

Package: cleangeo (via r-universe)

July 27, 2024

Version 0.3-1

Date 2023-09-01

Title Cleaning Geometries from Spatial Objects

Maintainer Emmanuel Blondel <emmanuel.blondel1@gmail.com>

Depends R (>= 2.15)

Imports methods, sp, sf

Suggests testthat, knitr, markdown, pbapply

Description Provides a set of utility tools to inspect spatial objects, facilitate handling and reporting of topology errors and geometry validity issue with sp objects. Finally, it provides a geometry cleaner that will fix all geometry problems, and eliminate (at least reduce) the likelihood of having issues when doing spatial data processing.

License GPL (>= 2)

URL <https://github.com/eblondel/cleangeo>

VignetteBuilder knitr

BugReports <https://github.com/eblondel/cleangeo/issues>

RoxygenNote 7.2.3

Repository <https://eblondel.r-universe.dev>

RemoteUrl <https://github.com/eblondel/cleangeo>

RemoteRef HEAD

RemoteSha 7008ad5e3aaf55615a80aef96a75d00545785a35

Contents

cleangeo	2
clgeo_Clean	2
clgeo_CleanByPolygonation.Polygon	3
clgeo_CleanByPolygonation.Polygons	4
clgeo_CleanByPolygonation.SpatialPolygons	5

clgeo_CollectionReport	5
clgeo_GeometryReport	6
clgeo_IsValid	7
clgeo_SummaryReport	8
clgeo_SuspiciousFeatures	9

Index 10

cleangeo	<i>Clean Geometries from Spatial Objects</i>
----------	--

Description

cleangeo provides a set of utility tools to inspect spatial objects, facilitate handling and reporting of topology errors and geometry validity issues. Finally, it provides a geometry cleaner that will fix all geometry problems, and eliminate (at least reduce) the likelihood of having issues when doing spatial data processing.

Author(s)

Emmanuel Blondel <emmanuel.blondel1@gmail.com>

clgeo_Clean	<i>clgeo_Clean</i>
-------------	--------------------

Description

Function to clean a spatial data collection

Usage

```
clgeo_Clean(sp, errors.only = NULL, strategy = "SF", verbose = FALSE)
```

Arguments

sp	object extending the Spatial-class as defined in sp
errors.only	an object of class vector giving the types of errors for which the output should be bounded. Default value is NULL (<i>i.e.</i> the output will include features for which both errors and warnings were raised.).
strategy	advanced strategy to clean geometries. Default is "SF", alternate values are "POLYGONATION", "BUFFER" (old methods).
verbose	Indicates whether the clean logs have to be printed. Default value is FALSE.

Value

an object extending the [Spatial-class](#) as defined in **sp**, with cleaned geometries.

Note

About cleaning strategy: The polygonation method is a tentative alternate method to triangulation to clean geometries and to the classical often used 'buffer' approach. In the polygonation method, triangulation is skipped and a re-polygonation intuitive algorithm is applied to rebuild the source invalid geometry into one or more valid polygonal geometries. With the progress done on validating geometries, especially with **sf**, the default method in cleangeo has now been switched to the use of `sf::st_make_valid`

Author(s)

Emmanuel Blondel <emmanuel.blondel1@gmail.com>

Examples

```
require(sf)
file <- system.file("extdata", "example.shp", package = "cleangeo")
sf <- sf::st_read(file)
sp <- as(sf, "Spatial")

sp.clean <- clgeo_Clean(sp)
report.clean <- clgeo_CollectionReport(sp.clean)
clgeo_SummaryReport(report.clean)
```

clgeo_CleanByPolygonation.Polygon
clgeo_CleanByPolygonation.Polygon

Description

Function to clean a [Polygon-class](#) object by polygonation.

Usage

```
clgeo_CleanByPolygonation.Polygon(p, verbose = FALSE)
```

Arguments

p object of class [Polygon-class](#) as defined in **sp**
verbose Indicates whether the clean logs have to be printed. Default value is FALSE.

Value

a list of objects of class [Polygon-class](#) as defined in **sp**, with cleaned geometries.

Note

The polygonation method is a tentative alternate method to triangulation to clean geometries. In this method, triangulation is skipped and a re-polygonation algorithm is applied.

