Package 'SCDA'

October 22, 2024

Type Package

Title Spatially-Clustered Data Analysis

Version 0.0.2

Description Contains functions for statistical data analysis based on spatially-clustered techniques. The package allows estimating the spatially-clustered spatial regression models presented in Cerqueti, Maranzano \& Mattera (2024), ``Spatially-clustered spatial autoregressive models with application to agricultural market concentration in Europe", arXiv preprint 2407.15874 <doi:10.48550/arXiv.2407.15874>. Specifically, the current release allows the estimation of the spatially-clustered linear regression model (SCLM), the spatially-clustered spatial autoregressive model (SCSAR), the spatially-clustered spatial Durbin model (SCSEM), and the spatially-clustered linear regression model with spatially-lagged exogenous covariates (SCSLX). From release 0.0.2, the library contains functions to estimate spatial clustering based on Adiajacent Matrix K-Means (AMKM) as described in Zhou, Liu \& Zhu (2019), ``Weighted adjacent matrix for K-means clustering", Multimedia Tools and Applications, 78 (23) <doi:10.1007/s11042-019-08009-x>.

License GPL (>= 2)

Imports spatialreg,sp,spdep,utils,rlang,performance,stats,methods,dplyr,sf,NbClust,ggplot2,ggspatial

Depends R (>= 3.5.0)

Suggests tidyverse,

Encoding UTF-8

Language en-US

RoxygenNote 7.3.1

LazyData true

Maintainer Paolo Maranzano <pmaranzano.ricercastatistica@gmail.com>

NeedsCompilation no

Author Paolo Maranzano [aut, cre, cph] (<https://orcid.org/0000-0002-9228-2759>), Raffaele Mattera [aut, cph] (<https://orcid.org/0000-0001-8770-7049>), Camilla Lionetti [aut, cph], Francesco Caccia [aut, cph]

Repository CRAN

Date/Publication 2024-10-22 21:50:10 UTC

Contents

Data2010 2
Data2020 3
Elbow_finder 5
listW 6
SCSR_Estim 6
SCSR_InfoCrit 9
SC_AMKM 12
SpatReg_Extract 14
SpatReg_GoF 15
SpatReg_Perf 16
SpatReg_PseudoR2 16

Index **18**

`Data2010` *Spatial dataset to replicate the results for 2010 from Cerqueti, R., Maranzano, P. & Mattera, R. "Spatially-clustered spatial autoregressive models with application to agricultural market concentration in Europe". arXiv preprints (<https://doi.org/10.48550/arXiv.2407.15874>*

Description

The 'Data_RC_PM_RM_JABES2024' dataset was created merging information from the Eurostat regional database (<https://ec.europa.eu/eurostat/web/regions/database>). it is a spatial dataset to replicate the results for 2010 from Cerqueti, R., Maranzano, P. & Mattera, R. "Spatially-clustered spatial autoregressive models with application to agricultural market concentration in Europe". arXiv preprints (<https://doi.org/10.48550/arXiv.2407.15874>). Data contained in this file refer to the agricultural sector industry for 222 European regions (NUTS-2 classification) for 2010. For more information see the database 'Economic accounts for agriculture by NUTS 2 region' (agr_r_accts, DOI:10.2908/agr_r_accts). The file includes 6 mixed-type objects:

Usage

```
data(Data_RC_PM_RM_JABES2024)
```

Format

`Data2010` is a spatial data frame (sf/data.frame) with 222 rows, 13 variables and a geometry representing the regions' polygons:

Reference year for the data, that is, 2010

Year Year

Year_lab Extended name (English-translated) of the regions

geo Eurostat NUTS-2 code of the regions

Gini_SO Gini index for the standard output of farms and agricultural holdings in each region

GDPPC_PPS2020 Regional per capita GDP measured as Euros PPS 2020

Share_AgroEmp Share of employment in agriculture: relevance of agricultural industry on the regional labor market

HoursWorked_AgroEmp Hours worked per agro-employed: agricultural labor market intensity

GVA_AgroEmp Gross value added per agro-employed: agricultural productivity intensity

GFCF_AgroEmp Investment per agro-employed: propensity to invest according to the economic size

Share_AgroGVA Share of agricultural GVA on total GVA: relevance of agricultural industry on the regional economy

Share_AgroLand Share of agricultural land: relevance of agricultural industry on the regional activities

Alt_mean Average altitude: geography and landscape

HDD Heating degree days (HDD): proxy of temperature and weather conditions

Note

All source data files prepared by Paolo Maranzano (Department of Economics, Management and Statistics, University of Milano-Bicocca, Italy).

