

Package ‘wicket’

November 19, 2017

Type Package

Title Utilities to Handle WKT Spatial Data

Version 0.4.0

Date 2017-11-18

Author Oliver Keyes [aut, cre]

Maintainer Oliver Keyes <ironholds@gmail.com>

Description Utilities to generate bounding boxes from 'WKT' (Well-Known Text) objects and R data types, validate 'WKT' objects and convert object types from the 'sp' package into 'WKT' representations.

License MIT + file LICENSE

LazyData TRUE

LinkingTo Rcpp, BH

Imports Rcpp

RoxygenNote 6.0.1

Suggests testthat, knitr, rmarkdown

VignetteBuilder knitr

URL <https://github.com/ropensci/wicket>

BugReports <https://github.com/ropensci/wicket/issues>

NeedsCompilation yes

Repository CRAN

Date/Publication 2017-11-19 17:20:10 UTC

R topics documented:

bounding_wkt	2
sp_convert	3
validate_wkt	4
wicket	5
wkt_bounding	5

wkt_centroid	6
wkt_coords	6
wkt_correct	7

Index	9
--------------	----------

bounding_wkt	<i>Generate Bounding Boxes</i>
--------------	--------------------------------

Description

bounding_wkt takes bounding boxes, in various formats, and turns them into WKT POLYGONS.

Usage

```
bounding_wkt(min_x, min_y, max_x, max_y, values = NULL)
```

Arguments

min_x	a numeric vector of the minimum value for x coordinates.
min_y	a numeric vector of the minimum value for y coordinates.
max_x	a numeric vector of the maximum value for x coordinates.
max_y	a numeric vector of the maximum value for y coordinates.
values	as an alternative to specifying the various values as vectors, a list of length-4 numeric vectors containing min and max x and y values, or just a single vector fitting that spec. NULL (meaning that the other parameters will be expected) by default.

Value

a character vector of WKT POLYGON objects

See Also

[wkt_bounding](#), to turn WKT objects of various types into a matrix or data.frame of bounding boxes.

Examples

```
# With individual columns
bounding_wkt(10, 12, 14, 16)

# With a list
bounding_wkt(values = list(c(10, 12, 14, 16)))
```

sp_convert	<i>Convert SpatialPolygons or SpatialPolygonDataFrames into WKT objects</i>
------------	---

Description

sp_convert turns objects from the sp package - specifically, SpatialPolygons and SpatialPolygonDataFrames - into WKT POLYGONs or MULTIPOLYGONs

Usage

```
sp_convert(x, group = TRUE)
```

Arguments

x	a list of SP/SPDF objects (or a single object)
group	whether or not to group coordinates together in the case that an object in x has multiple sets of coordinates. If TRUE (the default), such objects will be returned as MULTIPOLYGONs - if FALSE, as a vector of POLYGONs.

Value

either a character vector of WKT objects - one per sp object - if group is TRUE, or a list of vectors if group is FALSE.

See Also

[bounding_wkt](#), for turning bounding boxes within sp objects into WKT objects.

Examples

```
## Not run:
library(sp)
s1 <- SpatialPolygons(list(Polygons(list(Polygon(cbind(c(2,4,4,1,2),c(2,3,5,4,2))))), "s1"))
sp_convert(s1)

## End(Not run)
```

validate_wkt	<i>Validate WKT objects</i>
--------------	-----------------------------

Description

`validate_wkt` takes a vector of WKT objects and validates them, returning a data.frame containing the status of each entry and (in the case it cannot be parsed) any comments as to what, in particular, may be wrong with it. It does not, unfortunately, check whether the object meets the WKT spec - merely that it is formatted correctly.

Usage

```
validate_wkt(x)
```

Arguments

`x` a character vector of WKT objects.

Value

a data.frame of two columns, `is_valid` (containing TRUE or FALSE values for whether the WKT object is parseable and valid) and `comments` (containing any error messages in the case that the WKT object is not). If the objects are simply NA, both fields will contain NA.

See Also

[sp_convert](#) for generating valid WKT objects from SpatialPolygons and SpatialPolygonDataFrames, or [wkt_correct](#) for correcting WKT objects that fail validity checks due to having a non-default orientation.

Examples

```
wkt <- c("POLYGON ((30 10, 40 40, 20 40, 10 20, 30 10))",  
        "ARGHLEFLARFDGF",  
        "LINESTRING (30 10, 10 90, 40 out of cheese error redo universe from start)")  
validate_wkt(wkt)
```

wicket	<i>Well-Known Text utilities</i>
--------	----------------------------------

Description

This package provides basic, standalone functions for WKT manipulation, generation and validation, backed by C++ for improved speed.

