

Package ‘cleangeo’

April 17, 2021

Version 0.2-4

Date 2021-04-17

Title Cleaning Geometries from Spatial Objects

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

Depends R (>= 2.15), rgeos, maptools

Imports methods, sp

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 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.

License GPL (>= 2)

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

VignetteBuilder knitr

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

RoxygenNote 7.1.0

NeedsCompilation no

Author Emmanuel Blondel [aut, cre] (<<https://orcid.org/0000-0002-5870-5762>>)

Repository CRAN

Date/Publication 2021-04-17 13:30:02 UTC

R topics documented:

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 11

cleangeo *Clean Geometries from Spatial Objects*

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.

Details

Package: cleangeo
 Type: Package
 Version: 0.2-4
 Date: 2021-04-17
 License: GPL(>=2.0)
 LazyLoad: yes

Author(s)

Emmanuel Blondel <emmanuel.blondel1@gmail.com>

clgeo_Clean *clgeo_Clean*

Description

Function to clean a spatial data collection

Usage

```
clgeo_Clean(sp, errors.only = NULL, strategy = "POLYGONATION", 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 errors were raised.). At now, this argument accepts the error type "ORPHANED_HOLE".
strategy	advanced strategy to clean geometries. Default is "POLYGONATION", alternate value is "BUFFER" (old method).
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.

Author(s)

Emmanuel Blondel <emmanuel.blondel1@gmail.com>

Examples

```
require(maptools)
file <- system.file("extdata", "example.shp", package = "cleangeo")
sp <- readShapePoly(file)

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

p object of class **Polygons-class** as defined in **sp**
verbose 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(maptools)
file <- system.file("extdata", "example.shp", package = "cleangeo")
sp <- readShapePoly(file)

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
- *error_msg* caught message when error raised about geometry
- *warning_msg* caught message when warning 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
- *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

[gIsValid](#)

clgeo_IsValid *clgeo_IsValid*

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(maptools)
file <- system.file("extdata", "example.shp", package = "cleangeo")
sp <- readShapePoly(file)
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 **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_CollectionReport](#)

Examples

```
require(maptools)
file <- system.file("extdata", "example.shp", package = "cleangeo")
sp <- readShapePoly(file)

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
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.). At now, this argument only accepts the error type "ORPHANED_HOLE".

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(maptools)
file <- system.file("extdata", "example.shp", package = "cleangeo")
sp <- readShapePoly(file)

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
- 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
- gIsValid, 7
- Polygons, 4
- SpatialPolygons, 5