

Package ‘sigloc’

February 20, 2015

Version 0.0.4

Date 2013-04-23

Title Signal Location Estimation

Author Sergey S. Berg

Maintainer Sergey S. Berg <berg1546@umn.edu>

Depends R (>= 2.15.3), nleqslv, ellipse

Description A collection of tools for estimating the location of a transmitter signal from radio telemetry studies using the maximum likelihood estimation (MLE) approach described in Lenth (1981).

License GPL (>= 2)

URL <http://www.r-project.org>

NeedsCompilation no

Repository CRAN

Date/Publication 2014-10-22 23:14:50

R topics documented:

as.receiver	1
bear	3
findintersects	4
locate	5

Index	8
--------------	----------

as.receiver	<i>Coerce to Class 'receiver'</i>
-------------	-----------------------------------

Description

The function `as.receiver` coerces an object to class `receiver`.

`as.data.frame` can be used to convert an object inheriting the class `receiver` into a `data.frame`.

`plot` allows a graphical display of the receiver locations.

Usage

```

as.receiver(x)

## S3 method for class 'receiver'
plot(x, add, pch, cex, col, bearings, ...)

## S3 method for class 'receiver'
print(x, ...)

## S3 method for class 'receiver'
as.data.frame(x, row.names = NULL, optional = FALSE, ...)

```

Arguments

x	An object inheriting the class <code>data.frame</code> or <code>table</code> containing the X and Y location of the receiver, the bearings of the transmitter signal, and the identification of the point groupings.
add	A logical specification for whether to plot locations onto an existing plot window.
pch	Either an integer specifying a symbol or a single character to be used as the default in plotting points.
cex	A numerical value giving the amount by which plotting locations should be magnified to the default.
col	A specification for the default plotting color.
bearings	A logical specification for whether azimuth bearings should be plotted alongside locations.
row.names	NULL or a character vector giving the row names for the data frame. Missing values are not allowed.
optional	Logical. If TRUE, setting row names and converging column names is optional.
...	Additional parameters to be passed to the generic function <code>plot</code> , <code>print</code> , and <code>as.data.frame</code> .

Details

The function `as.receiver` coerces an object inheriting the class `data.frame` or `table` into an object inheriting the class `receiver`. This class is used mainly to explore and standardize the output and input of radio telemetry data. The `plot` function for this class allows for the user to plot the azimuth bearings associated with each of the receiver locations by setting the input parameter `bearings` to TRUE and to append the plotted objects to an existing plot window by setting the input parameter `add` to TRUE.

Value

The function `as.receiver` returns an object belonging to the S4 class `receiver`

Author(s)

Sergey S. Berg <berg1546@umn.edu>

See Also

[locate](#) for additional information on the class transmitter.
[findintersects](#) for additional information on the class intersect

Examples

```
## Load the data
data(bear)

## Convert data to class 'receiver'
bear<-as.receiver(bear)

## Display resulting object with bearings
plot(bear,bearings=TRUE,xlab="Easting",ylab="Northing")
```

bear

Bear Tracking Data

Description

This data gives the results of a radio telemetry study conducted in northern Minnesota on a four-year-old female black bear. Due to data sensitivity, all presented results, including names and locations, were slightly altered. These data should not be used for any purpose other than to understand the functionality of the package sigloc

Usage

```
data(bear)
```

Format

A data frame with 102 observations on the following 7 variables:

Date Data of observation

Observers Names of observers

GID Observation grouping number

Time Time of observation

Easting X location of observation using NAD1983 UTM Zone 15N

Northing X location of observation using NAD1983 UTM Zone 15N

Azimuth Azimuth bearing toward transmitter signal

Source

Berg, S. Unpublished Data.

Examples

```
## Load and display the data
data(bear)
```

findintersects	<i>Calculate Bearing Intersections</i>
----------------	--

Description

The function `findintersects` calculates the locations of the bearing intersections from radio telemetry studies.

`as.data.frame` can be used to convert an object inheriting the class `intersect` into a `data.frame`.

`plot` allows a graphical display of the calculated intersection locations.

Usage

```
findintersects(x)

## S3 method for class 'intersect'
plot(x, add, pch, cex, col, ...)

## S3 method for class 'intersect'
print(x, ...)

## S3 method for class 'intersect'
as.data.frame(x, row.names=NULL, optional=FALSE, ...)
```

Arguments

x	An object inhering the class <code>receiver</code> containing the X and Y location of the receiver, the bearing of the transmitter signal, and the identification of the point groupings. If <code>data</code> inherits the class <code>data.frame</code> or <code>table</code> , it should contain these columns under the headings 'Easting', 'Northing', 'Azimuth', and 'GID' respectively
add	A logical specification for whether to plot locations onto an existing plot window.
pch	Either an integer specifying a symbol or a single character to be used as the default in plotting points.
cex	A numerical value giving the amount by which plotting locations should be magnified to the default.

