Package ‘geomerge’

July 31, 2018

Type Package
Title Geospatial Data Integration
Version 0.3.1
Date 2018-07-31
Author Karsten Donnay and Andrew M. Linke
Maintainer Karsten Donnay <kdonnay@gmx.net>
Description Geospatial data integration framework that merges raster, spatial polygon, and (dynamic) spatial points data into a spatial (panel) data frame at any geographical resolution.
License LGPL-3
Depends R (>= 2.10)
Imports inlmisc, raster, spdep, sp, geosphere, lubridate, methods, stats, ggplot2, gridExtra, scales
NeedsCompilation no
Repository CRAN
Date/Publication 2018-07-31 21:20:03 UTC

R topics documented:

geomerge-package ........................................ 2
ACLED .................................................... 3
AidData .................................................... 4
generateGrid .............................................. 5
geoEPR ..................................................... 6
geomerge ................................................... 7
geomerge.assign .......................................... 10
geomerge.merge ........................................... 12
geomerge.neighbor ....................................... 13
gpw ......................................................... 14
plot.geomerge ............................................. 15
print.geomerge ........................................... 16
states ....................................................... 17
summary.geomerge ........................................ 18
Description

geomerge is a framework for geospatial data integration that merges raster, spatial polygon, and (dynamic) spatial points data into a spatial (panel) data frame at any geographical resolution.

Details

The geomerge function conducts a series of spatial joins for Geographic Information Systems (GIS) data. It integrates three of R’s most commonly used GIS data classes - polygons, points and rasters. With flexible options for assignment rules and including the calculation of spatial and temporal lags, geomerge returns a time series SpatialPolygonsDataFrame that users may import into any predictive statistical analysis.

Note

The spatial resolution of the input datasets and scope of the area covered by the integration routine will influence the runtime of geomerge. Depending on the inputs, integration may therefore require some time.

Author(s)

Karsten Donnay and Andrew M. Linke

References


See Also

geomerge, geomerge.merge, geomerge.neighbor, geomerge.assign, generateGrid
ACLED

Point dataset to illustrate the functionality of geomerge

Description

ACLED conflict events for Nigeria in 2011 used as example for a SpatialPointsDataFrame available from http://www.acleddata.com/data. The dataset contains timestamped and geo-coded information on individual conflict events.

Usage

data(geomerge)

Format

A SpatialPointsDataFrame containing observations.

Details

The original ACLED "EVENT_DATE" column has been relabeled as "timestamp" in accordance with geomerge conventions.

Author(s)

Karsten Donnay and Andrew M. Linke

Source

http://www.acleddata.com/data

References

Description

AidData aid project locations for projects in Nigeria with start date in 2011 used as example for a SpatialPointsDataFrame. The dataset is available from http://aiddata.org. The dataset contains timestamped and geo-coded information on individual aid projects.

Usage

data(geomerge)

Format

A SpatialPointsDataFrame containing observations.

Details

The original AidData "start_date" column has been relabeled as "timestamp" in accordance with geomerge conventions.

Author(s)

Karsten Donnay and Andrew M. Linke

Source

http://aiddata.org


References

generateGrid

Generates a grid in a given local CRS that is, by default, returned as SpatialPolygonsDataFrame in WGS84.

Description

Implementation of a simple grid generation function producing a SpatialPolygonsDataFrame to be used as target in geomerge.

Usage

generateGrid(extent, size, local.CRS, makeWGS84 = TRUE, silent = FALSE)

Arguments

extent SpatialPolygonsDataFrame that defines the (minimum) extent of the grid to be generated.
size size of the grid cells in m.
local.CRS definition of the local (projected) CRS the grid is spanned in. Has to be class "CRS".
makeWGS84 Boolean switch indicating whether or not the grid is returned in WGS84. Default = TRUE.
silent Boolean switch to suppress any (non-critical) warnings and messages. Default = FALSE.

Value

Returns an object of SpatialPolygonsDataFrame that spans the grid with spatial resolution given by size.

Author(s)

Karsten Donnay and Andrew M. Linke.