See Also

[validate_wkt](#) for checking the validity of WKT objects, [sp_convert](#) for turning SP objects *into* WKT, and [wkt_bounding](#) and [bounding_wkt](#) for bounding box conversion in both directions.

wkt_bounding	<i>Convert WKT Objects into Bounding Boxes</i>
--------------	--

Description

[wkt_bounding](#) turns WKT objects (specifically points, linestrings, polygons, and multi-points/linestrings/polygons) into bounding boxes.

Usage

```
wkt_bounding(wkt, as_matrix = FALSE)
```

Arguments

wkt	a character vector of WKT objects.
as_matrix	whether to return the results as a matrix (TRUE) or data.frame (FALSE). Set to FALSE by default.

Value

either a data.frame or matrix, depending on the value of `as_matrix`, containing four columns - `min_x`, `min_y`, `max_x` and `max_y` - representing the various points of the bounding box. In the event that a valid bounding box cannot be generated (due to the invalidity or incompatibility of the WKT object), NAs will be returned.

See Also

[bounding_wkt](#), to turn R-size bounding boxes into WKT objects.

Examples

```
wkt_bounding("POLYGON ((30 10, 40 40, 20 40, 10 20, 30 10))")
```

wkt_centroid	<i>Extract Centroid</i>
--------------	-------------------------

Description

`get_centroid` identifies the 2D centroid in a WKT object (or vector of WKT objects). Note that it assumes cartesian values.

Usage

```
wkt_centroid(wkt)
```

Arguments

`wkt` a character vector of WKT objects, represented as strings

Value

a data.frame of two columns, `lng` and `lat`, with each row containing the centroid from the corresponding `wkt` object. In the case that the object is NA (or cannot be decoded) the resulting values will also be NA

See Also

[wkt_coords](#) to extract all coordinates, and [wkt_bounding](#) to extract a bounding box.

Examples

```
wkt_centroid("POLYGON((2 1.3,2.4 1.7))")
# lng lat
#1 2 1.3
```

wkt_coords	<i>Extract Latitude and Longitude from WKT polygons</i>
------------	---

Description

`wkt_coords` extracts lat/long values from WKT polygons, specifically the outer shell of those polygons (working on the assumption that said outer edge is what you want).

Because it assumes *coordinates*, it also assumes a sphere - say, the earth - and uses spherical coordinate values.

Usage

```
wkt_coords(wkt)
```

Arguments

wkt a character vector of WKT objects

Value

a data.frame of four columns; object (containing which object the row refers to), ring containing which layer of the object the row refers to, lng and lat.

See Also

[wkt_bounding](#) to extract a bounding box, and [wkt_centroid](#) to extract the centroid.

Examples

```
wkt_coords("POLYGON ((30 10, 40 40, 20 40, 10 20, 30 10))")
# object ring lng lat
# 1     1 outer 30 10
# 2     1 outer 40 40
# 3     1 outer 20 40
# 4     1 outer 10 20
# 5     1 outer 30 10
```

wkt_correct

Correct Incorrectly Oriented WKT Objects

Description

wkt_correct does precisely what it says on the tin, correcting the orientation of WKT objects that are improperly oriented (say, back to front). It can be applied to WKT objects that, when validated with [validate_wkt](#), fail for that reason.

Usage

```
wkt_correct(x)
```

Arguments

x a character vector of WKT objects to correct

Value

a character vector, the same length as x, containing either the original value (if there was no correction to make, or if the object was invalid for other reasons) or the corrected WKT value.

Examples

```
# A WKT object
wkt <- "POLYGON((30 20, 10 40, 45 40, 30 20), (15 5, 5 10, 10 20, 40 10, 15 5))"

# That's invalid due to a non-default orientation
validate_wkt(wkt)

# And suddenly isn't!
wkt_correct(wkt)
```


Index

`bounding_wkt`, [2](#), [3](#), [5](#)

`sp_convert`, [3](#), [4](#), [5](#)

`validate_wkt`, [4](#), [5](#), [7](#)

`wicket`, [5](#)

`wicket-package (wicket)`, [5](#)

`wkt_bounding`, [2](#), [5](#), [5](#), [6](#), [7](#)

`wkt_centroid`, [6](#), [7](#)

`wkt_coords`, [6](#), [6](#)

`wkt_correct`, [4](#), [7](#)