Author(s)

Emmanuel Blondel <emmanuel.blondel1@gmail.com>

`clgeo_CleanByPolygonation.Polygons`
clgeo_CleanByPolygonation.Polygons

Description

Function to clean a [Polygons](#) object by polygonation

Usage

```
clgeo_CleanByPolygonation.Polygons(p, verbose = FALSE)
```

Arguments

<code>p</code>	object of class Polygons-class as defined in sp
<code>verbose</code>	Indicates whether the clean logs have to be printed. Default value is FALSE.

Value

an object of class [Polygons-class](#) as defined in **sp**, with cleaned geometries.

Note

The polygonation method is a tentative alternate method to triangulation to clean geometries. In this method, triangulation is skipped and a re-polygonation algorithm is applied.

Author(s)

Emmanuel Blondel <emmanuel.blondel1@gmail.com>

clgeo_CleanByPolygonation.SpatialPolygons
clgeo_CleanByPolygonation.SpatialPolygons

Description

Function to clean a [SpatialPolygons](#) object by polygonation

Usage

```
clgeo_CleanByPolygonation.SpatialPolygons(sp, verbose = FALSE)
```

Arguments

sp object extending the [Spatial-class](#) as defined in **sp**
verbose Indicates whether the clean logs have to be printed. Default value is FALSE.

Value

an object extending the [Spatial-class](#) as defined in **sp**, with cleaned geometries.

Note

The polygonation method is a tentative alternate method to triangulation to clean geometries. In this method, triangulation is skipped and a re-polygonation algorithm is applied.

Author(s)

Emmanuel Blondel <emmanuel.blondel1@gmail.com>

clgeo_CollectionReport
clgeo_CollectionReport

Description

Function to get a spatial data collection validation report. The function outputs a `data.frame` binding all geometry validity reports, each one produced by [clgeo_GeometryReport](#)

Usage

```
clgeo_CollectionReport(sp)
```

Arguments

sp object extending the [Spatial-class](#) as defined in **sp**

Value

an object of class `data.frame` with the following columns:

- *type* eventual **rgeos** issue
- *valid* geometry validity status (according to OGC specifications)
- *issue_type* type of geometry issue
- *error_msg* caught message when error raised about geometry
- *warning_msg* caught message when warning raised about geometry

Author(s)

Emmanuel Blondel <emmanuel.blondel1@gmail.com>

See Also

[clgeo_GeometryReport](#)

Examples

```
require(sf)
file <- system.file("extdata", "example.shp", package = "cleangeo")
sf <- sf::st_read(file)
sp <- as(sf, "Spatial")

report <- clgeo_CollectionReport(sp)
```

`clgeo_GeometryReport` *clgeo_GeometryReport*

Description

Function to get a geometry validation report: The report informs on the following:

- *type* eventual **rgeos** issue
- *valid* geometry validity status (according to OGC specifications)
- *issue_type* type of geometry issue
- *msg* caught message when error raised about geometry

Usage

```
clgeo_GeometryReport(spgeom)
```

Arguments

`spgeom` object extending the [Spatial-class](#) as defined in **sp**

Value

an object of class `list` giving the following:

- *type* eventual **rgeos** issue
- *valid* geometry validity status (according to OGC specifications)
- *issue_type* type of geometry issue
- *msg* caught message when warning raised about geometry

Author(s)

Emmanuel Blondel <emmanuel.blondel1@gmail.com>

clgeo_IsValid	<i>clgeo_IsValid</i>
---------------	----------------------

Description

Wrapper method to try performing `rgeos::gIsValid` call and catch eventual warnings or errors (in particular GEOS exceptions).

Usage

```
clgeo_IsValid(sp, verbose = FALSE)
```

Arguments

`sp` object extending the [Spatial-class](#) as defined in `sp`
`verbose` object of class "logical". Default value is FALSE.