<code>col</code>	A specification for the default plotting color.
<code>row.names</code>	NULL or a character vector giving the row names for the data frame. Missing values are not allowed.
<code>optional</code>	Logical. If TRUE, setting row names and converging column names is optional.
<code>...</code>	Additional parameters to be passed to the generic function <code>plot</code> , <code>print</code> , and <code>as.data.frame</code> .

Details

This function uses data from radio telemetry studies to compute the location of bearing intersections using basic algebra. These intersections are a key component of the maximum likelihood estimation (MLE) approach described in Lenth (1981) that is used to estimate the location of the transmitter signal in such studies.

Value

The function `findintersects` returns an object belonging to the S4 class `intersect`

Author(s)

Sergey S. Berg <berg1546@umn.edu>

See Also

[locate](#) for additional information on the class `transmitter`.
[as.receiver](#) for additional information on the class `receiver`

Examples

```
## Load the data and convert to desired format
data(bear)
bear<-as.receiver(bear)

## Calculate the location of bearing intersections
(cross<-findintersects(bear))

## Display results
plot(cross,xlab="Easting",ylab="Northing")
```

Description

The function `locate` estimates the location of a transmitter signal from radio telemetry studies using the maximum likelihood estimation (MLE) approach.

`as.data.frame` can be used to convert an object inheriting the class `transmitter` into a `data.frame`.

`plot` allows a graphical display of the calculated transmitter signal locations.

Usage

```
locate(x)

## S3 method for class 'transmitter'
plot(x, add, errors, pch, cex, col, badcolor, ...)

## S3 method for class 'transmitter'
print(x, ...)

## S3 method for class 'transmitter'
as.data.frame(x, row.names=NULL, optional=FALSE, ...)
```

Arguments

<code>x</code>	An object inhering the class <code>receiver</code> containing the X and Y location of the receiver, the bearing of the transmitter signal, and the identification of the point groupings. If <code>data</code> inherits the class <code>data.frame</code> or <code>table</code> , it should contain these columns under the headings 'Easting', 'Northing', 'Bearing', and 'GID' respectively
<code>add</code>	A logical specification for whether to plot locations onto an existing plot window.
<code>errors</code>	A logical specification for whether error ellipses should be plotted alongside estimated locations.
<code>pch</code>	Either an integer specifying a symbol or a single character to be used as the default in plotting points.
<code>cex</code>	A numerical value giving the amount by which plotting locations should be magnified to the default.
<code>col</code>	A specification for the default plotting color.
<code>badcolor</code>	A logical specification for whether non-MLE-derived locations should be identified.
<code>row.names</code>	NULL or a character vector giving the row names for the data frame. Missing values are not allowed.
<code>optional</code>	Logical. If TRUE, setting row names and converging column names is optional.
<code>...</code>	Additional parameters to be passed to the generic function <code>plot</code> , <code>print</code> , and <code>as.data.frame</code> .

Details

This function uses data from radio telemetry studies to compute the location of a transmitter signal using the maximum likelihood estimation (MLE) approach described in Lenth (1981). If the MLE approach fails to provide a reasonable estimate of the transmitter signal location, the function will provide a warning message and instead use the midpoint of the intersections to estimate the location. This will also appear in the resulting object as a '2' under the `BadPoint` heading and can be displayed in the `plot` function by setting the input parameter `badcolor` to `TRUE`.

Value

The function `locate` returns an object belonging to the S4 class `transmitter`

Author(s)

Sergey S. Berg <berg1546@umn.edu>

References

Lenth, R.V. (1981). On Finding the Source of a Signal. *Technometrics*, 23(2), 149-154.

See Also

[as.receiver](#) for additional information on the class `receiver`.

[findintersects](#) for additional information on the class `intersect`

Examples

```
## Load the data and convert to desired format
data(bear)
bear<-as.receiver(bear)

## Estimate location of transmitter signal
(loc<-locate(bear))

## Display results with a different color for bad points
plot(loc,badcolor=TRUE,xlab="Easting",ylab="Northing")
```

Index

`as.data.frame.intersect`
 (`findintersects`), 4
`as.data.frame.receiver` (`as.receiver`), 1
`as.data.frame.transmitter` (`locate`), 5
`as.receiver`, 1, 5, 7

`bear`, 3

`findintersects`, 3, 4, 7

`locate`, 3, 5, 5

`plot.intersect` (`findintersects`), 4
`plot.receiver` (`as.receiver`), 1
`plot.transmitter` (`locate`), 5
`print.intersect` (`findintersects`), 4
`print.receiver` (`as.receiver`), 1
`print.transmitter` (`locate`), 5