Package: crawlUtils (via r-universe)

September 12, 2024

Type Package

Title Enhance And Integrate the {crawl} Package For Spatial Analysis Of Telemetry Output

Version 0.1.62

Date 2024-09-12

Maintainer Devin S. Johnson <devin.johnson@noaa.gov>

Description Utility functions to augment the the {crawl} package and integrate it with the {sf} package for spatial analysis of telemetry model output. The additional function are targeted toward analysis of marine mammal telemetry, but can be used or easily modified for other situations.

License CC0

Encoding UTF-8

LazyData true

RoxygenNote 7.3.2

Roxygen list(markdown = TRUE)

Imports fuzzyjoin, dplyr, lubridate, magrittr, purrr, tidyr, sf, crawl, foreach, units, janitor, readr, ggplot2, mgcv, mclust, terra, MASS

Suggests pathroutr, ctmm

Remotes jmlondon/pathroutr

Repository https://dsjohnson.r-universe.dev

RemoteUrl https://github.com/dsjohnson/crawlUtils

RemoteRef HEAD

RemoteSha 87b78e7a13479d5e67514fb07275b25c172f6bca

Contents

---

crawlUtils-package *Functions To Increase Usability Of The* crawl *Package*

Description

This package is a collection of functions that enhance the crawl package for for analysis of animal telemetry data. The functions integrate crawl output and the sf package for ease of model fitting and track prediction, notably in marine environments.

Package:	crawlUtils
Type:	Package
Version:	0.1.61
Date:	September 11, 2024
License:	CC0
LazyLoad:	yes

Note

This software package is developed and maintained by scientists at the NOAA Fisheries Pacific Islands Fisheries Science Center and should be considered a fundamental research communication. The recommendations and conclusions presented here are those of the authors and this software should not be construed as official communication by NMFS, NOAA, or the U.S. Dept. of Commerce. In addition, reference to trade names does not imply endorsement by the National Marine Fisheries Service, NOAA. While the best efforts have been made to insure the highest quality, tools such as this are under constant development and are subject to change.

Author(s)

Devin S. Johnson Maintainer: Devin S. Johnson devin.johnson@noaa.gov

---

`bw_silver` *Silverman default bandwidth calculation*

Description

Silverman default bandwidth calculation

Usage

```
bw_silver(xy, ess = NULL)
```

Arguments

`xy`	Data coordinates
`ess`	An effective sample size. If left as `NULL`, `nrow(xy)` is used.

`cu_add_argos_cols` *Add columns for modeling FastGPS and ARGOS error structure in the same telemetry deployment*

Description

Columns are added to the telemetry data set so that multiple data types can be used simultaneously within the same animal: FastGPS, Argos least-squares, and Argos Kalman filter.

Usage

```
cu_add_argos_cols(x, units = "meter")
```

Arguments

`x`	Data frame containing location telemetry data and Argos quality information. See 'Details' for a description of the necessary data column names.
`units`	Units for movement and location error models. One of `"meter"` (`"metre"`) or `"kilometer"` (`"kilometre"`). If the data are projected, it will automatically use the units of the locations and ignore this argument.

Details

To use this function the data set must contain the following columns with exact names: (1) `"type"`, indicate the type of location, (2) `"quality"` which indicates the Argos quality of the location, (3) The Argos KF diagnostics: `"error_semi_major_axis"`, `"error_semi_minor_axis"`, and `"error_ellipse_orientation"`. If there are no Argos KF locations, the associated columns need not be present. Values of `type` are 'FastGPS', 'Argos', or 'known'. For 'FastGPS' an error radius of 100m is assumed. For 'known' an error radius of 20m is assumed. Values of `quality` must be '3','2','1','0','A',or 'B' for `type=='Argos'` locations and '4'-'10' for `type=='FastGPS'`. For other types it is ignored.

Author(s)

Devin S. Johnson

`cu_add_bouts` *Add migration detection results to original location data*

Description

Add bout data from `migr_evt` See `cu_bout_det`

Usage

```
cu_add_bouts(data, bout_tbl)
```

Arguments

`data`	Original data used by `cu_migration_det`.
`bout_tbl`	Results table produced by `cu_migration_det`.

