

# Package ‘cshapes’

October 31, 2015

**Version** 0.5-1

**Date** 2015-10-30

**Title** CShapes Dataset and Utilities

**Author** Nils B. Weidmann <nils.weidmann@uni-konstanz.de>, Kristian Skrede Gleditsch <ksg@essex.ac.uk>

**Maintainer** Nils B. Weidmann <nils.weidmann@uni-konstanz.de>

**URL** <http://nils.weidmann.ws/projects/cshapes>,  
[https://r-forge.r-project.org/scm/viewvc.php/\\*checkout\\*/pkg/inst/shp/cshapes\\_shapefile\\_documentation.txt?root=cshapes](https://r-forge.r-project.org/scm/viewvc.php/*checkout*/pkg/inst/shp/cshapes_shapefile_documentation.txt?root=cshapes)

**Depends** R (>= 2.8.0), sp (>= 1.0-5), maptools (>= 0.8-21), plyr (>= 1.8)

**LazyLoad** yes

**Description** Package for CShapes, a GIS dataset of country boundaries (1946-2015). Includes functions for data extraction and the computation of distance matrices and -lists.

**License** GPL (>= 2)

**NeedsCompilation** no

**Repository** CRAN

**Date/Publication** 2015-10-31 00:17:21

## R topics documented:

cshapes-package	2
cshp	3
distlist	4
distmatrix	5

<b>Index</b>	<b>7</b>
--------------	----------

## Description

R Package for CShapes, a GIS dataset of country boundaries (1946-2015). Includes functions for data extraction and the computation of weights matrices.

## Details

The cshapes package facilitates the use of CShapes from R. CShapes is a GIS dataset of historical country boundaries (1946-2015) and compatible with two country lists (Gleditsch and Ward 1999 and Correlates of War, see references below). In particular, the package enables access to the dataset directly, as well as distance computations on country polygons for specific points in time. Access to the dataset from within R is done using the `cshp` function. Two functions exist to compute minimum-, capital- and centroid distances between countries: the `distmatrix` function returns these as a matrix (convenient for many spatial statistical applications), and the `distlist` function returns a list of dyadic distances (distances between country pairs). See the examples given in the documentation of the functions in this package. The main cshapes website is located at <http://nils.weidmann.ws/projects/cshapes> and contains additional documentation and examples for the dataset and the R package.

For more information on the CShapes dataset, see Weidmann, Nils B., Doreen Kuse and Kristian Skrede Gleditsch. 2010. "The Geography of the International System: The CShapes Dataset." *International Interactions* 36(1).

The associated R package is introduced in Weidmann, Nils B. and Kristian Skrede Gleditsch. 2010. "Mapping and Measuring Country Shapes: The cshapes Package." *R Journal* 2(1). Available online at [http://journal.r-project.org/archive/2010-1/RJournal\\_2010-1\\_Weidmann+Skrede~Gleditsch.pdf](http://journal.r-project.org/archive/2010-1/RJournal_2010-1_Weidmann+Skrede~Gleditsch.pdf).

## Author(s)

Nils B. Weidmann <nils.weidmann@uni-konstanz.de>, Doreen Kuse <doreen.kuse@gmail.com>, Kristian S. Gleditsch <ksg@essex.ac.uk>

Maintainer: Nils B. Weidmann <nils.weidmann@uni-konstanz.de>, Kristian S. Gleditsch <ksg@essex.ac.uk>

## References

Correlates of War Project. 2011. "State System Membership List, v2011.1." Online, <http://correlatesofwar.org>.

Gleditsch, Kristian S. & Michael D. Ward. 1999. "Interstate System Membership: A Revised List of the Independent States since 1816." *International Interactions* 25: 393-413. Online, <http://privatewww.essex.ac.uk/~ksg/statelist.html>.

---

cshp	<i>Access the CShapes dataset in R</i>
------	--

---

### Description

The `cshp` function makes the `cshapes` dataset available for usage in R. If no date is given, it returns a `SpatialPolygonsDataFrame` with the complete CShapes dataset. If specified, the date is used to create a snapshot of the dataset, containing all `cshapes` polygons active at the given date.

### Usage

```
cshp(date=NA, useGW=TRUE)
```

### Arguments

<code>date</code>	The date for which the <code>cshapes</code> polygons should be extracted. This argument must be of type <code>Date</code> and must be in the range 1/1/1946 - 30/6/2012. If omitted, the complete dataset is returned.
<code>useGW</code>	Boolean argument specifying the system membership coding. <code>TRUE</code> (Default): Gleditsch and Ward (1999). <code>FALSE</code> : Correlates of War.

### Value

A `SpatialPolygonsDataFrame`, containing the complete CShapes dataset, or the CShapes snapshot for the specified date.