Source

Eurostat – Economic accounts for agriculture by NUTS 2 region' (agr_r_accts, DOI:10.2908/agr_r_accts)

`Data2020` *Spatial dataset to replicate the results for 2020 from Cerqueti, R., Maranzano, P. & Mattera, R. "Spatially-clustered spatial autoregressive models with application to agricultural market concentration in Europe". arXiv preprints (<https://doi.org/10.48550/arXiv.2407.15874>)*

Description

The 'Data_RC_PM_RM_JABES2024' dataset was created merging information from the Eurostat regional database (<https://ec.europa.eu/eurostat/web/regions/database>). it is a spatial dataset to replicate the results for 2020 from Cerqueti, R., Maranzano, P. & Mattera, R. "Spatially-clustered spatial autoregressive models with application to agricultural market concentration in Europe". arXiv preprints (<https://doi.org/10.48550/arXiv.2407.15874>). Data contained in this file refer to the agricultural sector industry for 222 European regions (NUTS-2 classification) for 2020. For more information see the database 'Economic accounts for agriculture by NUTS 2 region' (agr_r_accts, DOI:10.2908/agr_r_accts). The file includes 6 mixed-type objects:

Usage

```
data(Data_RC_PM_RM_JABES2024)
```

Format

`Data2020` is a data frame with 222 rows, 13 variables and a geometry representing the regions' polygons:

Reference year for the data, that is, 2020

Year_lab Extended name (English-translated) of the regions

geo Eurostat NUTS-2 code of the regions

Gini_SO Gini index for the standard output of farms and agricultural holdings in each region

GDPPC_PPS2020 Regional per capita GDP measured as Euros PPS 2020

Share_AgroEmp Share of employment in agriculture: relevance of agricultural industry on the regional labor market

HoursWorked_AgroEmp Hours worked per agro-employed: agricultural labor market intensity

GVA_AgroEmp Gross value added per agro-employed: agricultural productivity intensity

GFCF_AgroEmp Investment per agro-employed: propensity to invest according to the economic size

Share_AgroGVA Share of agricultural GVA on total GVA: relevance of agricultural industry on the regional economy

Share_AgroLand Share of agricultural land: relevance of agricultural industry on the regional activities

Alt_mean Average altitude: geography and landscape

HDD Heating degree days (HDD): proxy of temperature and weather conditions

Note

All source data files prepared by Paolo Maranzano (Department of Economics, Management and Statistics, University of Milano-Bicocca, Italy).

Source

Eurostat – Economic accounts for agriculture by NUTS 2 region' (agr_r_accts, DOI:10.2908/agr_r_accts)

`Elbow_finder` *Automatically selects the optimal number of clusters based on elbow criterion.*

Description

Automatically selects the optimal number of clusters (X-axis) based on elbow criterion computed on a metric (Y-axis). Potential metrics are the AIC and the BIC. The function can be applied to any other context in which the objective is to find the optimal X producing an elbow in Y.

Usage

```
Elbow_finder(x, y, Plot = TRUE)
```

Arguments

`x`	Numeric (m x 1) `vector` of integer values (usually, the number of clusters from 1 to G)
`y`	Numeric (m x 1) `vector` of values (usually, the criterion values) associated with the number of groups.
`Plot`	`Logical` value (`TRUE` or `FALSE`). If `Plot = TRUE` a plot of the relationship between x and y is produced. The plot is a scatterplot with connecting lines. A vertical line is depicted in correspondence of the optimal value of x.

Value

Returns the following outputs:

- x_max_dist: optimal value of x (i.e., the x satisfying the elbow rule)
- y_max_dist: optimal value of y (i.e., the y satisfying the elbow rule)
- Scatterplot (non-compulsory) of x and y with connecting lines and vertical line in correspondence of the optimal value of x.