References


See Also

geomerge-package, geomerge
Examples

```r
require(sp)
data(geomerge)

# Generate grid with 10 km cell size in local CRS for Nigeria
states.grid <- generateGrid(states, 10000, local.CRS=CRS("+init=epsg:26391"),
   silent=TRUE)
```

---

**geoEPR**

*Polygon dataset to illustrate the functionality of geomerge*

---

**Description**

`geoEPR` Nigeria dataset used as example for a SpatialPolygonsDataFrame can be accessed and downloaded at https://icr.ethz.ch/data/epr/geoepr/. The dataset contains geo-locations for all politically relevant ethnic groups from the EPR-Core 2014 dataset. It assigns every politically relevant group one of six settlement patterns and provides polygons describing their location.

**Usage**

```r
data(geomerge)
```

**Format**

A SpatialPolygonsDataFrame containing observations.

**Author(s)**

Karsten Donnay and Andrew M. Linke

**Source**

https://icr.ethz.ch/data/epr/geoepr/


**References**

**geomerge**  
*Geospatial Data Integration*

**Description**
This function conducts a series of spatial joins for Geographic Information Systems (GIS) data. It integrates three of R’s most commonly used GIS data classes - polygons, points and rasters. With flexible options for assignment rules and including the calculation of spatial and temporal lags, geomerge returns a spatial (panel) dataset in the form of a SpatialPolygonsDataFrame that users may import into any predictive statistical analysis.

**Usage**
```
geomerge(..., target=NULL, time=NA, time.lag=TRUE, spat.lag=TRUE,
        zonal.fun=sum, assignment="max(area)", population.data = NA,
        point.agg = "cnt", t_unit="days", silent=FALSE)
```

**Arguments**
- `...` input datasets and, if provided, optional arguments. See Details.
- `target` SpatialPolygonsDataFrame representing desired units of analysis. See Details.
- `time` temporal window for dynamic temporal binning of point data. Required format is `c(start_date, end_date, interval_length)`, each specified as String. Default = `NA`. See Details.
- `time.lag` Boolean indicating whether or not first and second order temporal lag values of all variables are returned. Only affects dynamic point data integration. Default = `TRUE`.
- `spat.lag` Boolean indicating whether or not first and second order spatial lag values of all variables are returned. Default = `TRUE`.
- `zonal.fun` object of class `function` applied to values of `RasterLayer` when generating zonal statistics for each target polygon. Default = `sum`. See Details.
- `assignment` identification of either population- or area-weighting assignment rules when handling `SpatialPolygonsDataFrame` joins to `target`. Default = "max(area)". See Details.
- `population.data` specifies data used for weighting if a population-based assignment rule is selected. See Details.
- `point.agg` specification of aggregation format for data of type `SpatialPointsDataFrame`. Default = "cnt". See Details.
- `t_unit` temporal unit used for dynamic point aggregation. Default = "days".
- `silent` Boolean switch to suppress any (non-critical) warnings and messages. Default = `FALSE`. 
Details

geomerge accepts any number of data inputs of the most common spatial data classes in R - SpatialPolygonsDataFrame, SpatialPointsDataFrame, and RasterLayer. The target they are merged to may be of any shape but must be a SpatialPolygonsDataFrame. The extent of each data input should at least match the extent of the target; if not, the package returns a warning. In order to perform accurate area calculations at any scale, geomerge projects any data geometry into WGS84. Input data (including target) not in WGS84 are automatically re-projected.

geomerge assumes that all inputs of type SpatialPolygonsDataFrame and RasterLayer are static and contemporary. If polygons or raster are changing, we advise to simply rerun geomerge for each interval in which data are static and contemporary. The package allows for dynamic integration of all inputs that are a SpatialPointsDataFrame, i.e., one can, for example, automatically generate the counts of events that occur within a specific unit of target within a specific time period. Further details are given below.

If SpatialPolygonsDataFrame data are joined to target, they must contain only one column with the data of interest. The package also accepts the short-hand variable specification using the standard "$" notation to denote the selection of a specific variable from the SpatialPolygonsDataFrame. RasterLayer are by default single-valued. These data may be of class factor or numeric.

If SpatialPointsDataFrame are joined to target they must have one column coding the variable of interest and, if points carry timestamps, dates must be given in a second column timestamp and formatted as a UTC date string with format "YYYY-MM-DD" or "YYYY-MM-DD hh:mm:ss".

In practice, our input logic implies that if more than one variable of interest are to be merged to target, statically or dynamically, each has to be separately entered as argument. Note that variable names in target derive from the name of the input data and it is therefore advised to use meaningful labels for input data.

