Package 'activity'

September 26, 2016

Type Package

Title Animal Activity Statistics

Version 1.1

Date 2016-09-23

Author Marcus Rowcliffe <marcus.rowcliffe@ioz.ac.uk>

Maintainer Marcus Rowcliffe <marcus.rowcliffe@ioz.ac.uk>

Description Provides functions to fit kernel density functions to animal activity time data; plot activity distributions; quantify overall levels of activity; statistically compare activity metrics through bootstrapping; and evaluate variation in linear variables with time (or other circular variables).

License GPL-3

Depends methods

Imports circular, overlap, pbapply,

LazyData TRUE

RoxygenNote 5.0.1

NeedsCompilation no

Repository CRAN

Date/Publication 2016-09-26 08:03:36

R topics documented:

`activity` *Animal activity statistics*

Description

Provides functions to estimate and compare activity parameters from sensor data.

Details

Sensors that record active animals (eg camera traps) build up a record of the distribution of activity over the course of the day. Records are more frequent when animals are more active, and less frequent or absent when animals are inactive. The area under the distribution of records thus contains information on the overall level of activity in a sampled population. This package provides tools for plotting activity distributions, quantifying the overall level of activity with error, and statistically comparing distributions through bootstrapping.

The core function is `fitact`, which creates an `actmod` object containing the circular kernel PDF, and the activity level estimate derived from this. The generic plot function for `actmod` objects plots the distribution. Functions starting with `compare` make statistical comparisons between distributions or activity estimates. Note that all time or other circular data should be in radians (in the range 0 to 2*pi).

References

Rowcliffe, M., Kays, R., Kranstauber, B., Carbone, C., Jansen, P.A. (2014) Quantifying animal activity level using camera trap data. Methods in Ecology and Evolution.

See Also

`overlap`

actmod-class *Activity model class.*

Description

An S4 class describing activity models fitted to time of observation data.

Slots

`data` Object of class `"numeric"`, the input data.

`wt` Object of class `"numeric"`, weights applied to the data.

`bw` Object of class `"numeric"`, kernel bandwidth.

`adj` Object of class `"numeric"`, kernel bandwidth adjustment multiplier.

`pdf` Object of class `"matrix"` describing fitted probability density function: Column 1: Sequence of radian times at which PDF evaluated (specifically seq(0,2*pi,pi/256)). Column 2: Corresponding circular kernel PDF values. Additionally if errors boostrapped: Column 3: PDF standard error. Column 4: PDF lower confidence interval. Column 5: PDF upper confidence interval.

`act` Object of class `"numeric"` giving activity level estimate and, if errors boostrapped, standard error and 95 percent confidence limits.

BCIspeed *Animal speed data*

Description

Barro Colorado Island 2008 data: speeds of animal passages past camera traps (`speed`), together with species (`species`) and time of day (`time`) for each record.

Format

A dataframe with 2204 observations and 3 variables.

Source

http://dx.doi.org/10.6084/m9.figshare.1160536

BCItime *Animal record time of day data*

Description

Barro Colorado Island 2008 data: times of day at which animal records occured (`time`), together with species (`species`).

Format

A dataframe with 17820 observations and 2 variables.

Source

http://dx.doi.org/10.6084/m9.figshare.1160536

bwcalc *Calculate circular kernel bandwidth.*

Description

Uses an optimisation procedure to calculate the circular kernel bandwidth giving the best fit to the data.

Usage

```
bwcalc(dat, K = 3)
```

Arguments

`dat`	Numeric data vector of radian times.
`K`	Integer number of values of kappa over which to maximise (see references for details).

Details

Mainly for internal use.

Value

Single numeric bandwidth value.

References

Ridout, M.S. & Linkie, M. (2009) Estimating overlap of daily activity patterns from camera trap data. Journal of Agricultural Biological and Environmental Statistics, 14, 322-337.

compareAct *Compare activity level estimates*

Description

Wald test for the statistical difference between two or more activitiy level estimates.