Examples

```
## Compute the Elbow criterion on two generic vectors x and y
x <- 1:10
y <- c(10,9,6,5,4,3,2,1,1,1)
Elbow_finder(x,y,Plot = TRUE)
```

`listW`	*List of 222 spatial weights (style = "W", zero.policy=TRUE) used in Cerqueti, R., Maranzano, P. & Mattera, R. "Spatially-clustered spatial autoregressive models with application to agricultural market concentration in Europe". arXiv preprints (<https://doi.org/10.48550/arXiv.2407.15874>)*

Description

The 'Data_RC_PM_RM_JABES2024' dataset was created merging information from the Eurostat regional database (<https://ec.europa.eu/eurostat/web/regions/database>). it is a spatial dataset to replicate the results for 2020 from Cerqueti, R., Maranzano, P. & Mattera, R. "Spatially-clustered spatial autoregressive models with application to agricultural market concentration in Europe". arXiv preprints (<https://doi.org/10.48550/arXiv.2407.15874>). Data contained in this file refer to the agricultural sector industry for 222 European regions (NUTS-2 classification) for 2020. For more information see the database 'Economic accounts for agriculture by NUTS 2 region' (agr_r_accts, DOI:10.2908/agr_r_accts). The file includes 6 mixed-type objects:

Usage

```
data(Data_RC_PM_RM_JABES2024)
```

Format

`listW` is a list of 222 spatial weights (style = "W", zero.policy=TRUE) for the European NUTS-2 regions

Note

All source data files prepared by Paolo Maranzano (Department of Economics, Management and Statistics, University of Milano-Bicocca, Italy).

Source

Eurostat – GISCO Territorial units for statistics (NUTS) (https://ec.europa.eu/eurostat/web/gisco/geodata/statistical-units/territorial-units-statistics)

`SCSR_Estim`	*Estimate spatially-clustered spatial regression models*

Description

Estimates spatially-clustered spatial regression (SCSR) models, such as the spatially-clustered linear regression model (SCLM), the spatially-clustered spatial autoregressive model (SCSAR), the spatially-clustered spatial durbin model (SCSEM), and the spatially-clustered linear regression model with spatially-lagged exogenous covariates and response variable (SCSLX). Estimation is performed via cluster-wise maximum likelihood as presented in <https://arxiv.org/abs/2407.15874>.

Usage

```
SCSR_Estim(
  Formula,
  Data_sf,
  listW,
  G = 2,
  Phi = 1,
  Type = c("SCLM", "SCSAR", "SCSEM", "SCSLX"),
  CenterVars = FALSE,
  ScaleVars = FALSE,
  Maxitr = 100,
  RelTol = 10^-6,
  AbsTol = 10^-5,
  Verbose = TRUE,
  Seed = 123456789
)
```

Arguments

Formula	a symbolic description of the regression model to be fit. The details of model specification are given for `lm(...)`
Data_sf	A `data.frame` object of class `sf` with n rows (each one corresponding to a location/polygon) and a user-defined number of columns. The data frame must contain the response variable and all the covariates to be used in the model. Also, it must include the `geometry` feature for spatial modelling and representation. Typically, `sf data.frame` are built using the `st_as_sf(...)` command from the `sf` package (see its documentation for details).
listW	`listw` object. It contains the spatial weights for the spatial autoregressive component. Typically, listW is built using the `nb2listw(...)` command from the `spdep` package (see its documentation for details). We suggest to adopt one of matrix styles suggested in the `spdep` package, such as `W` (row-standardized) or `B` (binary). We also suggest to adopt a `zero.policy = TRUE` option to allow the computation of groups/clusters with isolated units. In this regard, we recall that if `zero.policy = FALSE` and `Type = "SCSAR"` causes `SCSR_Estim(...)` to terminate with an error. See package `spatialreg` for details on the `zero.policy` input.
G	Integer value. Number of clusters to be considered. When 'G=1', the pooled regression (no clusterwise) is estimated. Default is 'G = 2'.
Phi	Non-negative (>=0) real value. Spatial penalty parameter. Default is 'Phi = 1'.
Type	Character. Declares which model specification has to be estimated. Admitted strings are: • `"SCLM"` for linear regression model without spatial effects (LM); • `"SCSAR"` for spatial autoregressive (SAR) model; • `"SCSEM"` for linear regression model with spatial autoregressive error term or spatial Durbin model (SEM);

- "SCSLX" for linear regression model with spatially-lagged response variable and covariates (SLX);

CenterVars — Logical value (TRUE or FALSE) stating whether the response variable and the covariates have to be centered around the mean in the iterative algorithm to update memberships and group-wise parameters. Centering is only use in the iterative procedure, while final estimates provided to the user are computed original (i.e., non-centered) variables.

