Package 'ezTrack'

May 8, 2026

Type Package

Title Exploring Animal Movement Data

Version 0.1.0

Description Streamlines common steps for working with animal tracking data, from raw telemetry points to summaries, interactive maps, and home range estimates. Designed to be beginner-friendly, it enables rapid exploration of spatial and movement data with minimal wrangling, providing a unified workflow for importing, summarizing, and visualizing, and analyzing animal movement datasets.

License MIT + file LICENSE

Encoding UTF-8

LazyData true

RoxygenNote 7.3.2

Depends R (>= 4.1.0)

Imports methods, sf, sp, geosphere, leaflet, adehabitatHR, magrittr, htmltools, utils, stats, ggplot2, readxl, kableExtra, dplyr

VignetteBuilder knitr

Suggests knitr, viridisLite, rmarkdown

BugReports https://github.com/taylorbcraft/ezTrack/issues

URL https://github.com/taylorbcraft/ezTrack

NeedsCompilation no

Author Taylor Craft [aut, cre]

Maintainer Taylor Craft <taylor.craft.mail@gmail.com>

Repository CRAN

Date/Publication 2025-09-16 06:40:07 UTC

Contents

`ez_fix_rate_plot` *Plot Fix Rate Timelines for Each Individual*

Description

Creates a timeline plot showing fix events as horizontal tick marks for each individual. This allows a quick visual assessment of tracking effort, fix density, and coverage gaps.

Usage

```
ez_fix_rate_plot(
  data,
  date_breaks = NULL,
  date_format = NULL,
  start_date = NULL,
  end_date = NULL
)
```

Arguments

`data`	A data frame or ‘sf‘ object with columns ‘id‘ and ‘timestamp‘.
`date_breaks`	Optional. Spacing of x-axis breaks (e.g., ‘"1 month"‘, ‘"2 weeks"‘). If ‘NULL‘, ggplot2 chooses automatically.
`date_format`	Optional. Format string for x-axis date labels (e.g., "%b %Y", "%d-%m"). If ‘NULL‘, ggplot2 chooses automatically.
`start_date`	Optional. A ‘Date‘ or string (e.g., ‘"2024-01-01"‘). Filters out fixes before this date.
`end_date`	Optional. A ‘Date‘ or string (e.g., ‘"2024-12-31"‘). Filters out fixes after this date.

Value

A ‘ggplot‘ object showing fix timelines per individual.

Examples

```
df <- data.frame(
  id = rep(c("a", "b"), each = 3),
  timestamp = as.POSIXct(c("2025-01-01", "2025-01-02", "2025-01-03",
                           "2025-01-01", "2025-01-05", "2025-01-06"))
)
ez_fix_rate_plot(df)
```

---

ez_home_range *Estimate Home Ranges for Tracked Individuals or Population*

Description

Computes home range polygons using either Minimum Convex Polygon (MCP) or Kernel Density Estimation (KDE).

Usage

```
ez_home_range(
  data,
  method = "mcp",
  level = 95,
  crs = NULL,
  population = FALSE,
  start_date = NULL,
  end_date = NULL,
  kde_extent = 1,
  h = "href",
  hlim = NULL
)
```

Arguments

data	A data frame or ‘sf‘ object with columns ‘id‘, ‘timestamp‘, ‘x‘, and ‘y‘.
method	Method for home range estimation. One of ‘"mcp"‘ (default) or ‘"kde"‘.
level	Percentage of points to include in the home range (e.g., 95 for 95%). Default is 95.
crs	Optional CRS to project the data before calculation. If NULL, uses EPSG:3857 (Web Mercator).
population	Logical. If TRUE, returns a single home range polygon for all data combined.
start_date	Optional. A ‘Date‘ object or string ("YYYY-MM-DD"). Filters out data before this date.
end_date	Optional. A ‘Date‘ object or string ("YYYY-MM-DD"). Filters out data after this date.

kde_extent	Numeric. When method = "kde", passed to ‘adehabitatHR::kernelUD()‘ to control the extent of the grid for KDE. Default is 1.
h	Bandwidth method when method = "kde". One of ‘"href"‘ (default), "LSCV", or a numeric value.
hlim	Optional vector of length 2 passed to ‘adehabitatHR::kernelUD()‘ when method = "kde" to constrain the bandwidth search (used with h = "LSCV").

Value

An ‘sf‘ object of home range polygon(s).

