Package ‘spatialsample’

December 2, 2025

Title Spatial Resampling Infrastructure

Version 0.6.1

Description Functions and classes for spatial resampling to use with the 'rsample' package, such as spatial cross-validation (Brenning, 2012) <doi:10.1109/IGARSS.2012.6352393>. The scope of 'rsample' and 'spatialsample' is to provide the basic building blocks for creating and analyzing resamples of a spatial data set, but neither package includes functions for modeling or computing statistics. The resampled spatial data sets created by 'spatialsample' do not contain much overhead in memory.

License MIT + file LICENSE

URL `https://github.com/tidymodels/spatialsample`, `https://spatialsample.tidymodels.org`

BugReports `https://github.com/tidymodels/spatialsample/issues`

Depends R (>= 3.6)

Imports dplyr (>= 1.0.0), ggplot2, glue, purrr, rlang (>= 1.0.0), rsample (>= 1.1.1), sf (>= 1.0-9), stats, tibble, tidyselect, units, vctrs (>= 0.3.6)

Suggests covr, gifski, knitr, lwgeom, modeldata, rmarkdown, testthat (>= 3.0.0), tidyr, vdiffr, whisker, withr, yardstick

LinkingTo cpp11

VignetteBuilder knitr

Config/Needs/website tidyverse/tidytemplate

Config/testthat/edition 3

Config/testthat/parallel true

Encoding UTF-8

LazyData true

RoxygenNote 7.3.3

NeedsCompilation yes

Author Michael Mahoney [aut, cre] (ORCID:
<https://orcid.org/0000-0003-2402-304X>),
Julia Silge [aut] (ORCID: <https://orcid.org/0000-0002-3671-836X>),
Posit Software, PBC [cph, fnd]

Maintainer Michael Mahoney <mike.mahoney.218@gmail.com>

Repository CRAN

Date/Publication 2025-12-02 06:10:16 UTC

Contents

`autoplot.spatial_rset` *Create a ggplot for spatial resamples.*

Description

This method provides a good visualization method for spatial resampling.

Usage

```
## S3 method for class 'spatial_rset'
autoplot(object, ..., alpha = 0.6)

## S3 method for class 'spatial_block_cv'
autoplot(object, show_grid = TRUE, ..., alpha = 0.6)
```

Arguments

`object`	A `spatial_rset` object or a `spatial_rsplit` object. Note that only resamples made from `sf` objects create `spatial_rset` and `spatial_rsplit` objects; this function will not work for resamples made with non-spatial tibbles or data.frames.
`...`	Options passed to `ggplot2::geom_sf()`.
`alpha`	Opacity, passed to `ggplot2::geom_sf()`. Values of alpha range from 0 to 1, with lower values corresponding to more transparent colors.
`show_grid`	When plotting spatial_block_cv objects, should the grid itself be drawn on top of the data? Set to FALSE to remove the grid.

Details

The plot method for `spatial_rset` displays which fold each observation is assigned to. Note that if data is assigned to multiple folds (which is common if resamples were created with a non-zero `radius`) only the "last" fold for each observation will appear on the plot. Consider adding `ggplot2::facet_wrap(~ fold)` to visualize all members of each fold separately. Alternatively, consider plotting each split using the `spatial_rsplit` method (for example, via `lapply(object$splits, autoplot)`).

Value

A ggplot object with each fold assigned a color, made using `ggplot2::geom_sf()`.

Examples

```
boston_block <- spatial_block_cv(boston_canopy, v = 2)
autoplot(boston_block)
autoplot(boston_block$splits[[1]])
```

`boston_canopy` *Boston tree canopy and heat index data.*

Description

A dataset containing data on tree canopy coverage and change for the city of Boston, Massachusetts from 2014-2019, as well as temperature and heat index data for July 2019. Data is aggregated to a grid of regular 25 hectare hexagons, clipped to city boundaries. This data is made available under the Public Domain Dedication and License v1.0 whose full text can be found at: `https://opendatacommons.org/licenses/pddl/1-0/`.

