

# polyArea

November 7, 2019

---

readWKTSplit	<i>Get polygon area and convert to or from geographic and Cartesian coordinates.</i>
--------------	--------------------------------------------------------------------------------------

---

## Description

polyArea calculates the area in nautical mile squared of a multipolygon (for one stratum).

geo2xy converts from geographic to or from Cartesian coordinates or the inverse.

getMatrixList converts the input to a list of matrices.

getMultipolygon converts the input to a multipolygon wkt string.

getSpatial converts the input to a Spatial object.

matrix2multipolygon identical to getMultipolygon.

multipolygon2matrix identical to getMatrixList.

## Usage

```
readWKTSplit(x, ...)
```

```
polyArea(x, requireClosed = TRUE)
```

```
polyArea_test(x, par = list(proj = "laea", units = "kmi", lon_0 = NA,  
  lat_0 = NA, x_0 = 0, y_0 = 0, ellps = "WGS84", datum = "WGS84"),  
  longlat = TRUE, ...)
```

```
getMatrixList(x, drop = TRUE, data.frame.out = FALSE)
```

```
getMultipolygon(x, drop = TRUE, data.frame.out = FALSE,  
  requireClosed = TRUE)
```

```
getSpatial(x)
```

```
matrix2multipolygon(x, drop = TRUE, data.frame.out = FALSE,
  requireClosed = TRUE)
```

```
multipolygon2matrix(x, drop = TRUE, data.frame.out = FALSE)
```

### Arguments

<code>x</code>	Either a two column matrix of x and y coordinates, indicating only one polygon, or a list of such matrices, indicating several polygons. If a list of lists of two column matrices are given, the first matrix of each list is the polygon, and the following are subtractions. wkt can also be a wkt string such as "MULTIPOLYGON(((4 55, 15 56, 15 59, 4 59, 4 55)))".
<code>...</code>	Further proj4 parameters overriding those in <code>par</code> .
<code>requireClosed</code>	Logical: If TRUE (default) require polygons to be closed in the sense that the last point should equal the first. Set this to FALSE to allow adding the first point as the last point.
<code>par</code>	A list of proj4 parameters.
<code>longlat</code>	Logical: If TRUE (default) the input to <code>polyArea_test</code> is longitude, latitude.
<code>drop</code>	Logical: If TRUE drop the list if only one multipolygon or only one polygon is given.
<code>data.frame.out</code>	Logical: If TRUE convert the matrices to data frames with columns x and y.
<code>inv</code>	Logical: If TRUE, do the inverse conversion in <code>rgdal::project()</code> .

### Value

`polyArea` returns area in nmi squared, and `matrix2multipolygon` returns a MULTIPOLYGON wkt.

### Examples

```
projectName <- "Test_Rstox"
g <- getBaseline(projectName, input="proc", proc=NULL, drop=FALSE)
# Get the stratum polygons:
multipolygon <- g$processData$stratumpolygon$Polygon
# Get stratum area in square nautical miles:
lapply(g$processData$stratumpolygon$Polygon, polyArea)
# Get cartesian locations using Azimuthal Equidistant projection (preserving distance):
###proj <- getProjString(multipolygon)
###xy <- lapply(g$processData$stratumpolygon$Polygon, geo2xy)
###xlim=range(unlist(lapply(xy, "[", , "x")))
###ylim=range(unlist(lapply(xy, "[", , "y")))
###plot(NULL, xlim=xlim, ylim=ylim)
###lapply(xy, lines, col='black', pbg='white')
###lapply(xy, polyArea, input="xy")
```