Usage

```
compareAct(fits)
```

Arguments

fits	A list of fitted actmod objects

Details

Uses a Wald test to ask whether the difference between estimates a1 and a2 is significantly different from 0: statistic W = (a1-a2)^2 / (SE1^2+SE2^2) tested on chi-sq distribution with 1 degree of freedom.

Value

A matrix with 4 columns: 1. differences between estimates; 2. SEs of the differences; 3. Wald statistics; 4. p-values (H0 is no difference between estimates). Matrix rows give all possible pairwise comparisons, numbered in the order in which they entered in the list fits.

Examples

```
#Test whether paca have a sigificantly different activity level from rat.
#Bootstrap reps limited to speed up example.
data(BCItime)
tPaca <- 2*pi*BCItime$time[BCItime$species=="paca"]
tRat <- 2*pi*BCItime$time[BCItime$species=="rat"]
(fPaca <- fitact(tPaca, reps=10))
(fRat <- fitact(tRat, reps=10))
compareAct(list(fPaca,fRat))
```

compareCkern *Compare circular distributions.*

Description

Randomisation test for the probability that two sets of circular observations come from the same distribution.

Usage

```
compareCkern(y1, y2, reps = 1000, index = c("Dhat4", "Dhat5", "Dhat1"))
```

Arguments

y1, y2	Numeric vectors of radian data.
reps	Number of bootstrap iterations.
index	Which of the three indices returned by overlap::overlapEst to use.

Details

Calculates overlap index (see references) for the observed data samples, then generates a null distribution of overlap indices using data sampled randomly with replacement from the combined data. This randomised distribution is then used to estimate the probability that the observed overlap arose by chance.

Value

A named 2-element vector: Overlap = observed overlap index; p = probability observed index arose by chance.

References

Ridout, M.S. & Linkie, M. (2009) Estimating overlap of daily activity patterns from camera trap data. Journal of Agricultural Biological and Environmental Statistics, 14, 322-337.

See Also

overlapEst

Examples

```
#Example with bootstrap reps limited to speed up
data(BCItime)
tPaca <- 2*pi*BCItime$time[BCItime$species=="paca"]
tRat <- 2*pi*BCItime$time[BCItime$species=="rat"]
compareCkern(tPaca,tRat,reps=10)
```

compareTimes *Compare activity across between times of day*

Description

Uses a Wald test to statistically compare activity levels at given radian times of day for a fitted activity distribution.

Usage

```
compareTimes(fit, times)
```

Arguments

fit	Fitted actmod object with errors boostrapped (fit using fitact with sample argument != "none").
times	Numeric vector of radian times of day at which to compare activity levels. All pairwise comparisons are made.

Details

Bootrapping the activity model yields standard error estimates for the PDF. This function uses these SEs to compute a Wald statistic for the difference between PDF values (by inference activity levels) at given times of day: statistic W = (a1-a2)^2 / (SE1^2+SE2^2) tested on chi-sq distribution with 1 degree of freedom.

Value

A matrix with 4 columns: 1. differences between PDF values; 2. SEs of the differences; 3. Wald statistics; 4. p-values (H0 is no difference between estimates). Matrix rows give all possible pairwise comparisons, numbered in the order in which they appear in vector times.

Examples

```
data(BCItime)
tPaca <- 2*pi*BCItime$time[BCItime$species=="paca"]
fPaca <- fitact(tPaca, reps=10)
plot(fPaca, hrs=FALSE, frq=FALSE)
compareTimes(fPaca, c(5.5,6,0.5,1))
```

dvmkern *Circular kernel probability density function.*

Description

Optionally weighted Von Mises kernel probability densities.

Usage

```
dvmkern(x, dat, wt = NULL, bw = NULL, adj = 1)
```

Arguments

x	Numeric vector of radian times at which to evaluate the PDF.
dat	Numeric vector of radian time data to which the PDF is fitted.
wt	A numeric vector of weights for each dat value.
bw	Numeric value for kernel bandwidth.
adj	Numeric kernel bandwidth multiplier.