In merging SpatialPolygonsDataFrame values to units of analysis given by target, users have a choice among a number of different assignment rules based on area overlap and population size. Area-based assignment generally can take the values "max(area)" or "min(area)", i.e., the value assigned to a given unit in target comes from that polygon in the SpatialPolygonsDataFrame with maximal or minimal area overlap respectively. If the value of interest is of class numeric, the user may also choose "weighted(area)", i.e., the values is assigned as the area-weighted average of the values in all polygons intersecting a given unit in target.

The assignment rules "max(pop)", "min(pop)" and "weighted(pop)" (the latter again for numeric variables only) analogously use the population value given by population.data in overlapping areas as basis for assignment. If any of them is selected in the assignment argument, users must provide population.data as a RasterLayer. The geographical resolution of population.data should be the same or better than that of target. The zonal statistic used for population within overlapping polygons is sum.

When a SpatialPointsDataFrame is merged to target, one of two operations can be performed. For point.agg = "cnt" the function calculates the sum of the number of locations that fall within each unit of target. For numerical variables of interest, point.agg = "sum" returns the sum across for all values associated with points within each unit of target. If different aggregation formats are to be applied to different SpatialPointsDataFrame inputs, these have to be specified as a character vector, i.e., point.agg = c("sum", "cnt"), in the order of inputs.

Values for inputs of type SpatialPointsDataFrame are either calculated statically across the entire frame if time = NA or dynamically within a given time period that can be specified using
time = c(start_date, end_date, interval_length). All three inputs must be Strings where
interval_length is defined in multiples of t_unit. The default value is t_unit = "days", the
package also accepts inputs of "secs", "mins", "hours", "months" or "years".

Zonal statistics are applied to objects of class RasterLayer that are joined to target. The specific
operations are defined in the function call using the argument zonalNfun and each is added into the
result. Any zonal statistics compatible with the extract function in raster is accepted. Note that
gemerge does not accept raster stacks. If you have raster stacks they must be separated and the
layers integrated separately into the function.

If spat.lag = TRUE spatial lags of all numeric variables from a SpatialPolygonsDataFrame or
RasterLayer joined to target polygons are returned using first and also second order neighboring
weights matrices. The package assigns target polygons the mean value of units within each
neighborhood. When dynamic point aggregation is run and time.lag = TRUE, gemerge returns
the values of every target polygon, as well as its first and second order neighboring unit averages,
separately, at time t-1 and t-2 defined by interval in the argument time.

Value

Returns an object of class "geomerge".

The functions summary, print, plot overload the standard outputs for objects of type geomerge
providing summary information and and visualizations specific to the output object. An object of
class "geomerge" is a list containing the following three components:

data SpatialPolygonsDataFrame that contains all information merged with the target
layer. Column names are assigned the name of the input data object separated
by "." from a short description of the calculation, as well as modifiers such as
".1st" and ".2nd" for first- and second-order neighborhoods of target. In
the case of dynamic point data aggregation, ".t_1" and ".t_2" are used to label
first- and second-order temporal lags. For example, if geomerge is told to use
a SpatialPointsDataFrame called "vio" to count incidents of conflict con-
tained within units of target, the default output would include columns named
"vio.cnt", "vio.cnt.t_1", "vio.cnt.t_2", "vio.cnt.1st", "vio.cnt.1st.t_1",
"vio.cnt.1st.t_2", "vio.cnt.2nd", "vio.cnt.2nd.t_1", "vio.cnt.2nd.t_2".

inputData List containing the spatial objects used as input.

parameters List containing information on all input parameters used during integration.

Note

geomerge exclusively merges data using the global WGS84 coordinate reference system (CRS) to
ensure that areal statistics are accurate at all scales. If data are entered that are using a different
and/or projected CRS, the tool automatically first transforms the data. This on-the-fly transforma-
tion, however, may be very slow and it is advised to always enter inputs in WGS84.

Author(s)

Karsten Donnay and Andrew M. Linke.
References


See Also

geomerge-package, print.geomerge, plot.geomerge, summary.geomerge, generateGrid

Examples

data(geomerge)

# 1) Simple static integration of polygon data
output <- geomerge(geoEPR,target=states,silent=TRUE)
summary(output)

# 2) Static integration for point, polygon, raster data
output <- geomerge(ACLEDEVENT_TYPE,AidData$project_id,geoEPR,
gpw,na.rm=TRUE,target=states)
summary(output)
plot(output)

# 3) Dynamic point data integration for numeric variables
output <- geomerge(ACLED$FATALITIES,AidData$commitme_1,geoEPR,
target=states,time=c("2011-01-01", "2011-12-31","1"),
t_unit='months',point_agg='sum')
summary(output)
plot(output)

# 4) Population weighted assignment
output <- geomerge(geoEPR,target=states,assignment='max(pop)',
    population.data = gpw)
summary(output)
plot(output)

---

geomerge.assign Implement different assignment rules using SQL [Auxiliary Function]

Description

Implements assignment of polygon values to the target frame using different assignment rules. For efficient performance implemented using SQL.

