

Package ‘trackdf’

August 4, 2019

Type Package

Title Data Frame Class for Tracking Data

Version 0.2.1

Maintainer Simon Garnier <garnier@njit.edu>

Description Data frame class for storing collective movement data (e.g. fish schools, ungulate herds, baboon troops) collected from GPS trackers or computer vision tracking software.

License GPL-3

Encoding UTF-8

LazyData true

RoxygenNote 6.1.1

Depends R (>= 3.5.0)

Imports tibble, data.table, rgdal, sp, lubridate, methods

Suggests readr, dplyr, ggplot2, knitr, rmarkdown, adehabitatLT, move, moveVis, ctm, moveHMM, mapproj

VignetteBuilder knitr

URL <https://swarm-lab.github.io/trackdf/>,
<https://github.com/swarm-lab/trackdf>

BugReports <https://github.com/swarm-lab/trackdf/issues>

NeedsCompilation no

Author Simon Garnier [aut, cre] (<<https://orcid.org/0000-0002-3886-3974>>)

Repository CRAN

Date/Publication 2019-08-04 14:20:02 UTC

R topics documented:

.reclass	2
conversions	3
dplyr_track	5

is_geo	6
is_track	7
n_dims	7
n_tracks	8
projection	9
rbind_track	10
short_tracks	10
tracks	11
track_	11
[.track	13

Index	15
--------------	-----------

.reclass	<i>Maintain Class After Modification</i>
----------	--

Description

Copy class and attributes from the original version of an object to a modified version.

Usage

```
.reclass(x, result)
```

Arguments

x	The original object, which has a class/attributes to copy
result	The modified object, which is / might be missing the class/attributes.

Value

result, now with class/attributes restored.

Author(s)

Simon Garnier, <garnier@njit.edu>

conversions	<i>Convert a Track Table to/from Other Formats</i>
-------------	--

Description

The following methods will convert track tables to and from other common formats used for processing tracking and spatial data.

Usage

```
as_track(x, table = "df", ...)

## S3 method for class 'MoveStack'
as_track(x, table = "df", ...)

## S3 method for class 'Move'
as_track(x, table = "df", ...)

as_move(x, ...)

## S3 method for class 'track'
as_move(x, ...)

## S3 method for class 'SpatialPointsDataFrame'
as_track(x, table = "df", ...)

as_sp(x, ...)

## S3 method for class 'track'
as_sp(x, ...)

## S3 method for class 'ltraj'
as_track(x, table = "df", ...)

as_ltraj(x, ...)

## S3 method for class 'track'
as_ltraj(x, ...)

## S3 method for class 'telemetry'
as_track(x, table = "df", ...)

## S3 method for class 'list'
as_track(x, table = "df", ...)

as_telemetry(x, ...)
```

```
## S3 method for class 'track'
as_telemetry(x, ...)

## S3 method for class 'moveData'
as_track(x, table = "df", type = c("LL", "UTM"),
  ...)

as_moveHMM(x, ...)

## S3 method for class 'track'
as_moveHMM(x, ...)
```

Arguments

x	An object to convert.
table	A string indicating the class of the table on which the track table should be built. It can be a <code>data.frame</code> ("df", the default), a <code>tibble</code> ("tbl"), or a <code>data.table</code> ("dt").
...	Other parameters to be passed to: <ul style="list-style-type: none"> • <code>track_df</code>, <code>track_tbl</code> or <code>track_dt</code> if <code>as_track</code> is used. • <code>moveVis::df2move</code> if <code>as_move</code> is used. • <code>sp::SpatialPointsDataFrame</code> if <code>as_sp</code> is used. • <code>adehabitatLT::as.ltraj</code> if <code>as_ltraj</code> is used. • <code>ctmm::as.telemetry</code> if <code>as_telemetry</code> is used. • <code>moveHMM::prepData</code> if <code>as_moveHMM</code> is used.
type	For converting <code>moveHMM::moveData</code> to track table only, a character string indicating the type of coordinates stored in the <code>moveHMM::moveData</code> object: "LL" if longitude/latitude (default), "UTM" if easting/northing.

Value

The coordinates converted in the chosen format.