ScaleVars — Logical value (TRUE or FALSE) stating whether the response variable and the covariates have to be scaled with respect to their standard deviation in the iterative algorithm to update memberships and group-wise parameters. Scaling is only used in the iterative procedure, while final estimates provided to the user are computed original (i.e., non-scaled) variables.

Maxitr — Integer value. Maximum number of iterations for the iterative algorithm. Convergence criterion is fixed to ε = 10^(-5).

RelTol — Tolerance for the relative improvement in the log-likelihood (exit criterion) from iteration k to k+1. Default is ε_{Rel} = 10^-6

AbsTol — Tolerance for the absolute improvement in the log-likelihood (exit criterion) from iteration k to k+1. Default is ε_{Abs} = 10^-5

Verbose — Logical value (TRUE or FALSE). Toggle warnings and messages. If verbose = TRUE (default) the function prints on the screen some messages describing the progress of the tasks. If verbose = FALSE any message about the progression is suppressed.

Seed — Integer value. Define the random number generator (RNG) state for random number generation in R. Deafult is seed = 123456789.

Details

The package SCSR computes the spatially-clustered spatial regression models based on the spatialreg package (see <https://cran.r-project.org/web/packages/spatialreg/index.html>). SCSAR model is estimated using the function lagsarlm; SCSEM model is estimated using the function errorsarlm; SCSLX model is estimated using the function lmSLX. SCLM model is estimated using the lm function from package stats. Thus, estimated SCSAR, SCSEM and SCSLX models belong to class Sarlm, while estimated SCLM belongs to class lm. We kindly refer to the package spatialreg for any detail regarding computational aspects (e.g., optimization). Also, we refer to the package spdep for computational details on the spatial weighting matrix via listw2mat(...), nb2listw(...) and nb2mat(...) from the spdep package. For computional details on the spatially-clustered models, we kindly refer to Cerqueti, R., Maranzano, P. & Mattera, R. "Spatially-clustered spatial autoregressive models with application to agricultural market concentration in Europe". arXiv preprints (<doi:10.48550/arXiv.2407.15874>)

Value

A list object containing the following outputs:

- ClusterFitModels: G-dimensional list containing the estimated clustered regression models of class lm or Sarlm
- Beta: (G x p) matrix of cluster-wise or pooled regression coefficients

- Sig: G-dimensional vector of cluster-wise standard deviations
- VCov: (p x p x G) array of cluster-wise variance-covariance matrices of coefficients
- W_g: G-dimensional list containing for the g-th cluster with cardinality n_g a (n_g x n_g) spatial weighting matrix
- listW_g: G-dimensional list containing for the g-th cluster the weights list
- Group: (n x 1) vector of group assignment
- sBeta: (n x p) matrix of location-wise regression coefficients
- sSig: (n x 1) vector of location-wise standard deviations
- MLE: Estimated maximum log-likelihood
- Iter: The number of iteration needed to satisfy the convergence criterion and end up the clustering iterative loop

Examples

```
data(Data_RC_PM_RM_JABES2024, package="SCDA")
SCSAR <- SCSR_Estim(Formula = "Gini_SO ~ GDPPC_PPS2020 + Share_AgroEmp",
                    Data_sf = Data2020, G=3, listW=listW, Type="SCSAR", Phi = 0.50)
SCLM <- SCSR_Estim(Formula = "Gini_SO ~ GDPPC_PPS2020 + Share_AgroEmp",
                   Data_sf = Data2020, G=3, listW=listW, Type="SCLM", Phi = 0.50)
```

`SCSR_InfoCrit` *Automatically select the optimal number of clusters based on likelihood information criteria (i.e., AIC, BIC and HQC) for a given SCSR model.*

Description

Computes the likelihood-based information criteria (i.e, Akaike's IC, Bayesian IC, and Hannan–Quinn IC) for every SCSR model given by the combination of the G and Phi contained in the `G.set` and `Phi.set` inputs and provides the associated likelihood-based information criteria. Given the minimization rule, `SCSR_InfoCrit` automatically identifies the optimal number of clusters for every criterion.

