

Package: stcvlite (via r-universe)

October 9, 2026

Title Lightweight rsample extension for spatiotemporal cross-validation

Version 0.3.1.9000

Description The main functions provide rset generators for spatiotemporal cross-validation with low footprint.

License MIT + file LICENSE

Imports rsample, data.table, FNN, dbscan, terra, sf, methods

Suggests testthat (>= 3.0.0), plotly, sftime, rmarkdown, knitr, gstat, spacetime, xgboost, dbarts, torch

VignetteBuilder knitr

Encoding UTF-8

Roxygen list(markdown = TRUE)

Config/testthat/edition 3

Depends R (>= 4.1)

LazyData true

URL <https://sigmafelix.github.io/stcvlite/>,
<https://github.com/sigmafelix/stcvlite>

BugReports <https://github.com/sigmafelix/stcvlite/issues>

Config/roxygen2/version 8.0.0

Config/pak/sysreqs libabsl-dev cmake libgdal-dev gdal-bin libgeos-dev libicu-dev libssl-dev libproj-dev libsqlite3-dev libudunits2-dev

Repository <https://sigmafelix.r-universe.dev>

Date/Publication 2026-08-10 10:58:50 UTC

RemoteUrl <https://github.com/sigmafelix/stcvlite>

RemoteRef HEAD

RemoteSha f7b981c25ba1b146c57ab2bc251355a04a6e9907

Contents

convert_cv_index_rset	2
generate_block_sp_index	3
generate_cv_index	4
generate_cv_index_lblo	5
generate_cv_index_lblo	6
generate_cv_index_lblo	6
generate_cv_index_lolo	7
generate_cv_index_lolto	8
generate_cv_index_loto	8
generate_cv_index_random	9
generate_spt_index	9
plot_cv_folds	10
prep_input	11
spdat	11
Index	13

convert_cv_index_rset	<i>Generate manual rset object from spatiotemporal cross-validation indices</i>
-----------------------	---

Description

Generate manual rset object from spatiotemporal cross-validation indices

Usage

convert_cv_index_rset(cvindex, data, cv_mode)

Arguments

- cvindex integer. Output of [generate_cv_index](#).
- data data.frame-like object. Should be the same object as what was used for covars argument of [generate_cv_index](#)
- cv_mode character(1). Spatiotemporal cross-validation indexing method. See cv_mode description in [generate_cv_index](#).

Value

rset object of rsample package. A tibble with a list column of training-test data.frames and a column of labels.

Author(s)

Insang Song

generate_block_sp_index

Generate blocked spatial index

Description

Generate blocked spatial index

Usage

```
generate_block_sp_index(covars, cv_fold = NULL, blocks = NULL, block_id = NULL)
```

Arguments

covars	data.frame-like object with lon, lat, and time columns.
cv_fold	integer(1). Number of folds for cross-validation.
blocks	numeric(2)/sf/SpatVector configuration of blocks.
block_id	character(1). The unique identifier of each block.

Details

"Block" in this function refers to a group of contiguous spatial/temporal entities. Regardless of mode value (should be one of "spatial" or "spatiotemporal"), the index column will be named "sp_index". block_id should be set with a proper field name when blocks is a sf or SpatVector object. When cv_fold is an integer, then the *unique* lon/lat coordinates in covars are clustered (not every row), so the result is unaffected by how many times each location was observed — this keeps the block assignment correct for irregular/unbalanced spacetime panels. Location matching when broadcasting the cluster/block id back onto covars is by exact lon/lat equality; coordinates that jitter slightly across repeated observations of "the same" site should be rounded/snapped beforehand. When cv_fold is an integer, the result object will include attributes (accessible with attr function) about "dbscan_eps" and "dbscan_minPts".

Value

data.frame-like object (same class as covars) with a sp_index column

Author(s)

Insang Song

Examples

```
data(spdats)
covars <- spdats
covars_block <- generate_block_sp_index(covars, cv_fold = 5)
covars_block

plot_cv_folds(covars_block, covars_block$sp_index)
```

generate_cv_index	<i>Generate spatio-temporal cross-validation index</i>
-------------------	--

Description

Generate spatio-temporal cross-validation index

Usage

```
generate_cv_index(
  covars,
  cv_mode = c("lolo", "loto", "random", "lblo", "lbto", "lblto", "lolto", "loo"),
  cv_fold = 5L,
  sp_fold = NULL,
  t_fold = NULL,
  blocks = NULL,
  block_id = NULL
)
```

Arguments

covars	data.frame-like object with lon, lat, and time columns. See prep_input() for details.
cv_mode	character(1). One of "lolo" (leave-one-location-out), "loto" (leave-one-time-out), "lolto" (leave-one-location-time-out), "lblo" (leave-block-location-out), "lbto" (leave-block-time-out), "loo", "lblto" (leave-block-location-time-out) "random" (full random selection)
cv_fold	integer(1). Number of folds for cross-validation.
sp_fold	integer(1). Number of subfolds for spatial blocks.
t_fold	integer(1). Number of subfolds for temporal blocks.
blocks	integer(2)/sf/SpatVector object.
block_id	character(1). The unique identifier of each block.

