Package ‘paar’

March 19, 2026

Type Package

Title Precision Agriculture Data Analysis

Version 1.0.2

Description Precision agriculture spatial data depuration and homogeneous zones (management zone) delineation. The package includes functions that performs protocols for data cleaning management zone delineation and zone comparison; protocols are described in Paccioretti et al., (2020) <doi:10.1016/j.compag.2020.105556>.

License MIT + file LICENSE

Encoding UTF-8

LazyData true

RoxygenNote 7.3.3

Imports data.table, e1071, gstat, sf, spdep, stats

Depends R (>= 2.10)

Suggests testthat, concaveman, units, SpatialPack, stars, knitr, rmarkdown, ggplot2

URL https://ppaccioretti.github.io/paar/, https://github.com/PPaccioretti/paar

VignetteBuilder knitr, rmarkdown

BugReports https://github.com/PPaccioretti/paar/issues

NeedsCompilation no

Author Pablo Paccioretti [aut, cre, cph],
Mariano Córdoba [aut],
Franca Giannini-Kurina [aut],
Mónica Balzarini [aut]

Maintainer Pablo Paccioretti <pablopaccioretti@agro.unc.edu.ar>

Repository CRAN

Date/Publication 2026-03-19 20:40:02 UTC

Contents

`barley` *Barley grain yield*

Description

A dataset containing Barley grain yield using calibrated commercial yield monitors mounted on combines equipped with DGPS.

Usage

```
barley
```

Format

A data frame with 7395 rows and 3 variables:

X X coordinate, in meters

Y Y coordinate, in meters

Yield grain yield, in ton per hectare

Details

Coordinate reference system is "WGS 84 / UTM zone 20S", epsg:32720

bind *Bind outlier condition to an object.*

Description

Bind outlier condition to an object.

Usage

```
## S3 method for class 'paar'
cbind(..., deparse.level = 1)
```

Arguments

`...`	objects to bind.
`deparse.level`	integer controlling the construction of labels in the case of non-matrix-like arguments (for the default method): `deparse.level = 0` constructs no labels; the default `deparse.level = 1` typically and `deparse.level = 2` always construct labels from the argument names, see the 'Value' section below.

Value

`cbind` called with m.

compare_zone *Compare means between spatial zones*

Description

Compares variable means across spatial zones using a spatially-adjusted least significant difference (LSD) approach based on kriging variance.

The function accounts for spatial variability by estimating semivariograms and deriving a spatial variance component, which is then used to assess differences between zone means.

Usage

```
compare_zone(
  data,
  variable,
  zonesCol,
  alpha = 0.05,
  join = sf::st_nearest_feature,
  returnLSD = FALSE,
  grid_dim
)
```

Arguments

`data`	an `sf` object containing the spatial zones
`variable`	either: • a `character` vector with column names in `data`, or • an `sf` object with external variables to be compared. In this case, values are spatially joined to `data`.
`zonesCol`	`character`. Column name in `data` defining zones
`alpha`	`numeric`. Significance level for mean comparison
`join`	function used in `sf::st_join` when `variable` is an external `sf` object (default: `sf::st_nearest_feature`)
`returnLSD`	`logical`. If `TRUE`, returns the LSD value used for comparisons
`grid_dim`	`numeric`. Grid resolution used to estimate spatial variance when interpolating external variables. If missing, it is automatically determined.

Details

When `variable` is an external `sf` object, values are interpolated using ordinary kriging before comparison. Otherwise, cross-validation of the variogram model is used to estimate spatial variance.

Pairwise comparisons between zones are evaluated using a spatially-adjusted LSD criterion:

$$LSD = z_{1-\alpha/2} \times \sigma_{spatial}$$

where $\sigma_{spatial}$ is derived from kriging variance.

Results are presented using compact letter displays to indicate groups of zones that are not significantly different.

Value

A list with:

differences list of data frames with mean comparisons per variable

descriptive_stat data frame with descriptive statistics and spatial variance

References

Paccioretti, P., Córdoba, M., & Balzarini, M. (2020). FastMapping: Software to create field maps and identify management zones in precision agriculture. *Computers and Electronics in Agriculture*, 175, 105556. doi:10.1016/j.compag.2020.105556

Examples

```
library(sf)
data(wheat, package = "paar")

##Convert to an sf object
wheat <- sf::st_as_sf(wheat, coords = c("x", "y"), crs = 32720)
```

```
clusters <- paar::kmspc(
  wheat,
  variables = c('CE30', 'CE90', 'Elev', 'Pe', 'Tg'),
  number_cluster = 3:4
)

data_clusters <- cbind(wheat, clusters$cluster)

compare_zone(data_clusters, "Elev", "Cluster_3")
```

depurate *Spatial data depuration (error removal)*

Description

Filters spatial point data by removing erroneous observations based on geometric, statistical, and spatial criteria. The function implements a sequential depuration workflow commonly used in precision agriculture.

