (but for instance a data.frame), the csquares argument should specify the name of the element of x containing the c-square codes as character strings.
...	Currently ignored
validate	A logical value indicating whether the created object needs to be validated. Defaults to TRUE. Validation can be time-consuming so set to FALSE to save computing time.
use_centroids	In case x is a simple features object and use_centroids is TRUE, the centroid of each geometry is used for deriving c-squares. If it is FALSE all coordinates in the geometry are used.

Value

Returns a csquares object that contains c-squares codes.

Author(s)

Pepijn de Vries

Examples

```
as_csquares(cbind(x = 5.2399066, y = 52.7155812), resolution = 1)
orca_csq <- as_csquares(orca, csquares = "csquares")
```

---

drop_csquares *Drop c-square information from object*

Description

Drops c-square data from an object, but keeps the parent class of the object intact. You cannot deselect the csquare column from a csquares object as this will render the object invalid. Use drop_csquares instead.

Usage

```
drop_csquares(x, ...)
```

Arguments

x	An object of class csquares from which the c-square information needs to be dropped.
...	ignored

Value

Returns a copy of x inheriting its parent classes but with out csquares info.

Author(s)

Pepijn de Vries

Examples

```
csq <- as_csquares("1000")
drop_csquares(csq)

csq <-
  data.frame(csquares = "1000", foo = "bar") |>
  as_csquares(csquares = "csquares")

drop_csquares(csq)
```

---

expand_wildcards *Expand c-squares with wildcards to all matching c-squares*

Description

The asterisk (*) can be used as a wildcard, for a compact notation of csquares. expand_wildcards will replace all wild cards with valid combinations of values and expands the compact notation to an explicit notation without wildcards. Check out vignette("Wildcards") for more details.

Usage

```
expand_wildcards(x, ...)
```

Arguments

x	A character string containing csquares codes with wildcards (asterisk character).
...	ignored

Value

Returns a csquares object with full notation

Author(s)

Pepijn de Vries

Examples

```
expand_wildcards("1000:*")
expand_wildcards("1000:***")
expand_wildcards("1000:1**")
expand_wildcards("1000:***:*")
expand_wildcards(c("1000:*", "1000:***", "1000:1**", "1000:***:*"))
```

---

format.csquares *Basic csquares methods*

Description

Basic S3 methods for csquares objects for formatting and printing the objects

Usage

```
## S3 method for class 'csquares'
format(x, ...)

## S3 method for class 'csquares'
print(x, ...)

## S3 method for class 'csquares'
as.character(x, ...)

## S3 method for class 'csquares'
summary(object, ...)

## S3 method for class 'csquares'
as.data.frame(x, ...)
```

Arguments

x, object	A csquares object to be handled by the s3 methods
...	Ignored

Value

Returns (a formatted version of) x

in_csquares *Match c-squares against other c-squares (with wildcards)*

Description

Checks if csquares codes in `table` matches values in `x`. Wildcards are allowed in `table` for this comparison. Check out `vignette("Wildcards")` for more details.

Usage

```
in_csquares(x, table, strict = FALSE, mode = "any", ...)
```

Arguments

`x`	An object of class 'csquares' that will be checked for matching values in `table`
`table`	A `character` string representing a csquares code. The code can contain wildcards (asterisk `*` and percentage `%` characters, both having identical meaning). Any symbol in `x` will result in a positive match against the wildcard. `table` can also be of class `csquares`, but these objects cannot contain wildcards.
`strict`	When set to `FALSE`, a match is positive when the start of `x`, matches against values in `table`, even when `x` has a higher resolution. When set to `TRUE`, a match is only positive when the resolution of `x` and `table` is identical.
`mode`	Two modes are allowed: `"all"` and `"any"`. When an element of `x` consists of multiple raster cells, it the mode will determine whether a match is positive or not. In case of `"all"`, all raster cells in the element of `x` need to match with the cells in `table`, for a positive match. In case of `"any"`, any match will do.
`...`	Ignored

Value

Returns a vector of `logical` values with the same number of elements or rows as `x`

Author(s)

Pepijn de Vries

Examples

```
library(dplyr)

in_csquares(orca$csquares, c("3400:2", "5515:3"))
in_csquares(orca$csquares, "3400:2|5515:3")

## Percentage symbols are interpreted the same as asterisk symbols
## both are wild cards
in_csquares(orca$csquares, "1%%%:%") |>
  table()
```

```
## Same as above
in_csquares(orca$csquares, "1***:*") |>
  table()

## Also same as above
in_csquares(orca$csquares, "1***", strict = FALSE) |>
  table()

## Strict interpretation results in no matches
in_csquares(orca$csquares, "1***", strict = TRUE) |>
  table()

## Filter orca data to North Eastern quadrant (1***:*) only:
orca |>
  filter(
    in_csquares(csquares, "1***:*")
  ) |>
  nrow()
```

`new_csquares` *Create a c-squares raster from a bounding box*

Description

Creates a spatial raster (`stars`) with c-square codes for a specified bounding box, using a specified resolution. The raster will be conform c-squares specifications.