Author(s)

Devin S. Johnson

cu_add_gaps *Label telemetry bout segments*

Description

Label segments of telemetry data such that gaps between observed locations are less than or equal to the interval specified.

Usage

```
cu_add_gaps(x, gap = 7, time_unit = "days", ...)
```

Arguments

`x` A data set of telemetry locations and times.

`gap` specified maximum time interval for which a new bout is started. Defaults to `gap=7` days.

`time_unit` Unit of time of the gap specification. Defaults to `"day"`.

`...` Ignored arguments

Author(s)

Devin S. Johnson

cu_argosDiag2Cov *Transform Argos diagnostic data to covariance matrix form*

Description

Using this function the user can transform the Argos diagnostic data for location error into a form usable as a covariance matrix to approximate the location error with a bivariate Gaussian distribution. The resulting data.frame should be attached back to the data with `cbind` to use with the `crwMLE` function.

Usage

```
cu_argosDiag2Cov(Major, Minor, Orientation)
```

Arguments

`Major` A vector containing the major axis information for each observation (na values are ok)

`Minor` A vector containing the minor axis information for each observation (na values are ok)

`Orientation` A vector containing the angle orientation of the Major axis from North (na values are ok)

Value

A data.frame with the following columns

ln.sd.x The log standard deviation of the location error in the x coordinate

ln.sd.y The log standard deviation of the location error in the x coordinate

rho The correlation of the bivariate location error ellipse

Author(s)

Devin S. Johnson

cu_bout_det *Movement bout detection*

Description

Creates a data table that indicates the times of different bouts of movement. This method uses changes in the overall dispersion rate of the animal from the 'base' time to detect changes in overall movement from small scale local movement to large scale migration.

Usage

```
cu_bout_det(
  data,
  min_disp,
  migr_disp_cut = 1,
  min_bout_len = 3,
  grid_res = "day",
  base = "first",
  max_k = 100
)
```

Arguments

data An sf data set of locations with times noted by the 'datetime' column

min_disp The minimum dispersion rate to be considered a migration interval, e.g. 10 for a 10km dispersion minimum.

migr_disp_cut The minimum per day dispersal rate to categorize a movement bout as "migratory" Defaults to 1.

min_bout_len The mimimum length of time that a migration or non-migration event will take, e.g., 7 implies a minimum of 7 time intervals for a bout.

grid_res The temporal resolution at which migrations are detected. e.g., "day" (default) implies migration start and end is detected on a daily resolution.

base The location at which dispersion is measured. Can be one of "first" (first location), "last" (final location), or some other sf::sfc point location.

max_k The maximum degrees of freedom used by mgcv::gam to model dispersion and estimate the derivative of the dispersion function.

Author(s)

Devin S. Johnson

`cu_bout_det_mbc` *Movement bout detection*

Description

Creates a data table that indicates the times of different bouts of movement. This method uses changes in the overall dispersion rate of the animal from the 'base' time to detect changes in overall movement from small scale local movement to large scale migration.

Usage

```
cu_bout_det_mbc(
  data,
  min_disp,
  migr_speed_cut = 1,
  min_bout_len = 3,
  Gmax = 3,
  grid_res = "day",
  base = "first"
)
```

Arguments

`data` An `sf` data set of locations with times noted by the 'datetime' column

`min_disp` The minimum dispersion rate to be considered a migration interval, e.g. 10 for a 10km dispersion minimum.

`migr_speed_cut` The minimum speed to categorize a movement bout as "migratory" Defaults to 1 km/h.

`min_bout_len` Minimum length for each bout, defaults to 3

`Gmax` Maximum number of movement clusters, defaults to 3.

`grid_res` The temporal resolution at which migrations are detected. e.g., "day" (default) implies migration start and end is detected on a daily resolution.

`base` The location at which dispersion is measured. Can be one of `"first"` (first location), `"last"` (final location), or some other `sf::sfc` point location.

Author(s)

Devin S. Johnson

cu_bout_summary *Create a Table of Start and End Times for each Bout in a telemetry data set*

Description

Summarizes a telemetry dataframe with a bout column into a data frame with start and end times for each bout. For use with `cu_join_interval_tbl`.