Usage

```
boston_canopy
```

Format

A data frame (of class `sf`, `tbl_df`, `tbl`, and `data.frame`) containing 682 records of 22 variables:

grid_id Unique identifier for each hexagon. Letters represent the hexagon's X position in the grid (ordered West to East), while numbers represent the Y position (ordered North to South).

land_area Area excluding water bodies

canopy_gain Area of canopy gain between the two years

canopy_loss Area of canopy loss between the two years

canopy_no_change Area of no canopy change between the two years

canopy_area_2014 2014 total canopy area (baseline)

canopy_area_2019 2019 total canopy area

change_canopy_area The change in area of tree canopy between the two years

change_canopy_percentage Relative change calculation used in economics is the gain or loss of tree canopy relative to the earlier time period: (2019 Canopy-2014 Canopy)/(2014 Canopy)

canopy_percentage_2014 2014 canopy percentage

canopy_percentage_2019 2019 canopy percentage

change_canopy_absolute Absolute change. Magnitude of change in percent tree canopy from 2014 to 2019 (% 2019 Canopy - % 2014 Canopy)

mean_temp_morning Mean temperature for July 2019 from 6am - 7am

mean_temp_evening Mean temperature for July 2019 from 7pm - 8pm

mean_temp Mean temperature for July 2019 from 6am - 7am, 3pm - 4pm, and 7pm - 8pm (combined)

mean_heat_index_morning Mean heat index for July 2019 from 6am - 7am

mean_heat_index_evening Mean heat index for July 2019 from 7pm - 8pm

mean_heat_index Mean heat index for July 2019 from 6am - 7am, 3pm - 4pm, and 7pm - 8pm (combined)

geometry Geometry of each hexagon, encoded using EPSG:2249 as a coordinate reference system (NAD83 / Massachusetts Mainland (ftUS)). Note that the linear units of this CRS are in US feet.

Details

Note that this dataset is in the EPSG:2249 (NAD83 / Massachusetts Mainland (ftUS)) coordinate reference system (CRS), which may not be installed by default on your computer. Before working with `boston_canopy`, run:

- `sf::sf_proj_network(TRUE)` to install the CRS itself
- `sf::sf_add_proj_units()` to add US customary units to your units database

These steps only need to be taken once per computer (or per PROJ installation).

Source

Canopy data is from `https://data.boston.gov/dataset/hex-tree-canopy-change-metrics`. Heat data is from `https://data.boston.gov/dataset/hex-mean-heat-index`. Most field definitions are from `https://data.boston.gov/dataset/canopy-change-assessment-data-dictionary`.

`spatial_block_cv` *Spatial block cross-validation*

Description

Block cross-validation splits the area of your data into a number of grid cells, or "blocks", and then assigns all data into folds based on the blocks their centroid falls into.

Usage

```
spatial_block_cv(
  data,
  method = c("random", "snake", "continuous"),
  v = 10,
  relevant_only = TRUE,
  radius = NULL,
  buffer = NULL,
  ...,
  repeats = 1,
  expand_bbox = 1e-05
)
```

Arguments

`data`	An object of class `sf` or `sfc`.
`method`	The method used to sample blocks for cross validation folds. Currently supports `"random"`, which randomly assigns blocks to folds, `"snake"`, which labels the first row of blocks from left to right, then the next from right to left, and repeats from there, and `"continuous"`, which labels each row from left to right, moving from the bottom row up.
`v`	The number of partitions for the resampling. Set to `NULL` or `Inf` for the maximum sensible value (for leave-one-X-out cross-validation).
`relevant_only`	For systematic sampling, should only blocks containing data be included in fold labeling?
`radius`	Numeric: points within this distance of the initially-selected test points will be assigned to the assessment set. If `NULL`, no radius is applied.
`buffer`	Numeric: points within this distance of any point in the test set (after `radius` is applied) will be assigned to neither the analysis or assessment set. If `NULL`, no buffer is applied.
`...`	Arguments passed to `sf::st_make_grid()`.
`repeats`	The number of times to repeat the V-fold partitioning.
`expand_bbox`	A numeric of length 1, representing a proportion to expand the bounding box of `data` by before building a grid. Without this expansion, grids built from data in geographic coordinates may exclude observations and grids built from regularly spaced data might have observations fall exactly on the boundary between folds, duplicating them. In spatialsample < 0.5.0, this was 0.00001 for data in a geographic CRS and 0 for data in a planar CRS. In spatialsample >= 0.5.0, this is 0.00001 for all data.