Value

an object of class "logical". TRUE if valid, FALSE otherwise

Author(s)

Emmanuel Blondel <emmanuel.blondel1@gmail.com>

Examples

```
require(sf)
file <- system.file("extdata", "example.shp", package = "cleangeo")
sf <- sf::st_read(file)
sp <- as(sf, "Spatial")
clgeo_IsValid(sp)
```

clgeo_SummaryReport *clgeo_SummaryReport*

Description

Function to get summary of a spatial data collection report returned by [clgeo_CollectionReport](#)

Usage

```
clgeo_SummaryReport(report)
```

Arguments

report a report object as returned by [clgeo_CollectionReport](#)

Value

an object of class `table` giving the report summary. The summary gives the counting by value for each of the report columns:

- *type* eventual geometry issue
- *valid* geometry validity status (according to OGC specifications)
- *issue_type* type of geometry issue
- *msg* caught message when error raised about geometry

Author(s)

Emmanuel Blondel <emmanuel.blondel1@gmail.com>

See Also

[clgeo_CollectionReport](#)

Examples

```
require(sf)
file <- system.file("extdata", "example.shp", package = "cleangeo")
sf <- sf::st_read(file)
sp <- as(sf, "Spatial")

report <- clgeo_CollectionReport(sp)
clgeo_SummaryReport(report)
```

clgeo_SuspiciousFeatures
clgeo_SuspiciousFeatures

Description

Function to get the list of index of suspicious geometries within a spatial data collection, given a spatial data collection report returned by the function [clgeo_CollectionReport](#)

Usage

```
clgeo_SuspiciousFeatures(report, errors.only = NULL)
```

Arguments

report	a report object as returned by clgeo_CollectionReport for which the output should be bounded. Default value is NULL (<i>i.e.</i> the output will include features for which both errors and errors were raised.).
errors.only	an object of class vector giving the types of errors for which the output should be bounded. Default value is NULL (<i>i.e.</i> the output will include features for which both errors and errors were raised.).

Value

an object of class vector giving the numeric indexes of spatial objects tagged as suspicious (*i.e.* that are not valid according to OGC specifications)

Author(s)

Emmanuel Blondel <emmanuel.blondel1@gmail.com>

See Also

[clgeo_CollectionReport](#)

Examples

```
require(sf)
file <- system.file("extdata", "example.shp", package = "cleangeo")
sf <- sf::st_read(file)
sp <- as(sf, "Spatial")

report <- clgeo_CollectionReport(sp)
nv <- clgeo_SuspiciousFeatures(report)
```

Index

- * **IsValid**
 - clgeo_IsValid, 7
- * **clean**
 - clgeo_Clean, 2
 - clgeo_CleanByPolygonation.Polygon, 3
 - clgeo_CleanByPolygonation.Polygons, 4
 - clgeo_CleanByPolygonation.SpatialPolygons, 5
 - clgeo_IsValid, 7
- * **gIsValid**
 - clgeo_IsValid, 7
- * **geometry**
 - clgeo_Clean, 2
 - clgeo_CleanByPolygonation.Polygon, 3
 - clgeo_CleanByPolygonation.Polygons, 4
 - clgeo_CleanByPolygonation.SpatialPolygons, 5
 - clgeo_CollectionReport, 5
 - clgeo_GeometryReport, 6
 - clgeo_IsValid, 7
 - clgeo_SummaryReport, 8
 - clgeo_SuspiciousFeatures, 9
- * **summary**
 - clgeo_Clean, 2
 - clgeo_CleanByPolygonation.Polygon, 3
 - clgeo_CleanByPolygonation.Polygons, 4
 - clgeo_CleanByPolygonation.SpatialPolygons, 5
 - clgeo_IsValid, 7
 - clgeo_SummaryReport, 8
- * **validity**
 - clgeo_Clean, 2
 - clgeo_CleanByPolygonation.Polygon, 3
 - clgeo_CleanByPolygonation.Polygons, 4
 - clgeo_CleanByPolygonation.SpatialPolygons, 5
 - clgeo_CollectionReport, 5
 - clgeo_GeometryReport, 6
 - clgeo_IsValid, 7
 - clgeo_SummaryReport, 8
 - clgeo_SuspiciousFeatures, 9
- cleangeo, 2
- cleangeo-package (cleangeo), 2
- clgeo_Clean, 2
- clgeo_CleanByPolygonation.Polygon, 3
- clgeo_CleanByPolygonation.Polygons, 4
- clgeo_CleanByPolygonation.SpatialPolygons, 5
- clgeo_CleanByTriangulation (clgeo_CleanByPolygonation.SpatialPolygons), 5
- clgeo_CollectionReport, 5, 8, 9
- clgeo_Geometry (clgeo_GeometryReport), 6
- clgeo_GeometryReport, 5, 6, 6
- clgeo_IsValid, 7
- clgeo_SummaryReport, 8
- clgeo_SuspiciousFeatures, 9
- Polygons, 4
- SpatialPolygons, 5