Usage

```
SCSR_InfoCrit(
  Formula,
  Data_sf,
  listW,
  Phi.set = c(0.5, 1),
  G.set = c(2, 3, 4),
  Type = c("SCLM", "SCSAR", "SCSEM", "SCSLX"),
  CenterVars = TRUE,
```

```
  ScaleVars = TRUE,
  Maxitr = 200,
  RelTol = 10^-6,
  AbsTol = 10^-5,
  Verbose = TRUE,
  Seed = 123456789
)
```

Arguments

Formula	a symbolic description of the regression model to be fit. The details of model specification are given for `lm(...)`
Data_sf	A `data.frame` object of class `sf` with n rows (each one corresponding to a location/polygon) and a user-defined number of columns. The data frame must contain the response variable and all the covariates to be used in the model. Also, it must include the `geometry` feature for spatial modelling and representation. Typically, `sf data.frame` are built using the `st_as_sf(...)` command from the `sf` package (see its documentation for details).
listW	`listw` object. It contains the spatial weights for the spatial autoregressive component. Typically, listW is built using the `nb2listw(...)` command from the `spdep` package (see its documentation for details). We suggest to adopt one of matrix styles suggested in the `spdep` package, such as `W` (row-standardized) or `B` (binary). We also suggest to adopt a `zero.policy = TRUE` option to allow the computation of groups/clusters with isolated units. In this regard, we recall that if `zero.policy = FALSE` and `Type = "SCSAR"` causes `SCSR_Estim(...)` to terminate with an error. See package `spatialreg` for details on the `zero.policy` input.
Phi.set	Non-negative (>=0) real-valued vector. Sequence of spatial penalty parameter. Default is `Phi = c(0.50,1)`.
G.set	Integer vector. Sequence of clusters to be considered. Default is `G = c(2,3,4)`.
Type	Character. Declares which model specification has to be estimated. Admitted strings are: • `"SCLM"` for linear regression model without spatial effects (LM); • `"SCSAR"` for spatial autoregressive (SAR) model; • `"SCSEM"` for linear regression model with spatial autoregressive error term or spatial Durbin model (SEM); • `"SCSLX"` for linear regression model with spatially-lagged response variable and covariates (SLX);
CenterVars	`Logical` value (`TRUE` or `FALSE`) stating whether the response variable and the covariates have to be centered around the mean in the iterative algorithm to update memberships and group-wise parameters. Centering is only use in the iterative procedure, while final estimates provided to the user are computed original (i.e., non-centered) variables.
ScaleVars	`Logical` value (`TRUE` or `FALSE`) stating whether the response variable and the covariates have to be scaled with respect to their standard deviation in the iterative algorithm to update memberships and group-wise parameters. Scaling is

only used in the iterative procedure, while final estimates provided to the user are computed original (i.e., non-scaled) variables.

`Maxitr` Integer value. Maximum number of iterations for the iterative algorithm. Convergence criterion is fixed to ε = 10^(-5).

`RelTol` Tolerance for the relative improvement in the log-likelihood (exit criterion) from iteration k to k+1. Default is ε_{Rel} = 10^-6

`AbsTol` Tolerance for the absolute improvement in the log-likelihood (exit criterion) from iteration k to k+1. Default is ε_{Abs} = 10^-5

`Verbose` `Logical` value (`TRUE` or `FALSE`). Toggle warnings and messages. If `verbose = TRUE` (default) the function prints on the screen some messages describing the progress of the tasks. If `verbose = FALSE` any message about the progression is suppressed.

`Seed` Integer value. Define the random number generator (RNG) state for random number generation in R. Deafult is `seed = 123456789`.