Details

blocks is NULL as default (i.e., no block structure is considered); then "lb*" cv_mode is not working. blocks should be one of integer vector (length 2), sf, sftime, or SpatVector object. Please be advised that we cannot provide error messages related to the blocks type invalidity due to the cyclometric complexity. If any valid object is assigned here, cv_fold will not make sense and be ignored. Definition of unit grid (horizontal, vertical) size, arbitrary shape of blocks (sf/SpatVector case) where users can consider common spatial hierarchies such as states, counties, hydrological unit code areas, etc. lblto mode will accept sp_fold and t_fold. The maximum of results in this case is sp_fold * t_fold.

Value

A numeric vector with length of the number of input rows

Author(s)

Insang Song

Examples

```
data(spdat)

lolo <- generate_cv_index(spdat, cv_mode = "lolo")
loto <- generate_cv_index(spdat, cv_mode = "loto")
lolto <- generate_cv_index(spdat, cv_mode = "lolto")
lblo <- generate_cv_index(spdat, cv_mode = "lblo", cv_fold = 5)
lbto <- generate_cv_index(spdat, cv_mode = "lbto", cv_fold = 5)
lblto <- generate_cv_index(spdat, cv_mode = "lblto", sp_fold = 5, t_fold = 4)
random <- generate_cv_index(spdat, cv_mode = "random", cv_fold = 5)
plot_cv_folds(spdat, lblto)
```

```
generate_cv_index_lblo
```

Generate spatio-temporal cross-validation index (leave-block-location-out)

Description

Generate spatio-temporal cross-validation index (leave-block-location-out)

Usage

```
generate_cv_index_lblo(covars, cv_fold = NULL, blocks = NULL, block_id = NULL)
```

Arguments

<code>covars</code>	data.frame-like object with lon, lat, and time columns. See prep_input() for details.
<code>cv_fold</code>	integer(1). Number of folds for cross-validation.
<code>blocks</code>	integer(2)/sf/SpatVector object.
<code>block_id</code>	character(1). The unique identifier of each block.

Value

An integer vector.

Author(s)

Insang Song

```
generate_cv_index_lblto
```

Generate spatio-temporal cross-validation index (leave-block-location-time-out)

Description

Generate spatio-temporal cross-validation index (leave-block-location-time-out)

Usage

```
generate_cv_index_lblto(covars, sp_fold, t_fold, blocks, block_id = NULL)
```

Arguments

<code>covars</code>	data.frame-like object with lon, lat, and time columns. See prep_input for details.
<code>sp_fold</code>	integer(1). Number of subfolds for spatial blocks.
<code>t_fold</code>	integer(1). Number of subfolds for temporal blocks.
<code>blocks</code>	integer(2)/sf/SpatVector object.
<code>block_id</code>	character(1). The unique identifier of each block.

Details

The maximum of results is `sp_fold * t_fold`.

Value

An integer vector.

Author(s)

Insang Song

```
generate_cv_index_lbto
```

Generate spatio-temporal cross-validation index (leave-block-time-out)

Description

Generate spatio-temporal cross-validation index (leave-block-time-out)

Usage

```
generate_cv_index_lbto(covars, cv_fold = NULL)
```

Arguments

<code>covars</code>	data.frame-like object with <code>lon</code> , <code>lat</code> , and <code>time</code> columns. See prep_input() for details.
<code>cv_fold</code>	<code>integer(1)</code> . Number of folds for cross-validation.

Value

An integer vector.

Author(s)

Insang Song

`generate_cv_index_lolo`

Generate spatio-temporal cross-validation index (leave-one-location-out)

Description

Generate spatio-temporal cross-validation index (leave-one-location-out)

Usage

```
generate_cv_index_lolo(covars)
```

Arguments

<code>covars</code>	data.frame-like object with <code>lon</code> , <code>lat</code> , and <code>time</code> columns. See prep_input() for details.
---------------------	--

Value

An integer vector.