Usage

```
cu_bout_summary(x, ...)
```

Arguments

`x`	A telemetry data frame containing a `bout_id` column.
`...`	Ignored arguments

Author(s)

Devin S. Johnson

cu_crw_argos *Batch Fitting CTCRW Models for Argos (and FastGPS) Data*

Description

A basic CTCRW model is fitted to a list of data sets where each element in the list represents the data for a single individual or deployment.

Usage

```
cu_crw_argos(
  data,
  move_phase = NULL,
  bm = FALSE,
  use_prior = TRUE,
  crw_control = NULL,
  fit = TRUE,
  skip_check = FALSE,
  ...
)
```

Arguments

`data` A data set containg the telemetry data for a single individual. Will also accept a single `sf` data frame as well.

`move_phase` An optional character value indicating a factor variable column in the data that designates different movement phases.

`bm` Fit a Brownian Motion model rather than in integrated OU model. Defaults to `bm = FALSE`.

`use_prior` Logical. Should a sensible mixture normal prior be use for the log beta and log error scale parameters to impose a soft constraint for better numerical optimization. Default is `TRUE`.

`crw_control` A named list passed to `crwMLE` for optimization. Alternatives for the default values of `initialSANN`, `attempts`, `control`, `theta`, `fixPar`, `constr`, and `prior` can be specified here. See `crwMLE` for a description of these arguments. WARNING!!! No checks are made for validity of the user override. So know what you are doing.

`fit` Logical. CTCRW parameters are estimated if `fit=TRUE` (default), else the results of `displayPar`.

`skip_check` See `crwMLE` v2.3.0. Currently ignored.

`...` Additional arguments passed to the `foreach` function, e.g., for error handling in the loop.

`cu_crw_covfun` *Create covariance function for a fitted CRW model object*

Description

A function is created to evaluate the covariance function of the fitted CRW movement model

Usage

```
cu_crw_covfun(x)
```

Arguments

`x` A crwFit object created by a call to `crwMLE`

Details

The function returns a function to evaluate the covariance of the fitted Integrated Ornstein-Ulenbeck movement model. The returned function has 3 arguments: (1) `t1` and (2) `t2` both vectors of times to evaluate the covariance function of the fitted IOU model, and (3) `E` (defaults to `E=0`). Which is the "zero" time of the process. Typically `E` will be the time of the first observation.

Author(s)

Devin S. Johnson

References

Taylor, J. M., Cumberland, W. G., & Sy, J. P. (1994). A stochastic model for analysis of longitudinal AIDS data. Journal of the American Statistical Association, 89(427), 727-736.

Johnson, D. S., London, J. M., Lea, M. A., & Durban, J. W. (2008). Continuous-time correlated random walk model for animal telemetry data. Ecology, 89(5), 1208-1215.

cu_crw_covmat *Calculate correlation matrix for a set of times from a CRW covariance function*

Description

Using a correlation function created by cu_crw_covfun from a fitted CRW model a covariance (correlation) matrix is created for observations at the user provided times.

Usage

```
cu_crw_covmat(x, corr = TRUE, cf, E = 0)
```

Arguments

x	Either a crwFit object from a call to crwMLE, or a vector of times.
corr	Should the function return a correlation or covariance matrix? Defaults to corr = TRUE.
cf	Covariance function created from a crwFit object with a call to cu_crw_covfun.
E	The 'zero' time used for the covariance function. Defaults to E = 0. (See cu_crw_covfun).

Details

If x is a crwFit object, then the cf and E arguments are ignored. The resulting matrix is the covariance (correlation) matrix for the observed location times conditioned on the first location. Therefore, for n observed locations an (n-1) by (n-1) matrix will result. This is most useful for the effective sample size computation for a kernel density estimate.

Author(s)

Devin S. Johnson

cu_crw_ess *Calculate Effective Sample Size for a Set of CRW locations*

Description

Estimates the number of independent locations in a CRW data set using the mutual information method of Bartoszek (2016).

