Package 'paar'

July 30, 2024

Type Package

Title Precision Agriculture Data Analysis

Version 1.0.1

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 7.3.2

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

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 2024-07-30 13:10:02 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 spatial zone means*

Description

Compare spatial zone means

Usage

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

Arguments

data	sf object with zones
variable	character or sf object to use for mean comparison
zonesCol	character colname from data were zone are specified
alpha	numeric Significance level to use for comparison
join	function to use for st_join if variable is sf object
returnLSD	logical when LSD calculates with spatial variance should be returned
grid_dim	numeric grid dimentins to estimate spatial variance

Value

list with differences and descriptive_stat

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 https://doi.org/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 *Remove errors from spatial data*

Description

Data can be filtered by null, edge values, global outliers and spatial outliers or local defective observations. Default values are optimized for precision agricultural data.

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 points object
y	character with the name of the variable to use for depuration/filtering process
toremove	character vector specifying the procedure to implement for errors removal. Default 'edges', 'outlier', 'inlier'. See Details.
crs	coordinate reference system: integer with the EPSG code, or character with proj4string to convert coordinates if x has longitude/latitude data
buffer	numeric distance in meters to be removed. Negative values are recommended
ylimitmax	numeric of length 1 indicating the maximum limit for the y variable. If NA Inf is assumed
ylimitmin	numeric of length 1 indicating the minimum limit for the y variable. If NA -Inf is assumed
sdout	numeric values outside the interval $meansdoutsdout$ values will be removed
ldist	numeric lower distance bound to identify neighbors
udist	numeric upper distance bound to identify neighbors
criteria	character with "LM" and/or "MP" for methods to identify spatial outliers
zero.policy	default NULL, use global option value; if FALSE stop with error for any empty neighbors sets, if TRUE permit the weights list to be formed with zero-length weights vectors
poly_border	sf object with one polygon or NULL. Can be the result of concaveman::concaveman

Details

Possible values for toremove are one or more elements of:

edges All data points for a distance of buffer m from data edges are deleted.

outlier Values that are outside the mean±sdout are removed

inlier Local Moran index of spatial autocorrelation is calculated for each datum as a tool to identify inliers

Value

an object of class `paar` with two elements:

depurated_data `sf` object with the data after the removal process

condition `character` vector with the condition of each observation

References

Vega, A., Córdoba, M., Castro-Franco, M. et al. Protocol for automating error removal from yield maps. Precision Agric 20, 1030–1044 (2019). https://doi.org/10.1007/s11119-018-09632-8

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*

Description

Performs a vectorized fuzzy k-means clustering, this procedure it is not spatial. The function is almost a wrapper of the function cmeans from the package e1071. Is intended to be used when 'KM-sPC' procedure is not possible because data set has only 1 variable.

Usage

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

Arguments

data	sf object
variables	variables to use for clustering, if missing, all numeric variables will be used
number_cluster	numeric vector with number of final clusters
fuzzyness	A number greater than 1 giving the degree of fuzzification.
distance	character Must be one of the following: If "euclidean", the mean square error, if "manhattan", the mean absolute error is computed. Abbreviations are also accepted.

Value

a list with classification results and indices to select best number of clusters.

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 *MULTISPATI-PCA clustering*

Description

MULTISPATI-PCA clustering

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	sf object
variables	variables to use for clustering, if missing, all numeric variables will be used
number_cluster	numeric vector with number of final clusters
explainedVariance	numeric number in percentage of explained variance from PCA analysis to keep and make cluster process
ldist	numeric lower distance bound to identify neighbors
udist	numeric upper distance bound to identify neighbors
center	a logical or numeric value, centring option if TRUE, centring by the mean if FALSE no centring if a numeric vector, its length must be equal to the number of columns of the data frame df and gives the decentring
fuzzyness	A number greater than 1 giving the degree of fuzzification.
distance	character Must be one of the following: If "euclidean", the mean square error, if "manhattan", the mean absolute error is computed. Abbreviations are also accepted.
zero.policy	default NULL, use global option value; if FALSE stop with error for any empty neighbors sets, if TRUE permit the weights list to be formed with zero-length weights vectors
only_spca_results	logical; should return both PCA and sPCA results (FALSE), or only sPCA results (TRUE)? This can be a time consuming process if there are multiple variables.
all_results	logical; should return the results from the sPCA and PCA call?

Value

a list with classification results and indices to select best number of clusters.

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 version of the t test to assess the correlation between spatial processes. See SpatialPack::modified.ttest for details.

Usage

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

Arguments

data	sf data to extract coordinates or two columns matrix or data.frame specifying coordinates.
variables	character vector with column names to perform ttest

Value

a data.frame with the correlation and p-value for each pair of variables

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