Usage

```
depurate(
  x,
  y,
  toremove = c("edges", "outlier", "inlier"),
  crs = NULL,
  buffer = -10,
  ylimitmax = NA,
  ylimitmin = 0,
  sdout = 3,
  ldist = 0,
  udist = 40,
  criteria = c("LM", "MP"),
  zero.policy = NULL,
  poly_border = NULL
)
```

Arguments

`x` An `sf` object with POINT geometries.

`y` A `character` string indicating the variable name used for filtering. If missing and only one attribute column is present, it is used by default.

`toremove` A `character` vector specifying which procedures to apply. Options are `"edges"`, `"outlier"`, and `"inlier"`. The order of execution is fixed and cannot be modified.

`crs` Coordinate reference system used when transforming longitude/latitude data. Can be an EPSG code or proj4string.

`buffer` A `numeric` value indicating the distance (in meters) for edge removal. Negative values are recommended to shrink boundaries.

`ylimitmax` Numeric upper bound for `y`. If `NA`, `Inf` is used.

`ylimitmin` Numeric lower bound for `y`. If `NA`, `-Inf` is used.

`sdout` Numeric multiplier for standard deviation used to detect global outliers.

`ldist` Numeric lower distance bound for neighborhood definition.

`udist` Numeric upper distance bound for neighborhood definition.

`criteria` Character vector specifying spatial outlier detection methods: `"LM"` (Local Moran) and/or `"MP"` (Moran Plot).

`zero.policy` Logical. If `TRUE`, allows empty neighbor sets; if `FALSE`, stops with an error.

`poly_border` Optional `sf` polygon defining field boundaries. If `NULL`, a hull is computed automatically.

Details

The depuration process is applied in a fixed sequence:

1. Edge removal (`"edges"`)
2. Global outlier removal (`"outlier"`)
3. Spatial outlier removal (`"inlier"`)

The `toremove` argument controls which of these steps are applied, but **does not modify the order of execution**.

Available procedures are:

edges Removes points located within a specified `buffer` distance from the field boundary. The boundary is computed using a concave hull (`concaveman`) or a convex hull if the package is not available.

outlier Removes global outliers based on:

- user-defined limits (`ylimitmin`, `ylimitmax`)
- statistical thresholds defined as $mean \pm sdout \times sd$

inlier Identifies and removes spatial outliers using:

- Local Moran's I statistic ("LM")
- Moran scatterplot influence ("MP")

Default parameter values are tuned for precision agriculture datasets (e.g., yield maps).

Value

An object of class `paar` (list) with:

depurated_data Filtered `sf` object

condition Character vector indicating the reason each observation was removed (or `NA` if retained)

References

Vega, A., Córdoba, M., Castro-Franco, M. et al. (2019). Protocol for automating error removal from yield maps. *Precision Agriculture*, 20, 1030–1044. doi:10.1007/s11119018096328

Examples

```
library(sf)
data(barley, package = 'paar')
#Convert to an sf object
barley <- st_as_sf(barley, coords = c("X", "Y"), crs = 32720)

depurated <-
  depurate(barley, "Yield")

# Summary of depurated data
summary(depurated)

# Keep only depurate data
depurated_data <- depurated$depurated_data
# Combine the condition for all data
all_data_condition <- cbind(depurated, barley)
```

`fuzzy_k_means` *Fuzzy k-means clustering (non-spatial)*

Description

Performs fuzzy k-means clustering on tabular data (non-spatial). This function is a lightweight wrapper around `e1071::cmeans`, providing a vectorized workflow and clustering quality indices.

It is primarily intended as a fallback method when spatial clustering (e.g., `kmspc`) cannot be applied, such as when only one variable is available.

Usage

```
fuzzy_k_means(
  data,
  variables,
  number_cluster = 3:5,
  fuzzyness = 1.2,
  distance = "euclidean"
)
```

Arguments

`data` an `sf` object with point geometries

`variables` `character` vector with variable names used for clustering. If missing, all numeric variables in `data` are used.