Details

Given the vectors G.set = c(2,3,4) and Phi.set = c(0.50,1), the function 'SCSR_InfoCrit' will compute 3x2=6 models, each at a given combination of G and Phi. For computional details on the spatially-clustered models, we kindly refer to Cerqueti, R., Maranzano, P. & Mattera, R. "Spatially-clustered spatial autoregressive models with application to agricultural market concentration in Europe". arXiv preprints (<doi:10.48550/arXiv.2407.15874>)

Value

A list object containing the following outputs:

- IC: a `data.frame` object containing one row for each combination of the supplied vectors G.set and Phi.set and 5 columns (G,Phi,AIC,BIC,HQC).
- OptimPars: a `data.frame` object with 3 rows (criteria) and 2 columns (Parameters) with the optimal combination of G and Phi for every criterion.

Author(s)

Paolo Maranzano <>

Raffaele Mattera <>

Examples

```
data(Data_RC_PM_RM_JABES2024, package="SCDA")
SCSAR_IC <- SCSR_InfoCrit(Formula = "Gini_SO ~ GDPPC_PPS2020 + Share_AgroEmp",
                          Data_sf = Data2020, listW=listW, Type="SCSAR",
                          Maxitr = 100, Phi.set = c(0.50,1), G.set=c(2,3))
```

SC_AMKM *Spatial Clustering for sf data*

Description

Perform spatial clustering using K-means and AMKM (Adjacent Matrix K-Means) algorithms on sf data.

Usage

```
SC_AMKM(
  Data_sf,
  IndexCol,
  Method,
  Distance = "euclidean",
  MinNc = 2,
  MaxNc = 10,
  Metric = "silhouette",
  RidDim = "pca",
  CenterVars = T,
  ScaleVars = T,
  MakePlot = T,
  ExplainedVariance = 0.9,
  KeepCoord = T,
  Seed = 123456789,
  Verbose = T,
  CRS = 4326
)
```

Arguments

Data_sf	A data.frame object of class sf with n rows (each one corresponding to a location) and a user-defined number of columns. It must include the geometry feature for spatial modelling and representation. Typically, sf data.frame are built using the st_as_sf(...) command from the sf package (see its documentation for details).
IndexCol	Integer value. Number of the dataset ID column. If there isn't an ID column IndexCol=0.
Method	Character. Must be one of: 'AMKM' or 'K-means'. If method='AMKM', the Adjacent Matrix K-Means clustering is performed. If method='K-means', K-means clustering is performed.
Distance	Character. The distance measure to be used to compute the dissimilarity matrix. This must be one of: "euclidean", "maximum", "manhattan", "canberra", "binary", "minkowski". By default, distance='euclidean'.
MinNc	Integer value. Minimal number of clusters, between 1 and (number of objects - 1). Default is MinNc=2.

`MaxNc` `Integer` value. Maximal number of clusters, between 2 and (number of objects - 1), greater or equal to MinNc. Default is `MaxNc=10`.

`Metric` `Character`. The validation index to be calculated for the selection of the optimal clustering partition. This should be one of : "kl", "ch", "hartigan", "ccc", "scott", "marriot", "trcovw", "tracew", "friedman", "rubin", "cindex", "db", "silhouette", "duda", "pseudot2", "beale", "ratkowsky", "ball", "ptbiserial", "gap", "frey", "mcclain", "gamma", "gplus", "tau", "dunn", "hubert", "sdindex", "dindex", "sdbw", "all" (all indices except GAP, Gamma, Gplus and Tau), "alllong" (all indices with Gap, Gamma, Gplus and Tau included). Default is `Metric='silhouette'`.

`RidDim` `Character`.The dimensionality reduction method. This should be one of : 'pca' or 'laplacian'. if 'RidDim='pca", a principal component analysis is performed. if 'RidDim='laplacian" the laplacian matrix dimensionality reduction method is performed . Default is `RidDim='pca'`.

`CenterVars` `Logical` value (`TRUE` or `FALSE`) stating whether the features have to be centered around the mean. Default is `TRUE`.

`ScaleVars` `Logical` value (`TRUE` or `FALSE`) stating whether the features have to be scaled with respect to their standard deviation. Default is `TRUE`.

`MakePlot` `Logical` value (`TRUE` or `FALSE`) stating whether the plot must be displayed. Default is `TRUE`.

`ExplainedVariance` `numeric`. cumulate percentage of the variance explained by the eigenvalues of the dimesionality reduction method. Must be between 0 and 1. Default is `ExplainedVariance=0.9`.