Details

The grid blocks can be controlled by passing arguments to `sf::st_make_grid()` via `...`. Some particularly useful arguments include:

- `cellsize`: Target cellsize, expressed as the "diameter" (shortest straight-line distance between opposing sides; two times the apothem) of each block, in map units.

- `n`: The number of grid blocks in the x and y direction (columns, rows).
- `square`: A logical value indicating whether to create square (`TRUE`) or hexagonal (`FALSE`) cells.

If both `cellsize` and `n` are provided, then the number of blocks requested by `n` of sizes specified by `cellsize` will be returned, likely not lining up with the bounding box of `data`. If only `cellsize` is provided, this function will return as many blocks of size `cellsize` as fit inside the bounding box of `data`. If only `n` is provided, then `cellsize` will be automatically adjusted to create the requested number of cells.

Value

A tibble with classes `spatial_block_cv`, `spatial_rset`, `rset`, `tbl_df`, `tbl`, and `data.frame`. The results include a column for the data split objects and an identification variable `id`.

References

D. R. Roberts, V. Bahn, S. Ciuti, M. S. Boyce, J. Elith, G. Guillera-Arroita, S. Hauenstein, J. J. Lahoz-Monfort, B. Schröder, W. Thuiller, D. I. Warton, B. A. Wintle, F. Hartig, and C. F. Dormann. "Cross-validation strategies for data with temporal, spatial, hierarchical, or phylogenetic structure," 2016, Ecography 40(8), pp. 913-929, doi: 10.1111/ecog.02881.

Examples

```
spatial_block_cv(boston_canopy, v = 3)
```

spatial_buffer_vfold_cv

V-Fold Cross-Validation with Buffering

Description

V-fold cross-validation (also known as k-fold cross-validation) randomly splits the data into V groups of roughly equal size (called "folds"). A resample of the analysis data consists of V-1 of the folds while the assessment set contains the final fold. These functions extend `rsample::vfold_cv()` and `rsample::group_vfold_cv()` to also apply an inclusion radius and exclusion buffer to the assessment set, ensuring that your analysis data is spatially separated from the assessment set. In basic V-fold cross-validation (i.e. no repeats), the number of resamples is equal to V.

Usage

```
spatial_buffer_vfold_cv(
  data,
  radius,
  buffer,
  v = 10,
  repeats = 1,
  strata = NULL,
  breaks = 4,
  pool = 0.1,
  ...
)

spatial_leave_location_out_cv(
  data,
  group,
  v = NULL,
  radius = NULL,
  buffer = NULL,
  ...,
  repeats = 1
)
```

Arguments

`data`	A data frame.
`radius`	Numeric: points within this distance of the initially-selected test points will be assigned to the assessment set. If `NULL`, no radius is applied.
`buffer`	Numeric: points within this distance of any point in the test set (after `radius` is applied) will be assigned to neither the analysis or assessment set. If `NULL`, no buffer is applied.
`v`	The number of partitions for the resampling. Set to `NULL` or `Inf` for the maximum sensible value (for leave-one-X-out cross-validation).
`repeats`	The number of times to repeat the V-fold partitioning.
`strata`	A variable in `data` (single character or name) used to conduct stratified sampling. When not `NULL`, each resample is created within the stratification variable. Numeric `strata` are binned into quartiles.
`breaks`	A single number giving the number of bins desired to stratify a numeric stratification variable.
`pool`	A proportion of data used to determine if a particular group is too small and should be pooled into another group. We do not recommend decreasing this argument below its default of 0.1 because of the dangers of stratifying groups that are too small.
`...`	These dots are for future extensions and must be empty.
`group`	A variable in data (single character or name) used to create folds. For leave-location-out CV, this should be a variable containing the locations to group observations by, for leave-time-out CV the time blocks to group by, and for leave-location-and-time-out the spatiotemporal blocks to group by.

Details

When `radius` and `buffer` are both `NULL`, `spatial_buffer_vfold_cv` is equivalent to `rsample::vfold_cv()` and `spatial_leave_location_out_cv` is equivalent to `rsample::group_vfold_cv()`.