Author(s)

Simon Garnier, <garnier@njit.edu>

See Also

[track_df](#), [track_tbl](#), [track_dt](#)

Examples

```
data(short_tracks)

if (requireNamespace("moveVis", quietly = TRUE)) {
  mv <- as_move(short_tracks)
  as_track(mv)
```

```
}

if (requireNamespace("sp", quietly = TRUE)) {
  sp <- as_sp(short_tracks)
  as_track(sp)
}

if (requireNamespace("adehabitatLT", quietly = TRUE)) {
  lt <- as_ltraj(short_tracks)
  as_track(lt)
}

if (requireNamespace("ctmm", quietly = TRUE)) {
  tl <- as_telemetry(short_tracks)
  as_track(tl)
}

if (requireNamespace("moveHMM", quietly = TRUE)) {
  hhm <- as_moveHMM(short_tracks, type = "LL")
  as_track(hhm)
}
```

dplyr_track

Dplyr Methods For Track Tables

Description

`dplyr` methods for track tables objects.

Usage

```
select.track(.data, ...)

rename.track(.data, ...)

filter.track(.data, ...)

arrange.track(.data, ...)

mutate.track(.data, ...)

transmute.track(.data, ...)

summarise.track(.data, ...)

summarize.track(.data, ...)

group_by.track(.data, ...)
```

```
ungroup.track(.data, ...)  
sample_n.track(.data, ...)  
sample_frac.track(.data, ...)  
do.track(.data, ...)
```

Arguments

<code>.data</code>	A track table.
<code>...</code>	Additional arguments to be passed to the corresponding <code>dplyr</code> method.

See Also

[select](#), [rename](#), [filter](#), [arrange](#), [mutate](#), [transmute](#), [summarise](#), [summarize](#), [group_by](#), [ungroup](#), [sample_n](#), [sample_frac](#), [do](#)

is_geo	<i>Check if Track Table Uses Geographic Coordinates</i>
--------	---

Description

Track tables produced by `track_df` can use a cartesian (x, y, z) or a geographic (latitude, longitude, altitude) coordinate system. This function helps determine which is being used in a particular table.

Usage

```
is_geo(x)
```

Arguments

<code>x</code>	A track data table as produced by <code>track_df</code> .
----------------	---

Value

A logical.

Author(s)

Simon Garnier, <garnier@njit.edu>

Examples

```
data(tracks)  
is_geo(tracks)
```

is_track	<i>Check Validity of Track Table</i>
----------	--------------------------------------

Description

Test whether a variable contains a track table as produced by [track_df](#), [track_tbl](#), or [track_dt](#).

Usage

```
is_track(x)
```

Arguments

x An object to test.

Value

A logical indicating whether the variable contains a track table (TRUE) or not (FALSE).

Author(s)

Simon Garnier, <garnier@njit.edu>

See Also

[track_df](#), [track_tbl](#), [track_dt](#)

Examples

```
data(tracks)
is_track(tracks)
```

n_dims	<i>Number of Spatial Dimensions of a Track Table</i>
--------	--

Description

Track tables produced by [track_df](#) can have 2 (x,y) or 3 (x, y, z) spatial dimensions. This function returns the number of spatial dimensions of a track table.

Usage

```
n_dims(x)
```

Arguments

x A track data table as produced by [track_df](#).

Value

A numeric value.

Author(s)

Simon Garnier, <garnier@njit.edu>

Examples

```
data(tracks)
```

```
n_dims(tracks)
```

n_tracks	<i>Number of Tracks in a Track Table</i>
----------	--

Description

Track tables produced by [track_df](#) can contain multiple tracks (e.g., from different animals). This function returns the number of tracks in a track table.

Usage

```
n_tracks(x)
```

Arguments

x A track data table as produced by [track_df](#).

Value

A numeric value.

Author(s)

Simon Garnier, <garnier@njit.edu>

Examples

```
data(tracks)
```

```
n_tracks(tracks)
```

projection	<i>Access/Modify the Projection of a Track Table</i>
------------	--

Description

Functions to access or modify the projection of a data table. Changing the projection will trigger automatically the conversion of the locations in the new coordinate system.

Usage

```
projection(x)
projection(x) <- value
project(x, value)
```

Arguments

x	A track table.
value	A character string or a <code>sp::CRS</code> object representing the projection of the coordinates. "+proj=longlat" is suitable for the outputs of most GPS trackers.

Value

A track table.

Note

It is not possible to modify the projection if missing coordinates are present.