`KeepCoord` `Logical` value (`TRUE` or `FALSE`) stating whether the coordinate must be taken into account in K-means algorithm. Available only when 'method='K-means". Default is `TRUE`.

`Seed` `Integer` value. Define the random number generator (RNG) state for random number generation in R. Deafult is `seed = 123456789`.

`Verbose` `Logical` value (`TRUE` or `FALSE`). Toggle warnings and messages. If `verbose = TRUE` (default) the function prints on the screen some messages describing the progress of the tasks. If `verbose = FALSE` any message about the progression is suppressed. Default is `TRUE`.

`CRS` `Integer` value. Coordinate reference system. something suitable as input to st_crs.command from the `sf` package (see its documentation for details). Default is `CRS=4326`

Details

AMKM calculations is done decomposing the input dataset in two subset. The first one contains the features while the second one contains the coordinates (longitude and latitude). A dissimilarity matrix is calculated on both subset using the parameter distance for the feature and the Great Circle distance for coordinates. Then an adjacent matrix (n x n) is computed on every dissimilarity matrix using gaussian kernel. To reduce the dimensionality of the adjacent matrix a dimentionality reduction method is necessary (see RidDim param. for more) K-means is applied with no modification at its original algorithm.

Value

A list object containing the following outputs:

- df: n row dataframe with the following columns : ID, Longitude, Latitude and Cluster (the optimal partition)
- plot: Display cluster partition in a map.

Author(s)

Camilla Lionetti <lionetticamilla511@gmail.com>, Francesco Caccia <francesco.caccia2000@gmail.com>

Examples

```
library(sp)
library(sf)
data("meuse")
dati<-meuse
dati<-subset(dati,select=sapply(dati,is.numeric))
dati<-st_as_sf(dati, coords = c("x", "y"),crs =28992)
SC <- SC_AMKM(Data_sf=dati,IndexCol=0, Method="AMKM",MinNc = 5,MaxNc = 5 ,CRS=28992)
```

SpatReg_Extract *Extracts numerical values for the estimated regression parameters (i.e., spatial coefficients, regression coefficients, and residuals variance) for a given spatial regression model of class* `lm` *or* `Sarlm`.

Description

Extracts the numerical values for the regression parameters (i.e., estimated spatial parameters, regression coefficients, and residuals variance) for a given spatial regression model of class `lm` or `Sarlm` as defined in package `spatialreg`. The function can be applied to the output of any SCSR model and contained in the `ClusterFitModels` output of `SCSR_Estim` function.

Usage

```
SpatReg_Extract(SRModel)
```

Arguments

`SRModel`	Estimated spatial or non-spatial regression model of class `lm` or `Sarlm` (see package `spatialreg` for details.)

Value

A named `vector` containing numerical values for the estimated spatial parameters (e.g., ρ in SAR or λ in SEM), regression coefficients, and residuals variance for the input model in `SRModel`.

Examples

```
data(Data_RC_PM_RM_JABES2024, package="SCDA")
SCSAR <- SCSR_Estim(Formula = "Gini_SO ~ GDPPC_PPS2020 + Share_AgroEmp",
                    Data_sf = Data2020, G=3, listW=listW, Type="SCSAR", Phi = 0.50)
SpatReg_Extract(SRModel = SCSAR$ClusterFitModels[[1]])
SpatReg_Extract(SRModel = SCSAR$ClusterFitModels[[2]])
SpatReg_Extract(SRModel = SCSAR$ClusterFitModels[[3]])
```

SpatReg_GoF

Computes a set of goodness-of-fit indices (e.g., likelihood-based information criteria, Wald and LR test, Moran's I statistic) for a given spatial regression model of class `lm` *or* `Sarlm`.

Description

Computes a set of goodness-of-fit indices (e.g., likelihood-based information criteria, Wald and LR test, Moran's I statistic) for a given spatial regression model of class `lm` or `Sarlm` as defined in package `spatialreg`. The function can be applied to the output of any SCSR model and contained in the `ClusterFitModels` output of `SCSR_Estim` function.