### Author(s)

Nils B. Weidmann

### See Also

[SpatialPolygonsDataFrame](#)

### Examples

```
# Retrieve the dataset
cshp.data <- cshp()

# Get summary statistics
summary(cshp.data)

# Extract Switzerland
switzerland <- cshp.data[cshp.data$COWCODE==225,]

# Plot Switzerland
plot(switzerland)
```

```
# Extract a snapshot of cshapes as of June 30, 2002
# using the Gleditsch&Ward coding system
cshp.2002 <- cshp(date=as.Date("2002-6-30"), useGW=TRUE)
```

---

distlist

---

*Compute distance lists on the CShapes dataset*


---

### Description

This function computes a distance list for the given date. It selects all the active CShapes polygons, determines their distances and outputs a distance list. A distance list is a list of dyads of countries and the distances between them. This list is returned as a data frame with three columns:

1. `ccode1` – country 1's code in the coding system specified by the `useGW` parameter
2. `ccode2` – country 2's code in the coding system specified by the `useGW` parameter
3. `capdist`, `centdist`, `mindist` – distance between country 1 and country 2 in km, where distance can be either capital distance, centroid distance or minimum distance, as specified by the `type` parameter. The latter computation is very expensive if polygons have many nodes. For that reason, the function simplifies the country polygons according to the Douglas-Peucker algorithm ([http://en.wikipedia.org/wiki/Ramer-Douglas-Peucker\\_algorithm](http://en.wikipedia.org/wiki/Ramer-Douglas-Peucker_algorithm)), which eliminates points from the polygons and speeds up computation. The `tolerance` parameter specifies the tolerance for the simplification; a value of 0 disables it.

Note that the function returns directed dyads. For example, if there is a country with code 1 and a country with code 2, the resulting data frame contains the dyads (1,2), (2,1), (1,1) and (2,2). It is easy to extract the directed dyads from this data frame (see example below).

### Usage

```
distlist(date, type="mindist", tolerance=0.1, useGW=T)
```

### Arguments

<code>date</code>	The date for which the distance list should be computed. This argument must be of type <code>Date</code> and must be in the range 1/1/1946 - 30/6/2012.
<code>type</code>	Specifies the type of distance list: <code>capdist</code> for capital distances, <code>centdist</code> for centroid distances, and <code>mindist</code> for minimum distances.
<code>useGW</code>	Boolean argument specifying the system membership coding. <code>TRUE</code> (Default): Gleditsch and Ward (1999). <code>FALSE</code> : Correlates of War.
<code>tolerance</code>	Tolerance for polygon simplification according the the Douglas-Peucker algorithm. Only used for <code>mindist</code> computation ( <code>type="mindist"</code> ).

### Value

A distance list with three columns, the first two of which contain the identifiers for the two countries in the dyad, and the third one containing the distance between the two countries.

**Author(s)**

Nils B. Weidmann

**Examples**

```
# Compute a list of minimum distances
# for the international system on 1/1/1946
# using the Correlates of War list and the default accuracy
## Not run: dl <- distlist(as.Date("1946-1-1"), type="capdist", tolerance=0.5, useGW=FALSE)

# we eliminate duplicate dyads
## Not run: dl <- subset(dl, ccode1 < ccode2)
```

---

distmatrix*Compute distance matrices on the CShapes dataset*

---

**Description**

This function computes a distance matrix for the given date. It selects all the active CShapes polygons, determines their distances and outputs a distance matrix in kilometers. The function can compute different types of distance matrices, specified by the "type" parameter: (i) capital distances, and (ii) centroid distances, and (iii) minimum distances between polygons. The latter computation is very expensive if polygons have many nodes. For that reason, the function simplifies the country polygons according to the Douglas-Peucker algorithm ([http://en.wikipedia.org/wiki/Ramer-Douglas-Peucker\\_algorithm](http://en.wikipedia.org/wiki/Ramer-Douglas-Peucker_algorithm)), which eliminates points from the polygons and speeds up computation. The tolerance parameter specifies the tolerance for the simplification; a value of 0 disables it.

**Usage**

```
distmatrix(date, type="mindist", tolerance=0.1, useGW=T)
```

**Arguments**

date	The date for which the distance matrix should be computed. This argument must be of type Date and must be in the range 1/1/1946 - 30/6/2012.
type	Specifies the type of distance matrix: capdist for capital distances, centdist for centroid distances, and mindist for minimum distances.
useGW	Boolean argument specifying the system membership coding. TRUE (Default): Gleditsch and Ward (1999). FALSE: Correlates of War.
tolerance	Tolerance for polygon simplification according the the Douglas-Peucker algorithm. Only used for mindist computation (type="mindist").

**Value**

A quadratic weights matrix, with the row and column labels containing the country identifiers in the specified coding system (COW or G&W).

**Author(s)**

Nils B. Weidmann

**Examples**

```
# Compute a matrix of minimum distances
# for the international system on 1/1/1946
# using the Correlates of War list and the default accuracy
## Not run: wmat <- distmatrix(as.Date("1946-1-1"), type="capdist", tolerance=0.5, useGW=FALSE)

# For model estimation, our matrix should contain
# the inverted distances
## Not run: wmat <- 1/wmat

# Fix the values along the diagonale
## Not run: diag(wmat) <- 0
```

# Index

## \*Topic **package**

[cshapes-package](#), [2](#)

[cshapes \(cshapes-package\)](#), [2](#)

[cshapes-package](#), [2](#)

[cshp](#), [2](#), [3](#)

[distlist](#), [2](#), [4](#)

[distmatrix](#), [2](#), [5](#)

[SpatialPolygonsDataFrame](#), [3](#)