Usage

geomerge.assign(polygon_input,target,assignment,population.data,optional.inputs,silent)
Arguments

- **polygon_input**: input SpatialPolygonsDataFrame parsed from `geomerge` main function.
- **target**: SpatialPolygonsDataFrame representing desired units of analysis. See Details of `geomerge`.
- **assignment**: identification of either population- or area-weighting assignment rules when handling SpatialPolygonsDataFrame joins to `target`. Default = "max(area)". See Details in `geomerge`.
- **population.data**: specifies data used for weighting if a population-based assignment rule is selected. See Details in `geomerge`.
- **optional.inputs**: Any optional inputs compatible with the `extract` function in `raster`.
- **silent**: Boolean switch to suppress any (non-critical) warnings and messages. Default = FALSE.

Details

For details on different input parameters, please refer to the detailed documentation in `geomerge`.

Value

Returns an object of class `data.frame` that contains the column from `input`, after proper assignment, that is to be added to `target@data`.

Author(s)

Karsten Donnay and Andrew M. Linke.

References


See Also

`geomerge-package`, `geomerge`, `generateGrid`
**geomerge.merge**  

**Performing dataset merger [Auxiliary Function]**

**Description**

Auxiliary function that performs the actual integration of the target frame with specified input data. The routine proceeds on dataset at a time.

**Usage**

```
geomerge.merge(data, data.name, target, standard.CRS, outdata, wghts, 
time, time.lag, spat.lag, zonal.fun, assignment, 
population.data, point agg, t_unit, silent, optional.inputs)
```

**Arguments**

- `data`: input dataset. See Details in `geomerge`.
- `data.name`: name of input dataset.
- `target`: `SpatialPolygonsDataFrame` representing desired units of analysis. See Details in `geomerge`.
- `standard.CRS`: Defines the CRS used. Default used in `geomerge` is WGS84.
- `outdata`: `data.frame` containing integrated data relative to the `SpatialPolygonsDataFrame` target.
- `wghts`: spatial weights calculated by link(`geomerge.neighbor`).
- `time`: specification of temporal window for temporal binning of point data by c(start_date, end_date, interval_length). Default = NA. See Details in `geomerge`.
- `time.lag`: Boolean indicating whether or not first and second order temporal lag values of all variables are returned. Default = `TRUE`.
- `spat.lag`: Boolean indicating whether or not first and second order spatial lag values of all variables are returned. Default = `TRUE`.
- `zonal.fun`: object of class `function` applied to values of `RasterLayer` when generating zonal statistics for each target polygon. Default = `sum`. See Details in `geomerge`.
- `assignment`: identification of either population- or area-weighting assignment rules when handling `SpatialPolygonsDataFrame` joins to `target`. Default = "area.assign". See Details in `geomerge`.
- `population.data`: specifies data used for weighting if a population-based assignment rule is selected. See Details in `geomerge`.
- `point.agg`: specification of aggregation format for data of type `SpatialPointsDataFrame`. Default = "cnt". See Details in `geomerge`.
- `t_unit`: temporal unit used for dynamic point aggregation. Default = "days".
- `silent`: Boolean switch to suppress any (non-critical) warnings and messages. Default = `FALSE`.
- `optional.inputs`: Any optional inputs compatible with the `extract` function in `raster`.
Details

For details on different input parameters, please refer to the detailed documentation in geomerge.

Value

Returns an object of class data.frame that contains all information from merger to target to be added to target$data in the main geomerge function. The documentation in geomerge provides a detailed overview over the columns returned and their naming conventions.

Author(s)

Karsten Donnay and Andrew M. Linke.

References


See Also

geomerge-package, geomerge, generateGrid

Description

Auxiliary function that uses functionality from spdep to retrieve first and second order neighbor weights.

Usage

geomerge.neighbor(polygon_input)

Arguments

polygon_input  a SpatialPolygonsDataFrame.