Usage

```
SpatReg_GoF(SRModel_list, SRModel_W_list)
```

Arguments

`SRModel_list`	List of estimated spatial or non-spatial regression model of class `lm` or `Sarlm` (see package `spatialreg` for details.)
`SRModel_W_list`	List of `listw` objects (see package `spdep` for details) containing the spatial weights for the spatial autoregressive component for the G groups.

Value

A `matrix` containing 15 goodness-of-fit indices (e.g., likelihood-based information criteria, Wald and LR test, Moran's I statistic) for the list of models given as a input in `SRModel_list`.

Examples

```
data(Data_RC_PM_RM_JABES2024, package="SCDA")
SCSAR <- SCSR_Estim(Formula = "Gini_SO ~ GDPPC_PPS2020 + Share_AgroEmp",
                    Data_sf = Data2020, G=3, listW=listW, Type="SCSAR", Phi = 0.50)
reglist <- c(SCSAR$ClusterFitModels[1],SCSAR$ClusterFitModels[2],SCSAR$ClusterFitModels[3])
Wlist <- c(SCSAR$listW_g[1],SCSAR$listW_g[2],SCSAR$listW_g[3])
SpatReg_GoF(SRModel_list = reglist,SRModel_W_list = Wlist)
```

SpatReg_Perf *Computes a set of in-sample performance metrics (i.e., AIC, BIC, RMSE, Sigma, and Pseudo R^2) for a given spatial regression model of class* lm *or* Sarlm.

Description

Computes a set of in-sample performance metrics (i.e., AIC, BIC, RMSE, Sigma, and Pseudo R^2) for a given spatial regression model of class lm or Sarlm as defined in package spatialreg. The function can be applied to the output of any SCSR model and contained in the ClusterFitModels output of SCSR_Estim function.

Usage

```
SpatReg_Perf(SRModel)
```

Arguments

SRModel	Estimated spatial or non-spatial regression model of class lm or Sarlm (see package spatialreg for details.)

Value

A named vector containing numerical values for the estimated performance metrics (i.e., AIC, BIC, RMSE, Sigma, and Pseudo R^2) for the input model in SRModel.

Examples

```
data(Data_RC_PM_RM_JABES2024, package="SCDA")
SCSAR <- SCSR_Estim(Formula = "Gini_SO ~ GDPPC_PPS2020 + Share_AgroEmp",
                    Data_sf = Data2020, G=3, listW=listW, Type="SCSAR", Phi = 0.50)
SpatReg_Perf(SRModel = SCSAR$ClusterFitModels[[1]])
SpatReg_Perf(SRModel = SCSAR$ClusterFitModels[[2]])
SpatReg_Perf(SRModel = SCSAR$ClusterFitModels[[3]])
```

SpatReg_PseudoR2 *Computes the Pseudo R^2 metric for a given spatial regression model of class* lm *or* Sarlm.

Description

Computes the Pseudo R^2 metric for a given spatial regression model of class lm or Sarlm as defined in package spatialreg. The function can be applied to the output of any SCSR model and contained in the ClusterFitModels output of SCSR_Estim function.

Usage

```
SpatReg_PseudoR2(SRModel)
```

Arguments

SRModel	Estimated spatial or non-spatial regression model of class lm or Sarlm (see package spatialreg for details.)

Value

A numeric value reporting the Pseudo R^2 for the input model in SRModel.

Examples

```
data(Data_RC_PM_RM_JABES2024, package="SCDA")
SCSAR <- SCSR_Estim(Formula = "Gini_SO ~ GDPPC_PPS2020 + Share_AgroEmp",
                    Data_sf = Data2020, G=3, listW=listW, Type="SCSAR", Phi = 0.50)
SpatReg_PseudoR2(SRModel = SCSAR$ClusterFitModels[[1]])
SpatReg_PseudoR2(SRModel = SCSAR$ClusterFitModels[[2]])
SpatReg_PseudoR2(SRModel = SCSAR$ClusterFitModels[[3]])
```

Index

∗ **datasets**
    Data2010, 2
    Data2020, 3
    listW, 6

Data2010, 2
Data2020, 3

Elbow_finder, 5

listW, 6

SC_AMKM, 12
SCSR_Estim, 6
SCSR_InfoCrit, 9
SpatReg_Extract, 14
SpatReg_GoF, 15
SpatReg_Perf, 16
SpatReg_PseudoR2, 16