`number_cluster`	`numeric` vector indicating the number of clusters to evaluate (e.g., `3:5`)
`fuzzyness`	`numeric` value greater than 1 controlling the degree of fuzziness in clustering (see `e1071::cmeans`)
`distance`	`character` distance metric for clustering. One of `"euclidean"` or `"manhattan"` (abbreviations allowed)

Details

Missing values are removed prior to clustering. Observations with missing values are reintroduced in the output with `NA` cluster assignments.

Clustering is performed for each value in `number_cluster`, and several indices are returned to assist in selecting the optimal number of clusters:

- Xie-Beni index
- Partition coefficient
- Partition entropy
- Summary index

Value

A list with:

cluster `data.frame` with cluster assignments for each evaluated number of clusters

indices `data.frame` with clustering validity indices

summaryResults `data.frame` with clustering metrics

See Also

kmspc

Examples

```
library(sf)
data(wheat, package = 'paar')

# Transform the data.frame into a sf object
wheat_sf <- st_as_sf(wheat, coords = c('x', 'y'), crs = 32720)

# Run the fuzzy_k_means function
fuzzy_k_means_results <- fuzzy_k_means(
  wheat_sf,
  variables = 'Tg',
  number_cluster = 2:4
)

# Print the summaryResults
fuzzy_k_means_results$summaryResults

# Print the indices
```

```
fuzzy_k_means_results$indices

# Print the cluster
head(fuzzy_k_means_results$cluster, 5)

# Combine the results in a single object
wheat_clustered <- cbind(wheat_sf, fuzzy_k_means_results$cluster)

# Plot the results
plot(wheat_clustered[, "Cluster_2"])
```

kmspc *Spatial PCA-based fuzzy clustering (MULTISPATI-PCA)*

Description

Performs clustering of spatial data using a combination of spatial Principal Component Analysis (PCA), and fuzzy k-means clustering.

The workflow consists of:

1. Dimensionality reduction using spatial PCA
2. Selection of components based on explained spatial variance
3. Fuzzy clustering over selected components

Usage

```
kmspc(
  data,
  variables,
  number_cluster = 3:5,
  explainedVariance = 70,
  ldist = 0,
  udist = 40,
  center = TRUE,
  fuzzyness = 1.2,
  distance = "euclidean",
  zero.policy = FALSE,
  only_spca_results = TRUE,
  all_results = FALSE
)
```

Arguments

data	an sf object with point geometries
variables	character vector with variable names used for clustering. If missing, all numeric variables in data are used.
number_cluster	numeric vector indicating the number of clusters to evaluate (e.g., 3:5)

`explainedVariance`
: `numeric`. Percentage (0–100) of cumulative explained spatial variance used to select spatial principal components. Values between 0 and 1 are interpreted as proportions.

`ldist`, `udist`
: `numeric`. Lower and upper distance thresholds used to define spatial neighbors.

`center`
: centering option passed to PCA:

  TRUE center variables by their mean

  FALSE no centering

  numeric custom centering vector

`fuzzyness`
: `numeric` value greater than 1 controlling the degree of fuzziness in clustering (see `e1071::cmeans`)

`distance`
: `character` distance metric for clustering. One of `"euclidean"` or `"manhattan"` (abbreviations allowed)

`zero.policy`
: Logical. If `TRUE`, allows empty neighbor sets; if `FALSE`, stops with an error.

`only_spca_results`
: `logical`. If `TRUE`, only spatial PCA results are returned. If `FALSE`, both PCA and spatial PCA summaries are included.

`all_results`
: `logical`. If `TRUE`, full PCA and spatial PCA objects are returned (can increase computation time and memory use).

Details

Spatial relationships are defined using distance-based neighbors (`spdep::dnearneigh`). These relationships are incorporated into the spatial PCA analysis to extract spatially structured components.

Clustering is performed using fuzzy c-means over selected spatial components. Several indices are computed to help determine the optimal number of clusters:

- Xie-Beni index
- Partition coefficient
- Partition entropy
- Summary index (normalized combination)

Value

A list with the following elements:

cluster `data.frame` with cluster assignments for each evaluated number of clusters

indices `data.frame` with clustering validity indices

summaryResults `data.frame` with clustering metrics (iterations, SSDW)

pca_results (optional) PCA and/or spatial PCA summaries depending on arguments

Examples

```
library(sf)
data(wheat, package = 'paar')

# Transform the data.frame into a sf object
wheat_sf <- st_as_sf(wheat, coords = c('x', 'y'), crs = 32720)

# Run the kmspc function
kmspc_results <- kmspc(wheat_sf, number_cluster = 2:4)

# Print the summaryResults
kmspc_results$summaryResults

# Print the indices
kmspc_results$indices

# Print the cluster
head(kmspc_results$cluster, 5)

# Combine the results in a single object
wheat_clustered <- cbind(wheat_sf, kmspc_results$cluster)

# Plot the results
plot(wheat_clustered[, "Cluster_2"])
```

`print.paar` *Print paar objects*

Description

Print paar objects

Usage

```
## S3 method for class 'paar'
print(x, n = 3, ...)
```

Arguments

`x`	an object used to select a method.
`n`	an integer vector specifying maximum number of rows or elements to print.
`...`	further arguments passed to or from other methods.