Details

The function serves as a wrapper for the poly2nb, nblag and nb2listw functions from spdep and returns first and second order neighbor weights using zero.policy = TRUE.

Value

Returns a list of lists of neighbor weights named "wts1" and "wts2".


**Author(s)**

Karsten Donnay and Andrew M. Linke.

**References**


**See Also**

`geomerge-package`, `geomerge`, `generateGrid`

---

**Description**


**Usage**

data(geomerge)

**Format**

A `RasterLayer` containing observations.

**Author(s)**

Karsten Donnay and Andrew M. Linke

**Source**


**References**

plot.geomerge

Description

Overloads the default `plot` for objects of class 'geomerge'.

Usage

```r
## S3 method for class 'geomerge'
plot(x, ...)  
```

Arguments

- `x`: object of class `geomerge`.
- `...`: further optional arguments.

Details

Returns a series of maps that visualizes numeric variables produced by `geomerge`. It returns a map for each unique numeric variable including first order spatially and temporally lagged values if `spat.lag=TRUE` and `time.lag=TRUE` when running `geomerge`. For spatial panels, the function by default returns values for the last period.

Five optional arguments that are specific to this plotting function can be provided. The first is `period`, a numeric input that allows to specify a specific period to be plotted. `inputs` must be a sequence of character strings specifying select variables to be plotted only. These have to have been merged (with the same name) in `geomerge`. `time.lag` and `spat.lag` override the boolean values parsed automatically from the result of `geomerge`. They are mainly meant to switch off plotting of spatial and temporal lags as they are ignored if these lags were not generated in the first place. The last argument is `ncol`, a numeric input, which allows to specify the width of the panel of plotted maps. By default, always 2 maps are shown side-by-side.

Note

`plot` for objects of class 'geomerge' relies in many core aspects of its functionality on `ggplot2`. If the target `SpatialPolygonsDataFrame` is very large it may reach or exceed the limits of what the plotting functionality from `ggplot2` can handle and `plot` may be very slow or even stall.

Author(s)

Karsten Donnay and Andrew M. Linke.

References

See Also

geomerge

print.geomerge

Print function for objects of class 'geomerge'.

Description

Overloads the default print for objects of class 'geomerge'.

Usage

## S3 method for class 'geomerge'
print(x, ...)

Arguments

x object of class geomerge.

... further arguments passed to or from other methods.

Author(s)

Karsten Donnay and Andrew M. Linke.

References


See Also

geomerge
**states**

*Polygon dataset to illustrate the functionality of geomerge*

---

**Description**

Nigeria administrative units (ADM1) dataset used as example for the target `SpatialPolygonsDataFrame` data are merged. The dataset is available at [http://www.arcgis.com/home/item.html?id=0e58995046b74254911c1dc0eb756fa4](http://www.arcgis.com/home/item.html?id=0e58995046b74254911c1dc0eb756fa4).

**Usage**

data(geomerge)

**Format**

A `SpatialPolygonsDataFrame` containing observation and that data is merged to using `geomerge`.

**Details**

Note that the polygons in `states` have been simplified to reduce the size of the `SpatialPolygonsDataFrame` used as integration target for easier illustration. This applies, in particular, to the Niger Delta region of Nigeria.

**Author(s)**

Karsten Donnay and Andrew M. Linke

**Source**

http://www.arcgis.com/home/item.html?id=0e58995046b74254911c1dc0eb756fa4

**References**

Summary function for objects of class 'geomerge'.

Description

Overloads the default summary for objects of class 'geomerge'.

Usage

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

Arguments

- **object**: object of class `geomerge`.  
- **...**: further arguments passed to or from other methods.

Value

Returns a number of summary statistics describing the results of the `geomerge` integration, including how many variables were integrated, which of those are numerical vs. non numerical and whether spatially and/or temporally lagged values are available.

Author(s)

Karsten Donnay and Andrew M. Linke.

References


See Also

`geomerge`
Index

ACLED, 3
AidData, 4

generateGrid, 2, 5, 10, 11, 13, 14
geoEPR, 6
g geomerge, 2, 5, 7, 11–16, 18
g geomerge-package, 2
g geomerge.assign, 2, 10
g geomerge.merge, 2, 12
g geomerge.neighbor, 2, 13
gpw, 14

plot.geomerge, 10, 15
print.geomerge, 10, 16

states, 17
summary.geomerge, 10, 18