`ez_latitude_plot` *Plot Latitude Over Time*

Description

Creates a time series plot of latitude (y-axis) over timestamp (x-axis), with separate lines for each individual animal. Optionally facets the plot by animal and allows customization of x-axis date format and break spacing.

Usage

```
ez_latitude_plot(
  data,
  color_palette = "turbo",
  facet = FALSE,
  date_format = NULL,
  date_breaks = NULL,
  start_date = NULL,
  end_date = NULL
)
```

Arguments

data	A data frame or ‘sf‘ object with columns ‘id‘, ‘timestamp‘, and ‘y‘ (latitude).
color_palette	Character. Viridis palette option: "viridis", "magma", "plasma", "inferno", "cividis", or "turbo". Default is "turbo".
facet	Logical. If TRUE, creates a separate facet panel for each animal. Default is FALSE.
date_format	Optional. Format for date labels on the x-axis (e.g., "%b %d", "%Y-%m", "%H:%M"). Default is automatic.
date_breaks	Optional. Interval for x-axis breaks (e.g., "1 day", "2 weeks"). Default is automatic.
start_date	Optional. A ‘Date‘ object or string (e.g., "2023-01-01"). Filters out data before this date.
end_date	Optional. A ‘Date‘ object or string (e.g., "2023-02-01"). Filters out data after this date.

Value

A ggplot object.

---

ez_map *Plot Tracking Data and Home Ranges Together*

Description

A unified function for visualizing tracking data and home range polygons on an interactive map. Accepts tracking points (from ‘ez_track‘) and home ranges (from ‘ez_home_range‘), and allows layer-specific customization of appearance and filtering by time.

Usage

```
ez_map(
  tracks = NULL,
  home_ranges = NULL,
  individual = NULL,
  show_points = TRUE,
  show_paths = TRUE,
  point_color = "id",
  point_size = 4,
  point_opacity = 0.8,
  point_stroke = TRUE,
  point_stroke_color = "black",
  path_color = "id",
  path_width = 2,
  path_opacity = 1,
  polygon_color = "id",
  polygon_opacity = 0.4,
  color_palette = "viridis",
  show_labels = TRUE,
  start_date = NULL,
  end_date = NULL
)
```

Arguments

tracks	Output from ‘ez_track()‘ — a data frame or ‘sf‘ with ‘id‘, ‘timestamp‘, ‘x‘, ‘y‘.
home_ranges	Output from ‘ez_home_range()‘ — an ‘sf‘ polygon object.
individual	Optional. Character or character vector of individual ID(s) to display.
show_points	Logical, whether to draw individual locations. Default TRUE.
show_paths	Logical, whether to draw movement paths. Default TRUE.
point_color	Column name or static color for points. Default "id". Can also be "timestamp" for temporal gradient.

point_size	Radius of point markers. Default 4.
point_opacity	Fill opacity of points. Default 0.8.
point_stroke	Logical, whether to draw point borders. Default TRUE.
point_stroke_color	Border color for points. Default "black".
path_color	Column name or static color for paths. Default "id".
path_width	Line width of movement paths. Default 2.
path_opacity	Opacity of path lines. Default 1.
polygon_color	Column name or static color for home range polygons. Default "id".
polygon_opacity	Fill opacity of polygons. Default 0.4.
color_palette	Color palette to use for all mapped variables. Default "viridis".
show_labels	Logical, whether to show hover labels for locations. Default TRUE.
start_date	Optional filter (Date or string) to remove data before this date.
end_date	Optional filter (Date or string) to remove data after this date.

Value

A ‘leaflet‘ map object.

Examples

```
# Simulate tracks
tracks <- data.frame(
  id = rep(c("A", "B"), each = 10),
  timestamp = rep(seq.POSIXt(as.POSIXct("2023-01-01"), by = "1 day", length.out = 10), 2),
  x = c(runif(10, -1, 1), runif(10, 0, 2)),
  y = c(runif(10, 51, 52), runif(10, 51.5, 52.5))
)

# Plot map
ez_map(tracks, point_color = "timestamp")
```