Usage

```
new_csquares(x, resolution = 1, crs = 4326)
```

Arguments

`x`	An object of class `bbox` or an object that can be coerced to a `bbox`. It defines the bounding box for the c-squares grid created by this function.
`resolution`	Resolution (in WGS84 degrees) to be used for creating c-squares codes. As per c-square specifications, the resolution should be 10 or less, yet greater than 0. It should be a tenfold of 1 or 5. Valid resolutions are therefore: 10, 5, 1, 0.5, 0.1, etc.
`crs`	The projection to be used for the created grid. By default it is WGS84 (EPSG:4326).

Value

Returns a `stars` and `csquares` object based on the provided bounding box and resolution.

Author(s)

Pepijn de Vries

Examples

```
library(sf)
nc <- st_read(system.file("shape/nc.shp", package = "sf"))
new_csquares(nc)
```

`orca` *Killer whale realm*

Description

Killer whale realm

Usage

```
orca
```

Format

`orca`**:**

The `orca` object is a Killer whale realm data set extracted from the data as provided by Costello (2017) and published by Costello *et al.* (2017). It is a data frame with 2,058 rows and two columns:

csquares c-squares codes indicating spatial grid cells

orcinus_orca `logical` values indicating whether the corresponding c-squares grid cell belongs to the killer whales (Orcinus orca) biogeographic realm or not.

References

- Costello, M.J. (2017); University of Auckland doi:10.17608/k6.auckland.5086654 Licence CC BY 4.0
- Costello M.J., Tsai P., Wong P.S., Cheung A.K.L, Basher Z. & Chaudhary C. (2017); "Marine biogeographic realms and species endemicity" Nature Communications 8, 1057 doi:10.1038/s41467017011212

st_as_sf *Create a simple features object from c-squares*

Description

Converts a character string of c-squares in a spatially explicit simple features object (sf. It can also convert data.frames with a column of c-squares codes to an sf object.

Usage

```
st_as_sf.csquares(x, ..., use_geometry = TRUE)

st_as_sfc.csquares(x, ..., use_geometry = TRUE)
```

Arguments

x	A vector of character strings. Each element should hold a valid c-square code. x can also be a data.frame with a column of c-square codes. (Note that wildcard characters are not supported)
...	Ignored
use_geometry	If use_geometry is TRUE and x inherits a spatial feature, its geometry will be used to cast the object. This is much faster than its alternative when use_geometry is FALSE. In the latter case, the c-square codes are first translated into explicit spatial information. The latter is more reliable as it does not rely on the assumption that the geometry of x corresponds with the csquares codes in the object. In short: use TRUE for speed, use FALSE for reliability.

Value

In case of st_as_sfc.csquares a list of geometries (sfc, (MULTI)POLYGONS) is returned. In case of st_as_sf.csquares an object of class (sf) is returned.

Author(s)

Pepijn de Vries

Examples

```
library(sf)
st_as_sfc(as_csquares("7500:110:3|7500:110:1|1500:110:3|1500:110:1"))
st_as_sf(as_csquares("7500:110:3|7500:110:1|1500:110:3|1500:110:1"))
```

st_as_stars.csquares *Coerce csqaures object into a stars object*

Description

Take a csquares object created with new_csquares or as_csquares and coerce it to a spatiotemporal array (stars).

Usage

```
st_as_stars.csquares(x, ...)
```

Arguments

x	An object of class csquares created with new_csquares or as_csquares
...	ignored.

Value

Returns a spatiotemporal array (stars) object based on x.

Author(s)

Pepijn de Vries

Examples

```
library(stars)
st_as_stars(as_csquares("7500:110:3|7500:110:1|1500:110:3|1500:110:1"))
st_as_stars(as_csquares(orca, csquares = "csquares"))
```

summarise *Summarise c-square data to a lower resolution*

Description

This function acts very much like dplyr::summarise, but instead of using a column to group, c-square codes are used to aggregate to a lower resolution and summarise the data to that lower resolution.

Usage

```
summarise.csquares(x, ..., .by, tiers_down = 1L)
```

Arguments

x	The object to be summarised. Can be a data.frame, sf, or stars object.
...	<data-masking> Name-value pairs of summary functions. The name will be the name of the variable in the result. The value can be: • A vector of length 1, e.g. min(x), n(), or sum(is.na(y)). • A data frame, to add multiple columns from a single expression. **[Deprecated]** Returning values with size 0 or >1 was deprecated as of 1.1.0. Please use reframe() for this instead.
.by	The column name that holds the c-squares codes that need to be aggregated.
tiers_down	The number of tiers down from the current resolution to which you wish to summarise. If the current resolution is 5x5 degrees, the tier down would be 10x10 degrees (as is the case in the example below).

Value

Returns the summarised object inheriting its class from x

Author(s)

Pepijn de Vries

Examples

```
library(dplyr)
orca |>
  as_csquares(csquares = "csquares") |>
  summarise(
    .by = "csquares",
    orcinus_orca = any(na.omit(.data$orcinus_orca)))
```

---

validate_csquares *Test if a csquares object is valid*

Description

Tests if a csquares object is correctly specified and can be translated into valid coordinates

Usage

```
validate_csquares(x)
```

Arguments

x	An object of class csquares to be evaluated.

Value

Returns a `logical` value indicating whether the `csquares` object is valid or not.

Author(s)

Pepijn de Vries

Examples

```
validate_csquares(
  as_csquares("7500:110:3|7500:110:1|1500:110:3|1500:110:1")
)
```

Index