Usage

```
cu_crw_ess(fit, aug = NULL)
```

Arguments

fit	A crwFit object (See crwMLE).
aug	Either a crwPredict or crwPostIS objects from which the extra predTime location times will be used in the calculation. The crw_as_sf transformed versions of these objects will also work.

Details

Uses the "mutual information" formulation of Bartoszek (2016) to calculate the equivalent number of independent animal locations.

Author(s)

Devin S. Johnson

References

Bartoszek, K. (2016). Phylogenetic effective sample size. Journal of Theoretical Biology. 407:371-386. (See https://arxiv.org/pdf/1507.07113.pdf).

cu_crw_predict *Batch CRW Prediction for Multiple Animals*

Description

Uses a list of CRW fitted models and desired prediction times to make location (and velocity) predictions for telemetered animals.

Usage

```
cu_crw_predict(
  fit,
  predTime = NULL,
  barrier = NULL,
  vis_graph = NULL,
  as_sf = TRUE,
  ...
)
```

Arguments

`fit`	A CRW fit object
`predTime`	A character string describing the desired frequency of prediction, e.g., `predTime="1 hour"` or `predTime="15 min"`.
`barrier`	An `sf` polygon object representing areas where the animal cannot access.
`vis_graph`	A visibility graph constructed with the R package `pathroutr`, which is used to reroute paths around barriers.
`as_sf`	Logical. Return an `sf` points data frame (`TRUE`) or standard `crawl` prediction.
`...`	Additional arguments passed to the `foreach` function, e.g., for error handling in the loop.

Details

The R package `pathroutr` is necessary for use of the `barrier` rerouting. it can be installed with the command `install.packages('pathroutr', repos='https://jmlondon.r-universe.dev')`. See 'https://github.com/jmlondon/pathroutr' for a description of use and constructing the viability `vis_graph`.

Author(s)

Devin S. Johnson

`cu_crw_sample` *Batch CRW Posterior Path Simulation For Multiple Animals*

Description

Uses a list of CRW fitted models and desired simulation times to make draws from the location (and velocity) posterior distribution for telemetered animals.

Usage

```
cu_crw_sample(
  fit,
  size = 8,
  predTime = NULL,
  barrier = NULL,
  vis_graph = NULL,
  as_sf = TRUE,
  ...
)
```

Arguments

`fit` A CRW fit object

`size` The number of posterior draws. Defaults to 8 (See Details).

`predTime` A character string describing the desired frequency of prediction, e.g., `predTime="1 hour"` or `predTime="15 min"`.

`barrier` An `sf` polygon object representing areas where the animal cannot access.

`vis_graph` A visibility graph constructed with the R package `pathroutr`, which is used to reroute paths around barriers.

`as_sf` Logical. Return an `sf` points data frame list (`TRUE`) or standard `crawl` prediction list

`...` Additional arguments passed to the `foreach` function, e.g., for error handling in the loop.

Details

The R package `pathroutr` is necessary for use of the `barrier` rerouting. it can be installed with the command `install.packages('pathroutr', repos='https://jmlondon.r-universe.dev')`. See 'https://github.com/jmlondon/pathroutr' for a description of use and constructing the viability `vis_graph`.

Author(s)

Devin S. Johnson

`cu_disp_rate` *Movement bout detection*

Description

Creates a data table that indicates the times of different bouts of movement. This method uses changes in the overall dispersion rate of the animal from the 'base' time to detect changes in overall movement from small scale local movement to large scale migration.

Usage

```
cu_disp_rate(bout_tbl, data, grid_res = "day", base = "first")
```

Arguments

`bout_tbl`	A bout table for an individual animal created by `cu_bout_det`
`data`	An `sf` data set of locations with times noted by the 'datetime' column
`grid_res`	The temporal resolution at which migrations are detected. e.g., "day" (default) implies migration start and end is detected on a daily resolution.
`base`	The location at which dispersion is measured. Can be one of `"first"` (first location), `"last"` (final location), or some other `sf::sfc` point location.

