

Package ‘traipse’

July 2, 2019

Title Shared Tools for Tracking Data

Version 0.1.0

Description A collection of commonly used tools for animal movement and other tracking data. Various distance, angle, bearing, distance-to, bearing-to and speed are provided for geographic data that can be used directly or within 'tidyverse' syntax. Distances and bearings are calculated using modern geodesic methods as provided by Charles F. F. Karney (2013) <doi:10.1007/s00190-012-0578-z> via the 'geodist' and 'geosphere' packages.

License GPL-3

Encoding UTF-8

LazyData true

RoxygenNote 6.1.1

Depends R (>= 2.10)

Imports magrittr, geodist, geosphere

URL <https://github.com/Trackage/traipse>

BugReports <https://github.com/Trackage/traipse/issues>

Suggests covr, testthat (>= 2.1.0), dplyr, tibble, spelling

Language en-US

NeedsCompilation no

Author Michael Sumner [aut, cre] (<<https://orcid.org/0000-0002-2471-7511>>)

Maintainer Michael Sumner <mdsumner@gmail.com>

Repository CRAN

Date/Publication 2019-07-02 12:10:02 UTC

R topics documented:

track_angle	2
track_bearing	3
track_bearing_to	3

track_distance	4
track_distance_to	5
track_speed	6
track_time	6
track_turn	7
trips0	8

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

track_angle	<i>Track angle</i>
-------------	--------------------

Description

Calculate internal track angle on longitude, latitude input vectors.

Usage

```
track_angle(x, y)
```

Arguments

x	longitude
y	latitude

Details

By convention the first and last values are set to NA missing value, because the angle applies to the location between each previous and next location.

To use this on multiple track ids, use a grouped data frame with tidyverse code like `data %>% group_by(id) %>% mutate(an`

The maximum possible value is 180 and the minimum is 0.

Value

a numeric vector of the relative internal angle between sequential locations in degrees, see Details

Examples

```
track_angle(trips0$x, trips0$y)[1:10]

## maximum value
track_angle(c(0, 0, 0), c(0, 1, 2))
## minimum value
track_angle(c(0, 0, 0), c(0, 1, 0))
```

track_bearing	<i>Track bearing</i>
---------------	----------------------

Description

Calculate sequential bearing on longitude, latitude input vectors.

Usage

```
track_bearing(x, y)
```

Arguments

x	longitude
y	latitude

Details

By convention the last value is set to NA missing value, because the bearing applies to the segment extending from the current location.

To use this on multiple track ids, use a grouped data frame with tidyverse code like `data %>% group_by(id) %>% mutate(tu`

Absolute bearing is relative to North (0), and proceeds clockwise positive and anti-clockwise negative N = 0, E = 90, S = +/-180, W = -90.

The last value will be NA as the bearing is relative to the first point of each segment.

Value

a numeric vector of absolute bearing in degrees, see Details

Examples

```
track_bearing(trips0$x, trips0$y)[1:10]
```

track_bearing_to	<i>Track bearing to location/s</i>
------------------	------------------------------------

Description

Calculate geodesic bearing to a location or locations based on longitude, latitude (from) input vectors and longitude, latitude (to) input vectors. The *to* values may be a single value or individual to each *from* location.

Usage

```
track_bearing_to(x, y, to_x, to_y)
```

Arguments

x	longitude
y	latitude
to_x	longitude vector of <i>to</i> location/s
to_y	latitude vector of <i>to</i> locations/s

Details

No missing values are required as padding, but input data with NAs will incur an NA in the output.

To use this on multiple track ids, use a grouped data frame with tidyverse code like `data %>% group_by(id) %>% mutate(bearing = track_bearing_to(x, y, to_x, to_y))`

Absolute bearing is relative to North (0), and proceeds clockwise positive and anti-clockwise negative N = 0, E = 90, S = +/-180, W = -90.

There is no NA padding in the output value (though missing values in the input will be mirrored in the output).