References

K. Le Rest, D. Pinaud, P. Monestiez, J. Chadoeuf, and C. Bretagnolle. 2014. "Spatial leave-one-out cross-validation for variable selection in the presence of spatial autocorrelation," Global Ecology and Biogeography 23, pp. 811-820, doi: 10.1111/geb.12161.

H. Meyer, C. Reudenbach, T. Hengl, M. Katurji, and T. Nauss. 2018. "Improving performance of spatio-temporal machine learning models using forward feature selection and target-oriented validation," Environmental Modelling & Software 101, pp. 1-9, doi: 10.1016/j.envsoft.2017.12.001.

Examples

```
data(Smithsonian, package = "modeldata")
Smithsonian_sf <- sf::st_as_sf(
  Smithsonian,
  coords = c("longitude", "latitude"),
  crs = 4326
)

spatial_buffer_vfold_cv(
  Smithsonian_sf,
  buffer = 500,
  radius = NULL
)

data(ames, package = "modeldata")
ames_sf <- sf::st_as_sf(ames, coords = c("Longitude", "Latitude"), crs = 4326)
ames_neighborhoods <- spatial_leave_location_out_cv(ames_sf, Neighborhood)
```

`spatial_clustering_cv` *Spatial Clustering Cross-Validation*

Description

Spatial clustering cross-validation splits the data into V groups of disjointed sets by clustering points based on their spatial coordinates. A resample of the analysis data consists of V-1 of the folds/clusters while the assessment set contains the final fold/cluster.

Usage

```
spatial_clustering_cv(
  data,
  v = 10,
  cluster_function = c("kmeans", "hclust"),
  radius = NULL,
  buffer = NULL,
  ...,
  repeats = 1,
  distance_function = function(x) as.dist(sf::st_distance(x))
)
```

Arguments

`data`	An `sf` object (often from `sf::read_sf()` or `sf::st_as_sf()`) to split into folds.
`v`	The number of partitions of the data set.
`cluster_function`	Which function should be used for clustering? Options are either `"kmeans"` (to use `stats::kmeans()`) or `"hclust"` (to use `stats::hclust()`). You can also provide your own function; see `Details`.
`radius`	Numeric: points within this distance of the initially-selected test points will be assigned to the assessment set. If `NULL`, no radius is applied.
`buffer`	Numeric: points within this distance of any point in the test set (after `radius` is applied) will be assigned to neither the analysis or assessment set. If `NULL`, no buffer is applied.
`...`	Extra arguments passed on to `stats::kmeans()` or `stats::hclust()`.
`repeats`	The number of times to repeat the clustered partitioning.
`distance_function`	Which function should be used for distance calculations? Defaults to `sf::st_distance()`, with the output matrix converted to a `stats::dist()` object. You can also provide your own function; see Details.

Details

Clusters are created based on the distances between observations if `data` is an `sf` object. Each cluster is used as a fold for cross-validation. Depending on how the data are distributed spatially, there may not be an equal number of observations in each fold.

You can optionally provide a custom function to `distance_function`. The function should take an `sf` object and return a `stats::dist()` object with distances between data points.

You can optionally provide a custom function to `cluster_function`. The function must take three arguments:

- `dists`, a `stats::dist()` object with distances between data points
- `v`, a length-1 numeric for the number of folds to create
- `...`, to pass any additional named arguments to your function

The function should return a vector of cluster assignments of length `nrow(data)`, with each element of the vector corresponding to the matching row of the data frame.

Value

A tibble with classes `spatial_clustering_cv`, `spatial_rset`, `rset`, `tbl_df`, `tbl`, and `data.frame`. The results include a column for the data split objects and an identification variable `id`. Resamples created from non-`sf` objects will not have the `spatial_rset` class.

Changes in spatialsample 0.3.0

As of spatialsample version 0.3.0, this function no longer accepts non-`sf` objects as arguments to `data`. In order to perform clustering with non-spatial data, consider using `rsample::clustering_cv()`.

Also as of version 0.3.0, this function now calculates edge-to-edge distance for non-point geometries, in line with the rest of the package. Earlier versions relied upon between-centroid distances.