Author(s)

Devin S. Johnson

`cu_empirical_ud` *Calculate an empirical utilization distribution*

Description

Calculate an empirical utilization distribution

Usage

```
cu_empirical_ud(pts, grid, average = TRUE)
```

Arguments

`pts`	an sf point object
`grid`	a `SpatRaster` on which to evaluate the UD.
`average`	should an average UD be created from 1 or more pts layers

Value

grid with an additional column, *npts* or *mean_pts*

cu_extract_obst *Extract predicted locatiins closest in time to observed locations.*

Description

This function is intended for use after a predicted track or posterior simulation is routed around barriers using the `{pathroutr}` package.

Usage

```
cu_extract_obst(x, obs)
```

Arguments

`x`	A `crwPredict` or `crwIS` object. Time column must be named `datetime`.
`obs`	Observed locations. Time column must be named `datetime`.

Author(s)

Devin S. Johnson

cu_hud *Highest utilization density*

Description

The lowest 1-alpha percent of a utilization distribution is removed to give a highest prob\ reduce spatial extent of UDs for a animal with a small spatial scale of use relative to the study area

Usage

```
cu_hud(ud, prob = 0.9)
```

Arguments

`ud`	A `SpatRaster` object output from `cu_sf_kde`.
`prob`	The percent cutoff for the highest utilization probability cells. Defaults to `prob = 0.9`.

Author(s)

Devin S. Johnson

cu_join_interval_tbl *Join crawl prediction or simulation output with a table based on a time interval*

Description

Takes a data set with a POSIX time column named 'datetime' and another data set with `start` and `end` columns representing time intervals and merges the two depending whether or not the 'datetime' column is within the interval of the second.

Usage

```
cu_join_interval_tbl(x, int_tbl)
```

Arguments

`x`	A data frame with a column labeled `datetime`
`int_tbl`	A data frame with 'start' and 'end' columns that form non-overlapping intervals as well as at least one other column with interval level data.

Author(s)

Devin S. Johnson

cu_location_rate *Location rate statistics*

Description

Calculate location rate statistics such as mean location rate per day or maximum number of locations per day for a telemetry

Usage

```
cu_location_rate(x, time_name, time_unit = "day", stat = mean, ...)
```

Arguments

`x`	data set containing time of locations.
`time_name`	Character name of the POSIX time column when locations are observed.
`time_unit`	Time unit of the location summary. Defaults to `"day"`.
`stat`	Function used to summarize location times. Defaults to `mean`.
`...`	Additional arguments passed to `stat` function.

Author(s)

Devin S. Johnson

cu_plot_disp *Plot migration detection results*

Description

PLot the animal dispersion from the base location over time. Points are colored to reflect estimated migration and non-migration phases. See `cu_bout_det`

Usage

```
cu_plot_disp(data, bout_tbl)
```

Arguments

`data`	Original data used by `cu_bout_det`.
`bout_tbl`	Results table produced by `cu_bout_det`.

Author(s)

Devin S. Johnson

cu_read_wc_dirs *Read individual telemetry data from Wildlife Computers portal directories*

Description

Read and combine data downloaded from Wildlife Computers portal into individual directories.

Usage

```
cu_read_wc_dirs(x, remove_duplicates = TRUE)
```

Arguments

`x`	Directory containing the individual telemetry data directories.
`remove_duplicates`	Locgical. Should observations with duplicated times be removed? The observation with the highest quality will be retained.

Author(s)

Devin S. Johnson, Josh M. London

`cu_sf_kde` *Spatial kernel density estimate*

Description

A weighted or unweighted Gaussian Kernel Density estimate for point spatial data

Usage

```
cu_sf_kde(x, w = NULL, bw = NULL, ref, ess = NULL, mask = FALSE)
```

Arguments

`x`	sf POINT object
`w`	Optional values, associated with `x` coordinates, to be used as weights
`bw`	Standard deviation scale bandwidth of Gaussian Kernel, must be units of `x` projection.
`ref`	A terra SpatRaster
`ess`	A effective sample size to use instead of `nrow(x)` for determining the default bandwidth.
`mask`	(TRUE/FALSE) mask resulting raster if ref is provided as a SpatRaster

Details

Please note that ks methods for estimation has been reverted to the Gussian method proposed in Venables & Ripley (2002). There was not enought evendence that the Chacon & Duong (2018) multivariate method(s) for bandwidth selection and kernal estimation were suitable for spatial random fields.