Author(s)

Simon Garnier, <garnier@njit.edu>

Examples

```
data(tracks)

projection(tracks)
tracks_projected <- project(tracks, "+proj=somerc")
projection(tracks_projected)
projection(tracks_projected) <- "+proj=longlat"
projection(tracks_projected)
```

`rbind_track`*Bind Multiple Track Tables by Row*

Description

`rbind_track` uses `data.table::rbindlist` to combine track tables by rows, but makes sure that you cannot bind together two tables with different projections, that the projection attribute is inherited by the resulting track table, and that track tables based on different table classes are coerced to the same table class.

Usage

```
rbind_track(...)
```

Arguments

... Track tables to combine. Each argument can either be a track table or a list of track tables. The track tables must have the same projection.

Value

A track table.

Author(s)

Simon Garnier, <garnier@njit.edu>

Examples

```
data(tracks)

rbind_track(tracks, tracks)
rbind_track(list(tracks, tracks))
```

`short_tracks`*Trajectories of Two Goats Through the Namibian Desert (short version)*

Description

A dataset containing the trajectories of two goats through the Namibian desert.

Usage

```
short_tracks
```

Format

A track table with 18 rows and 4 variables:

id Identity of the goat

t Time of the observation

x Longitude

y Latitude

tracks

Trajectories of Two Goats Through the Namibian Desert

Description

A dataset containing the trajectories of two goats through the Namibian desert.

Usage

tracks

Format

A track table with 7194 rows and 4 variables:

id Identity of the goat

t Time of the observation

x Longitude

y Latitude

track_

Build a Track Table

Description

track constructs track tables based on `data.frame` (the default), `tibble`, or `data.table`. track is a convenience function that executes `track_df`, `track_tbl`, or `track_dt` based on the value of the 'table' parameter. Track tables can be used like the data structure they are build upon but with a notable difference: they have an extra attribute to store the projection of the track coordinates, and modifying the projection will automatically trigger the appropriate conversion of the coordinates.

Usage

```
track(x, y, z, t, id, ..., proj, origin, period, tz, table = "df")
```

```
track_df(x, y, z, t, id, ..., proj, origin, period, tz)
```

```
track_tbl(x, y, z, t, id, ..., proj, origin, period, tz)
```

```
track_dt(x, y, z, t, id, ..., proj, origin, period, tz)
```

Arguments

<code>x, y, z</code>	Numeric vectors representing the coordinates of the locations. <code>x</code> and <code>y</code> are required. <code>z</code> can be ignored if the trajectories are 2-dimensional. Note: if the vectors are not of the same length, the shorter ones will be recycled to match the length of the longer one.
<code>t</code>	A numeric vector or a vector of objects that can be coerced to date-time objects by <code>link[lubridate]{as_datetime}</code> representing the times (or frames) at which each location was recorded. If numeric, the origin and period of the time points can be set using <code>origin</code> and <code>period</code> below.
<code>id</code>	A vector that can be coerced to a character vector by <code>as.character</code> representing the identity of the animal to which each location belong.
<code>...</code>	A set of name-value pairs. Arguments are evaluated sequentially, so you can refer to previously created elements. These arguments are processed with <code>rlang::quos()</code> and support unquote via <code>!!</code> and unquote-splice via <code>!!!</code> . Use <code>:=</code> to create columns that start with a dot.
<code>proj</code>	A character string or a <code>sp::CRS</code> object representing the projection of the coordinates. Leave empty if the coordinates are not projected (e.g., output of video tracking). <code>"+proj=longlat"</code> is suitable for the outputs of most GPS trackers.
<code>origin</code>	Something that can be coerced to a date-time object by <code>link[lubridate]{as_datetime}</code> representing the start date and time of the observations when <code>t</code> is a numeric vector.
<code>period</code>	A character vector in a shorthand format (e.g. "1 second") or ISO 8601 specification. This is used when <code>t</code> is a numeric vector to represent time unit of the observations. All unambiguous name units and abbreviations are supported, "m" stands for months, "M" for minutes unless ISO 8601 "P" modifier is present (see examples). Fractional units are supported but the fractional part is always converted to seconds. See <code>period</code> for more details.
<code>tz</code>	A time zone name. See <code>OlsonNames</code> .
<code>table</code>	A string indicating the class of the table on which the track table should be built. It can be a <code>data.frame</code> ("df", the default), a <code>tibble</code> ("tbl"), or a <code>data.table</code> ("dt").