References

A. Brenning, "Spatial cross-validation and bootstrap for the assessment of prediction rules in remote sensing: The R package sperrorest," 2012 IEEE International Geoscience and Remote Sensing Symposium, Munich, 2012, pp. 5372-5375, doi: 10.1109/IGARSS.2012.6352393.

Examples

```
data(Smithsonian, package = "modeldata")

smithsonian_sf <- sf::st_as_sf(
  Smithsonian,
  coords = c("longitude", "latitude"),
  # Set CRS to WGS84
  crs = 4326
)

# When providing sf objects, coords are inferred automatically
spatial_clustering_cv(smithsonian_sf, v = 5)

# Can use hclust instead:
spatial_clustering_cv(smithsonian_sf, v = 5, cluster_function = "hclust")
```

`spatial_nndm_cv` *Nearest neighbor distance matching (NNDM) cross-validation*

Description

NNDM is a variant of leave-one-out cross-validation which assigns each observation to a single assessment fold, and then attempts to remove data from each analysis fold until the nearest neighbor distance distribution between assessment and analysis folds matches the nearest neighbor distance distribution between training data and the locations a model will be used to predict. Proposed by Milà et al. (2022), this method aims to provide accurate estimates of how well models will perform in the locations they will actually be predicting. This method was originally implemented in the CAST package.

Usage

```
spatial_nndm_cv(
  data,
  prediction_sites,
  ...,
  autocorrelation_range = NULL,
  prediction_sample_size = 1000,
  min_analysis_proportion = 0.5
)
```

Arguments

`data`	An object of class `sf` or `sfc`.
`prediction_sites`	An `sf` or `sfc` object describing the areas to be predicted. If `prediction_sites` are all points, then those points are treated as the intended prediction points when calculating target nearest neighbor distances. If `prediction_sites` is a single (multi-)polygon, then points are sampled from within the boundaries of that polygon. Otherwise, if `prediction_sites` is of length > 1 and not made up of points, then points are sampled from within the bounding box of `prediction_sites` and used as the intended prediction points.
`...`	Additional arguments passed to `sf::st_sample()`. Note that the number of points to sample is controlled by `prediction_sample_size`; trying to pass `size` via `...` will cause an error.
`autocorrelation_range`	A numeric of length 1 representing the landscape autocorrelation range ("phi" in the terminology of Milà et al. (2022)). If `NULL`, the default, the autocorrelation range is assumed to be the distance between the opposite corners of the bounding box of `prediction_sites`.
`prediction_sample_size`	A numeric of length 1: the number of points to sample when `prediction_sites` is not only composed of points. Note that this argument is passed to `size` in `sf::st_sample()`, meaning that no elements of `...` can be named `size`.
`min_analysis_proportion`	The minimum proportion of `data` that must remain after removing points to match nearest neighbor distances. This function will stop removing data from analysis sets once only `min_analysis_proportion` of the original data remains in analysis sets, even if the nearest neighbor distances between analysis and assessment sets are still lower than those between training and prediction locations.

Details

Note that, as a form of leave-one-out cross-validation, this method can be rather slow for larger data (and fitting models to these resamples will be even slower).

Value

A tibble with classes `spatial_nndm_cv`, `spatial_rset`, `rset`, `tbl_df`, `tbl`, and `data.frame`. The results include a column for the data split objects and an identification variable `id`.

References

C. Milà, J. Mateu, E. Pebesma, and H. Meyer. 2022. "Nearest Neighbour Distance Matching Leave-One-Out Cross-Validation for map validation." Methods in Ecology and Evolution 2022:13, pp 1304– 1316. doi: 10.1111/2041-210X.13851.

H. Meyer and E. Pebesma. 2022. "Machine learning-based global maps of ecological variables and the challenge of assessing them." Nature Communications 13, pp 2208. doi: 10.1038/s41467-022-29838-9.

Examples

```
data(ames, package = "modeldata")
ames_sf <- sf::st_as_sf(ames, coords = c("Longitude", "Latitude"), crs = 4326)

# Using a small subset of the data, to make the example run faster:
spatial_nndm_cv(ames_sf[1:100, ], ames_sf[2001:2100, ])
```

Index