Value

a terra SpatRaster class object containing kernel density estimate

Author(s)

Jeffrey S. Evans jeffrey_evans@tnc.org and Devin S. Johnson devin.johnson@noaa.gov

References

Duong, T. & Hazelton, M.L. (2005) Cross-validation bandwidth matrices for multivariate kernel density estimation. Scandinavian Journal of Statistics, 32, 485-506.

Wand, M.P. & Jones, M.C. (1994) Multivariate plug-in bandwidth selection. Computational Statistics, 9, 97-116.

Venables, W. N. and Ripley, B. D. (2002) Modern Applied Statistics with S. Fourth edition. Springer.

`cu_ud_contour` *Approximate contours for a UD*

Description

Approximate contours for a UD

Usage

```
cu_ud_contour(ud, prob = seq(0.9, 0.1, -0.1), trim_pixel = TRUE)
```

Arguments

`ud`	A `SpatRaster` created by `\link[crawlUtils]{cu_sf_kde}`
`prob`	a vector of probabilities for the contours
`trim_pixel`	Logical. Should contour areas less than a pixel size be trimmed? defaults to `TRUE`.

Author(s)

Devin S. Johnson

`hex_size` *Calculate cellsize value for hexagon grid*

Description

Calculates the appropriate `cellsize` argument for making a hexagon grid with `st_make_grid`.

Usage

```
hex_size(area = NULL, radius = NULL, sep = NULL)
```

Arguments

`area`	A value (m^2) for the resulting area of a full hexagon cell
`radius`	The value for the distance (m) from the centroids to the edge of full hexagon cells.
`sep`	The distance (m) between centoids of the hexagon grid.

Author(s)

Devin S. Johnson

References

See `https://github.com/r-spatial/sf/issues/1505`

st_bbox_list — *sf::st_bbox for a list of sf or sfc objects.*

Description

sf::st_bbox for a list of sf or sfc objects.

Usage

```
st_bbox_list(x, union = TRUE, as_sfc = FALSE)
```

Arguments

x	A list of sf or sfc objects.
union	Logical. Should the bounding box of the union be returned instead of a list of bounding boxes.
as_sfc	Logical. Should the bounding box (boxes) be returned as sfc objects.

Author(s)

Devin S. Johnson

st_expand — *Expand Spatial Bounding Box*

Description

Expand an sf bounding box by an expansion factor

Usage

```
st_expand(bbox, ef)
```

Arguments

bbox	An sf bounding box. See sf::st_bbox.
ef	Expansion factor, must be positive and length 1, 2 or 4. If length(ef)==1 then the expansion factor is applied equally in all 4 directions. If length(ef)==2 then the first refers to the x coordinate and the second is associated with y coordinate expansion. If length(ef)==4 then then the first 2 refer to expansion to the west and east of the x coordinate, and the last to expand the y coordinate in the south and north direction respectively.

Author(s)

Josh M. London

st_not_within *Predicate function for st_filter*

Description

Predicate function to use with `st_filter` such that such that elements of one spatial object are selected if they are not contained at all in the other. See `sf::st_within`

Usage

```
st_not_within(x, y, sparse = TRUE, prepared = TRUE, ...)
```

Arguments

`x`	object of class sf, sfc or sfg
`y`	object of class sf, sfc or sfg; if missing, x is used
`sparse`	ogical; should a sparse index list be returned (TRUE) or a dense logical matrix? See sf::st_within.
`prepared`	ogical; prepare geometry for x, before looping over y? See sf::st_within.
`...`	passed on to s2_options

st_to_360 *Convert longitude/latitude coordinates from -180/180 to 0/360*

Description

Converts sf data with EPSG = 4326 from -180/180 specification to 0/360 for plotting with the mapview package etc.

Usage

```
st_to_360(x)
```

Arguments

`x`	An sf data frame with EPSG=4326.

Author(s)

Josh London

st_union_list *sf::st_union for a list of sf or sfc objects.*

Description

sf::st_union for a list of sf or sfc objects.

Usage

```
st_union_list(x)
```

Arguments

x	A list of sf or sfc objects.

Author(s)

Devin S. Johnson

Index