Value

invisible object x

`print.summary.paar` *Print summarized paar object*

Description

Print summarized paar object

Usage

```
## S3 method for class 'summary.paar'
print(x, digits, ...)
```

Arguments

`x`	an object used to select a method.
`digits`	minimal number of *significant* digits, see `print.default`.
`...`	further arguments passed to or from other methods.

Value

A data.frame with the summarized condition of the object.

`spatial_t_test` *Modified t test*

Description

Performs a modified t-test to assess the correlation between variables while accounting for spatial autocorrelation. This implementation wraps `SpatialPack::modified.ttest`.

Usage

```
spatial_t_test(data, variables)
```

Arguments

`data`	An `sf` object containing geometry and variables, or a `matrix`/`data.frame` with two columns representing spatial coordinates (e.g., X and Y).
`variables`	A `character` vector with the names of the variables to be tested. If `data` is not an `sf` object, this should be a matrix or data.frame of variables to test.

Details

The function computes pairwise correlations between the specified variables and adjusts the significance test to account for spatial dependence using coordinates. If `data` is an `sf` object, coordinates are extracted automatically. Otherwise, coordinates must be provided as an object with two columns.

Value

A `data.frame` with the following columns:

Var1 Name of the first variable

Var2 Name of the second variable

corr Estimated correlation coefficient

p.value P-value adjusted for spatial autocorrelation

See Also

`modified.ttest`

Examples

```
if (requireNamespace("SpatialPack", quietly = TRUE)) {
  library(sf)
  data(wheat, package = 'paar')

  # Transform the data.frame into a sf object
  wheat_sf <- st_as_sf(wheat, coords = c('x', 'y'), crs = 32720)

  # Run spatial t test
  t_test_results <-
    spatial_t_test(
      wheat_sf,
      variables = c('CE30', 'CE90')
    )

  # Print the t_test_results
  t_test_results
}
```

`summary.paar` *Summarizing paar objects*

Description

Summarizing paar objects

Usage

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

Arguments

`object`	an object for which a summary is desired.
`...`	additional arguments affecting the summary produced.

Value

An object of class summary.paar (data.frame) with the following columns:

- `condition` a character vector with the final condition.
- `n` a numeric vector with the number of rows for each condition.
- `percentage` a numeric vector with the percentage of rows for each condition.

`wheat` *Database from a production field under continuous agriculture*

Description

A database from a wheat (Triticum aestivum L.) production field (60 ha) under continuous agriculture, located in south-eastern Pampas, Argentina.

Usage

```
wheat
```

Format

A data frame with 5982 rows and 7 variables:

x X coordinate, in meters

y Y coordinate, in meters

CE30 apparent electrical conductivity taken at 0–30 cm

CE90 apparent electrical conductivity taken at 0–90 cm

Elev elevation, in meters

Pe soil depth, in centimeters

Tg wheat grain yield

Details

Coordinate reference system is "WGS 84 / UTM zone 20S", epsg:32720 Wheat grain yield was recorded in 2009 using calibrated commercial yield monitors mounted on combines equipped with DGPS. Soil ECa measurements were taken using Veris 3100 (VERIS technologies enr., Salina, KS, USA). Soil depth was measured using a hydraulic penetrometer on a 30 × 30 m regular grid (Peralta et al., 2015). Re-gridding was performed to obtain values of all variables at each intersection point of a 10 × 10 m grid.

References

N.R. Peralta, J.L. Costa, M. Balzarini, M. Castro Franco, M. Córdoba, D. Bullock Delineation of management zones to improve nitrogen management of wheat Comput. Electron. Agric., 110 (2015), pp. 103-113, 10.1016/j.compag.2014.10.017

Paccioretti, P., Córdoba, M., & Balzarini, M. (2020). FastMapping: Software to create field maps and identify management zones in precision agriculture. Computers and Electronics in Agriculture, 175, 105556.

Index