Value

A track table, which is a colloquial term for an object of class `track`.

Author(s)

Simon Garnier, <garnier@njit.edu>

Examples

```
data(tracks)

t_df <- track(x = tracks$x, y = tracks$y, t = tracks$t, id = tracks$id,
  proj = "+proj=longlat", tz = "Africa/Windhoek", table = "df")

t_df <- track_df(x = tracks$x, y = tracks$y, t = tracks$t, id = tracks$id,
  proj = "+proj=longlat", tz = "Africa/Windhoek")

t_tbl <- track_tbl(x = tracks$x, y = tracks$y, t = tracks$t, id = tracks$id,
  proj = "+proj=longlat", tz = "Africa/Windhoek")

t_dt <- track_dt(x = tracks$x, y = tracks$y, t = tracks$t, id = tracks$id,
  proj = "+proj=longlat", tz = "Africa/Windhoek")
```

[.track

*Extract or Replace Parts of a Track Table***Description**

Accessing columns, rows, or cells via \$, [[, or [is mostly similar to regular [data frames](#). However, the behavior is sometimes different for track tables based on [tibble](#) and [data.table](#). For more info, refer to [tibble](#)'s and [data.table](#)'s subsetting documentation.

Usage

```
## S3 method for class 'track'
x[...]

## S3 replacement method for class 'track'
x[...] <- value
```

Arguments

x	A track table.
...	Other parameters to be passed to the extracting/subsetting functions of data.frame , tibble , and data.table .
value	A suitable replacement value: it will be repeated a whole number of times if necessary and it may be coerced: see the 'Coercion' section in data.frame . If 'NULL', deletes the column if a single column is selected.

Value

A subset of the track table is [is called, or a modified version of the track table if [<- is called.

Author(s)

Simon Garnier, <garnier@njit.edu>

See Also

[track_df](#), [track_tbl](#), [track_dt](#)

Examples

```
data(tracks)

tracks[1]
tracks[1, ]
tracks[1, 1]
tracks$id[tracks$id == "1"] <- "0"
tracks[tracks[, 1] == "0", 1] <- "1"
```

Index

*Topic **datasets**

short_tracks, 10

tracks, 11

.reclass, 2

[.track, 13

[<-.track ([.track), 13

adehabitatLT::as.ltraj, 4

arrange, 6

arrange.track (dplyr_track), 5

as.character, 12

as_ltraj (conversions), 3

as_move (conversions), 3

as_moveHMM (conversions), 3

as_sp (conversions), 3

as_telemetry (conversions), 3

as_track (conversions), 3

conversions, 3

ctmm::as.telemetry, 4

data frames, 13

data.frame, 4, 11–13

data.table, 4, 11–13

data.table::rbindlist, 10

do, 6

do.track (dplyr_track), 5

dplyr, 5, 6

dplyr_track, 5

filter, 6

filter.track (dplyr_track), 5

group_by, 6

group_by.track (dplyr_track), 5

is_geo, 6

is_track, 7

moveHMM::moveData, 4

moveHMM::prepData, 4

moveVis::df2move, 4

mutate, 6

mutate.track (dplyr_track), 5

n_dims, 7

n_tracks, 8

OlsonNames, 12

period, 12

project (projection), 9

projection, 9

projection<- (projection), 9

rbind_track, 10

rename, 6

rename.track (dplyr_track), 5

rlang::quos(), 12

sample_frac, 6

sample_frac.track (dplyr_track), 5

sample_n, 6

sample_n.track (dplyr_track), 5

select, 6

select.track (dplyr_track), 5

short_tracks, 10

sp::CRS, 9, 12

sp::SpatialPointsDataFrame, 4

summarise, 6

summarise.track (dplyr_track), 5

summarize, 6

summarize.track (dplyr_track), 5

tibble, 4, 11–13

track (track_), 11

track_, 11

track_df, 4, 6–8, 14

track_df (track_), 11

track_dt, 4, 7, 14

track_dt (track_), 11

track_tbl, 4, 7, 14

`track_tbl (track_)`, 11
`tracks`, 11
`transmute`, 6
`transmute.track (dplyr_track)`, 5

`ungroup`, 6
`ungroup.track (dplyr_track)`, 5