Author(s)

Insang Song

generate_cv_index_lolto

Generate spatio-temporal cross-validation index leave-one-location-time-out)

Description

Generate spatio-temporal cross-validation index leave-one-location-time-out)

Usage

```
generate_cv_index_lolto(covars)
```

Arguments

covars	data.frame-like object with lon, lat, and time columns. See prep_input() for details.
--------	---

Value

An integer vector.

Author(s)

Insang Song

generate_cv_index_loto

Generate spatio-temporal cross-validation index (leave-one-time-out)

Description

Generate spatio-temporal cross-validation index (leave-one-time-out)

Usage

```
generate_cv_index_loto(covars)
```

Arguments

covars	data.frame-like object with lon, lat, and time columns. See prep_input() for details.
--------	---

Value

An integer vector.

Author(s)

Insang Song

generate_cv_index_random*Generate spatio-temporal cross-validation index (random)*

Description

Generate spatio-temporal cross-validation index (random)

Usage

```
generate_cv_index_random(covars, cv_fold = NULL)
```

Arguments

covars	data.frame-like object with lon, lat, and time columns. See prep_input() for details.
cv_fold	integer(1). Number of folds for cross-validation.

Value

An integer vector with unique values of seq(1, cv_fold)

Author(s)

Insang Song

generate_spt_index*Generate unique spatiotemporal identifier from stdt object*

Description

It generates unique spatiotemporal identifier in the input stdt object. Regardless of mode values (should be one of "spatial" or "spatiotemporal"), the index column will be named "sp_index".

Usage

```
generate_spt_index(covars, mode = c("spatial", "spatiotemporal"))
```

Arguments

covars	data.frame-like object with lon, lat, and time columns. See prep_input()
mode	One of "spatial" or "spatiotemporal"

Value

stdt with a new index named "sp_index"

Author(s)

Insang Song

plot_cv_folds

Plot generated CV folds

Description

Creates a plotly scatterplot (x, y, fold index as z) with fold/cluster number shown by color. For large datasets, points are downsampled to at most max_points for display.

Usage

```
plot_cv_folds(covars, cv_index, max_points = 1200L, seed = 2026L)
```

Arguments

covars	A data.frame/data.table (or inheriting object) that contains lon and lat columns.
cv_index	Fold assignment vector returned by generate_cv_index() . Must have length nrow(covars).
max_points	Maximum number of points to display (default 10000).
seed	Random seed used when downsampling.

Value

A plotly 3D scatter object.

Examples

```
data(spdat)
cv_index <-
  generate_cv_index(spdat, cv_mode = "lblo", cv_fold = 6)

if (requireNamespace("plotly", quietly = TRUE)) {
  p <- plot_cv_folds(spdat, cv_index)
  p
}
```

prep_input	<i>Convert sf and sftime objects to stdt format</i>
------------	---

Description

Convert sf and sftime objects to stdt format

Usage

```
prep_input(input)
```

Arguments

input	An sf, sftime, SpatVector or SpatRasterDataset object
-------	---

Value

A data.table object with lon, lat, time and a crs attribute

Author(s)

Eva Marques, Insang Song

spdat	<i>Sample full spatiotemporal point data</i>
-------	--

Description

This dataset contains 12,000 spatiotemporal points with lon, lat, time, and value columns. The points can be used to demonstrate the functions in stcvlite.

Usage

```
spdat
```

Format

A data.table with 12,000 rows (100 locations * 120 times) and six variables, plus a "crs" attribute (attr(spdat, "crs")):

time Unique point identifier. Arbitrarily generated.

lon Longitude

lat Latitude

x1 covariate 1

x2 covariate 2

y response variable

Note

Coordinates are in EPSG:4326

Examples

```
data("spdat", package = "stcvlite")
```

Index

* **Dataset**

spdat, [11](#)

* **datasets**

spdat, [11](#)

convert_cv_index_rset, [2](#)

generate_block_sp_index, [3](#)

generate_cv_index, [2](#), [4](#)

generate_cv_index(), [10](#)

generate_cv_index_lblo, [5](#)

generate_cv_index_lblto, [6](#)

generate_cv_index_lbto, [6](#)

generate_cv_index_lolo, [7](#)

generate_cv_index_lolto, [8](#)

generate_cv_index_loto, [8](#)

generate_cv_index_random, [9](#)

generate_spt_index, [9](#)

plot_cv_folds, [10](#)

prep_input, [6](#), [11](#)

prep_input(), [4](#), [5](#), [7–9](#)

spdat, [11](#)