Value

a numeric vector of absolute bearing-to in degrees, see Details

Examples

```
track_bearing_to(trips0$x, trips0$y, to_x = 147, to_y = -42)[1:10]
# N E S W
track_bearing_to(0,0, c(0, 10, 0, -10), c(5, 0, -5, 0))

# maximum and minimum value are the same direction (due south)
track_bearing(c(0, -0.00001), c(0, -1))
track_bearing(c(0, 0.00001), c(0, -1))

# the absolute minimum is north
track_bearing(c(0, 0), c(0, 1))
```

track_distance	<i>Track distance</i>
----------------	-----------------------

Description

Calculate geodesic distance on longitude, latitude input vectors.

Usage

```
track_distance(x, y)
```

Arguments

x	longitude
y	latitude

Details

By convention the first value is set to NA missing value, because the distance applies to each sequential pair of locations.

To use this on multiple track ids, use a grouped data frame with tidyverse code like `data %>% group_by(id) %>% mutate(di`

Value

numeric vector of distances between sequential pairs of x, y in metres, see Details

Examples

```
track_distance(trips0$x, trips0$y)[1:10]
```

track_distance_to	<i>Track distance to location/s</i>
-------------------	-------------------------------------

Description

Calculate geodesic distance to a location or locations based on longitude, latitude (from) input vectors and longitude, latitude (to) input vectors. The *to* values may be a single value or individual to each *from* location.

Usage

```
track_distance_to(x, y, to_x, to_y)
```

Arguments

x	longitude
y	latitude
to_x	longitude vector of <i>to</i> location/s
to_y	latitude vector of <i>to</i> locations/s

Details

No missing values are required as padding, but input data with NAs will incur an NA in the output.

To use this on multiple track ids, use a grouped data frame with tidyverse code like `data %>% group_by(id) %>% mutate(di`

Value

a numeric vector of distance-to values in metres

Examples

```
track_distance_to(trips0$x, trips0$y, to_x = 147, to_y = -42)[1:10]
```

track_speed	<i>Track speed</i>
-------------	--------------------

Description

Calculate speed (m/s) based on geodesic distance with longitude, latitude, date-time input vectors.

Usage

```
track_speed(x, y, date)
```

Arguments

x	longitude
y	latitude
date	date-time in POSIXct

Details

By convention the first value is set to NA missing value, because the difference applies to each sequential pair of locations.

To use this on multiple track ids, use a grouped data frame with tidyverse code like `data %>% group_by(id) %>% mutate(sp`

Value

numeric vector of sequential distances in metres per second, see Details

Examples

```
track_speed(trips0$x, trips0$y, trips0$date)[1:10]
```

track_time	<i>Track time duration</i>
------------	----------------------------

Description

Calculate time duration based on sequential difference of date-time input.

Usage

```
track_time(date)
```

Arguments

date	date-time in POSIXct
------	----------------------

Details

By convention the first value is set to NA missing value, because the difference applies to each sequential pair of locations.

To use this on multiple track ids, use a grouped data frame with tidyverse code like `data %>% group_by(id) %>% mutate(du`

Value

numeric vector of duration between sequential date-time values in seconds, see Details

Examples

```
track_time(trips0$date)[1:10]
```

track_turn	<i>Track turn angle</i>
------------	-------------------------

Description

Calculate relative track turning angle on longitude, latitude input vectors.

Usage

```
track_turn(x, y)
```

Arguments

x	longitude
y	latitude

Details

By convention the last value is set to NA missing value, because the angle applies to the relative turn from the current location.

To use this on multiple track ids, use a grouped data frame with tidyverse code like `data %>% group_by(id) %>% mutate(tu`

The maximum possible value is 180 degrees and the minimum is -180, although these particular values are a special case and will probably always be positive. Turn angle is a signed quantity with negative values for a left turn and positive values for a right turn.

Value

a numeric vector of absolute turn angles, in degrees

Examples

```
track_turn(trips0$x, trips0$y)[1:10]

## maximum turn angle
track_turn(c(0, 0, 0), c(0, 1, 0))
## minimum turn angle
track_turn(c(0, 0, 0), c(0, 1, 2))
```

trips0

Simulated track data

Description

trips0 is an ungrouped data frame of x, y, date, id

Index

track_angle, 2
track_bearing, 3
track_bearing_to, 3
track_distance, 4
track_distance_to, 5
track_speed, 6
track_time, 6
track_turn, 7
trips0, 8