---

ez_summary *Summarize Animal Tracking Data*

Description

Calculate basic summary statistics per tracked individual. This function is useful for quickly understanding data coverage, gaps, and movement distance for each animal. The following summary statistics are returned for each unique ‘id‘: - ‘n_fixes‘: Number of location records - ‘first_location‘: Timestamp of the first recorded location - ‘last_location‘: Timestamp of the last recorded location - ‘tracking_duration_days‘: Duration between first and last fix (in days) - ‘fixes_per_day‘: Average number of fixes per day - ‘median_interval_hours‘: Median interval between fixes (in hours) - ‘max_time_gap_days‘: Longest time gap between consecutive fixes (in days) - ‘distance_km‘: Total distance traveled (in kilometers), calculated using the Haversine formula - ‘avg_speed_kmh‘: Average speed (km/h), computed as distance divided by tracking duration in hours

Usage

```
ez_summary(data, start_date = NULL, end_date = NULL, report = FALSE)
```

Arguments

data	A data frame or sf object with columns ‘id‘, ‘timestamp‘, ‘x‘, and ‘y‘.
start_date	Optional. A ‘Date‘ object or string (e.g., "2021-01-01"). Filters out data before this date.
end_date	Optional. A ‘Date‘ object or string (e.g., "2021-01-15"). Filters out data after this date.
report	Logical. If TRUE, opens an HTML summary table in your browser for easy copying into slides or documents.

Value

A data frame with summary statistics per ‘id‘, or an HTML table if ‘report = TRUE‘.

Examples

```
data(godwit_tracks)
clean <- ez_track(godwit_tracks)
ez_summary(clean)
```

---

ez_track *Create a Clean Tracking Object*

Description

Imports and standardizes tracking data into a tidy format with columns: ‘id‘, ‘timestamp‘, ‘x‘, and ‘y‘. Supports input as data frames, ‘sf‘, ‘Spatial*‘ objects, or file paths to CSV, Excel, Shapefiles, and GeoPackages. Optionally returns a spatial object projected to WGS84 (EPSG:4326), and supports subsampling (e.g., "1 per hour").

Usage

```
ez_track(
  data,
  format = NULL,
  tz = "UTC",
  crs = 4326,
  as_sf = TRUE,
  id = NULL,
  timestamp = NULL,
  x = NULL,
  y = NULL,
  keep_original_cols = TRUE,
```

```
  subsample = "none",
  verbose = TRUE,
  ...
)
```

Arguments

data	A tracking dataset or file path. Accepted types: ‘data.frame‘, ‘sf‘, ‘Spatial*‘, or path to CSV, XLSX, SHP, or GPKG.
format	Optional. File format to override detection. Choices: "csv", "xlsx", "shp", "gpkg".
tz	Timezone for timestamps. Default is "UTC".
crs	EPSG code or proj4string of the input CRS. Default is 4326 (WGS84).
as_sf	Logical. Return an ‘sf‘ object? Default is TRUE.
id	Optional. Column name for id.
timestamp	Optional. Column name timestamp.
x	Optional. Column name for longitude.
y	Optional. Column name for latitude.
keep_original_cols	Logical. If FALSE, drops non-standard columns and only retains ‘id‘, ‘timestamp‘, ‘x‘, and ‘y‘. Default is TRUE.
subsample	Optional. Specify how many fixes to keep per time unit. You can use any positive integer and ‘"hour"‘ or ‘"day"‘ as the unit (e.g.,‘"1 per hour"‘ or ‘"2 per day"‘).
verbose	Logical. Print messages? Default is TRUE.
...	Passed to the read function.

Value

A data.frame or ‘sf‘ object with columns ‘id‘, ‘timestamp‘, ‘x‘, ‘y‘.

---

godwit_tracks *Example Godwit Tracking Dataset*

Description

A dataset of godwit tracking data used as an example for ezTrack functions.

Usage

```
data(godwit_tracks)
```

Format

A data frame with columns:

individual.local.identifier Unique identifier for each tracked animal

timestamp Datetime of the location fix

location.long Longitude coordinate (WGS84)

location.lat Latitude coordinate (WGS84)

Source

Movebank

Examples

```
data(godwit_tracks)
godwit_tracks <- ez_track(godwit_tracks)
ez_summary(godwit_tracks)
ez_home_range(godwit_tracks)
```

`%>%` *Pipe operator*

Description

Re-exports the pipe operator `%>%` from 'magrittr', allowing you to chain commands together in a readable left-to-right style.

Arguments

`lhs`	A value or the result of a function call.
`rhs`	A function call using the value from 'lhs' as the first argument.

Value

The result of evaluating the right-hand side ('rhs') expression, where the left-hand side ('lhs') value is passed as the first argument.

Index