Package 'moveWindSpeed'

October 19, 2016

Title Estimate Wind Speeds from Bird Trajectories

Version 0.1.0

Description Estimating wind speed form trajectories of individually tracked birds using a maximum likelihood approach.

Depends R (>= 3.0.0), methods, move

Suggests testthat, knitr, rmarkdown

License GPL

LazyData true

RoxygenNote 5.0.1

Imports Rcpp

LinkingTo Rcpp

VignetteBuilder knitr

NeedsCompilation yes

Author Bart Kranstauber [aut, cre],
Rolf Weinzierl [aut]

Maintainer Bart Kranstauber <bart.kranstauber@ieu.uzh.ch>

Repository CRAN

Date/Publication 2016-10-19 13:07:36

R topics documented:

getDefaultIsThermallingFunction
getDefaultIsThermallingFunction

Description

getDefaultIsThermallingFunction

Usage

```
getDefaultIsThermallingFunction(totalAngle = 360)
```

Arguments

totalAngle	the cumilative angle that is required to consider an trajectory thermaling

Value

a function is returned that based on a series of headings returns a logical value to indicate is a track is thermaling or not

Examples

```
fun<-getDefaultIsThermallingFunction(170)
fun(1:160)
fun(1:190)
```

getSamplingIsRegularFunction
getSamplingIsRegularFunction

Description

getSamplingIsRegularFunction

Usage

```
getSamplingIsRegularFunction(samplingIntervalSeconds)
```

Arguments

samplingIntervalSeconds	the interval that is considered regular

Value

a function is returned that based on a series of timestamps decides if the segment is regular

Examples

```
fun<-getSamplingIsRegularFunction(10)
fun(Sys.time()+1:5)
fun(Sys.time()+c(0,10,20,30))
fun(Sys.time()+c(0,10,20,31))
```

`getWindEstimate` *Estimate wind speed from a sample of ground speeds*

Description

Estimate wind speed from a sample of ground speeds

Usage

```
getWindEstimate(groundSpeeds, phi, windStart = c(0, 0))

## S4 method for signature 'matrix,numeric'
getWindEstimate(groundSpeeds, phi, windStart = c(0,
  0))
```

Arguments

`groundSpeeds`	matrix with two columns representing the ground speeds.
`phi`	numeric of length one giving the auto correlation.
`windStart`	numeric of length 2 giving the wind speed where to optimize from.

Value

an list with parameter estimates

Examples

```
s<-seq(0,2*pi, .1)
set.seed(34)
getWindEstimate(cbind(4*cos(s)+3+rnorm(length(s)), 4*sin(s)+2+rnorm(length(s))),0)
getWindEstimate(cbind(4*cos(s)+3+rnorm(length(s),sd=.2), 4*sin(s)+2+rnorm(length(s),sd=.2)),0)
```

getWindEstimates *Title*

Description

Title

Usage

```
getWindEstimates(data, timestamps, ...)

## S4 method for signature 'MoveStack,missing'
getWindEstimates(data, timestamps, ...)

## S4 method for signature 'Move,missing'
getWindEstimates(data, timestamps, ...)

## S4 method for signature 'data.frame,POSIXct'
getWindEstimates(data, timestamps,
  windowSize = 29, phi = 0.5, isFocalPoint = function(i, ts) {     TRUE },
  isSamplingRegular = 1,
  hasVariationInHeadingFunction = getDefaultIsThermallingFunction(360),
  columnNamesWind = c("estimationSuccessful", "residualVarAirspeed", "windX",
  "windY", "windVarX", "windVarY", "windCovarXY", "windVarMax",
  "isThermalling"), ...)
```

Arguments

data	Move object, MoveStack or data.frame containing wind speeds
timestamps	timestamps of the speed observations
...	other possible arguments currently nothing else is implented
windowSize	a numeric vector of length 1 or 2, if length 1 it is the size of the focal window data will be assigned to the central location. If length 2 the window size is sum(windowSize)+1) and the first element is the number of location before the focal locations, the second is the number of locations after the focal location.
phi	todo
isFocalPoint	todo
isSamplingRegular	either a function that determines based on a vector of timestamps if the sampling interval is regular or a numeric value that corresponds to the time interval between observations in the dataset that is regular
hasVariationInHeadingFunction	todo
columnNamesWind	todo

Value

a Move object, dataframe or a MoveStack depending on input

Examples

```
data("storks")
# run example for reduced dataset
windEst<-getWindEstimates(storks[format(timestamps(storks),"%H")=="12",][[2:3]])
windEst<-spTransform(windEst, center=TRUE)
plot(windEst)
# only plot few arrows of estimates
s<-windEst$estimationSuccessful & format(timestamps(windEst), "%S")=='00'
# enlarge arrows 30 times
arrows(coordinates(windEst)[s,1],coordinates(windEst)[s,2],
   coordinates(windEst)[s,1]+ windEst$windX[s]*30,
   coordinates(windEst)[s,2]+windEst$windY[s]*30)
```

---

storks *Example stork data.*

Description

A dataset containing location data of 6 juvinile storks on the 18th of august when migration just started

Usage

```
storks
```

Format

A MoveStack consiting of 22333 locations

Source

http://www.movebank.org/

Examples

```
data("storks")
```

Index