Details

If bw not provided it is calculated internally using bw.calc. The adj argument is used to adjust bw to facilitate exploration of fit flexibility.

Value

Numeric vector of probability densities evaluated at x.

See Also

density.circular, bwcalc

Examples

```
#Example with made up input
tt <- runif(100,0,2*pi)
xx <- seq(0,2*pi, pi/256)
pdf <- dvmkern(xx, tt)
plot(xx, pdf, type="l")
```

fitact *Fit activity model to time-of-day data*

Description

Fits a circular kernel density to radian time-of-day data and estimates activity level from this distribution. Optionally bootstraps the distribution, in which case SEs and confidence limits are also stored for activity level and PDF.

Usage

```
fitact(dat, wt = NULL, reps = 1000, bw = NULL, adj = 1,
  sample = c("data", "model", "none"), show = TRUE)
```

Arguments

dat	A numeric vector of radian time-of-day data.
wt	A numeric vector of weights for each dat value.
reps	Number of boostrap iterations to perform. Ignored if sample=="none".
bw	Numeric value for kernel bandwidth. If NULL, calculated internally.
adj	Numeric bandwidth adjustment multiplier.
sample	Character string defining sampling method for bootstrapping errors (see details).
show	Logical whether or not to show a progress bar while bootstrapping.

Details

The bandwidth adjustment multiplier adj is provided to allow exploration of the effect of adjusting the internally calculated bandwidth on accuracy of activity level estimates. The alternative bootstrapping methods defined by sample are: data: sample from the data; model: sample from the fitted probability density distribution; none: no bootstrapping. Confidence interval coverage seems to be better at large sample size (greater than 100-200) using "model", but better at small sample size when using "data". The reason for this needs further investigation.

Value

An object of type actmod

Examples

```
#Fit without confidence limits
data(BCItime)
tdat <- 2*pi*BCItime$time[BCItime$species=="ocelot"]
mod1 <- fitact(tdat, sample="none")
plot(mod1)

#Fit with confidence limits (limited reps to speed up)
```

```
mod2 <- fitact(tdat, reps=10)
plot(mod2)

#Fit weighted function to correct for detection radius 1.21 times higher
#by day than by night, assuming day between pi/2 (6am) and pi*2/3 (6pm)
weight <- 1/ifelse(tdat>pi/2 & tdat<pi*3/2, 1.2, 1)
mod3 <- fitact(tdat, wt=weight, sample="none")
plot(mod3)
#Overplot unweighted version for comparison
plot(mod1, add=TRUE, lcol=3)
```

`fitlincirc` *Linear-circular regression*

Description

Fits a Von Mises kernel distribution describing a linear variable as a function of a circular predictor, and boostraps the null distribution in order to evaluate significance of radial variation in the linear variable.

Usage

```
fitlincirc(circdat, lindat, pCI = 0.95, reps = 1000, res = 512)
```

Arguments

`circdat`	Numeric vector of radian data matched with `lindat`.
`lindat`	Numeric vector of linear data matched with `circdat`.
`pCI`	Single numeric value between 0 and 1 defining proportional confidence interval to return.
`reps`	Integer number of bootstrap repetitions to perform.
`res`	Resolution of fitted distribution and null confidence interval - specifically a single integer number of points on the circular scale at which to record distributions.

Details

Deviation of `lindat` from the null expecation is assessed either visually by the degree to which the fitted distribution departs from the null confidence interval (use generic plot function), or quantitatively by column `p` of slot `fit` in the resulting `lincircmod-class` object.

Value

An object of type `lincircmod-class`

References

Xu, H., Nichols, K. & Schoenberg, F.P. (2011) Directional kernel regression for wind and fire data. Forest Science, 57, 343-352.

Examples

```
#Example with reps limited to increase speed
data(BCIspeed)
i <- BCIspeed$species=="ocelot"
sp <- log(BCIspeed$speed[i])
tm <- BCIspeed$time[i]*2*pi
mod <- fitlincirc(tm, sp, reps=50)
plot(mod, CircScale=24, xaxp=c(0,24,4),
     xlab="Time", ylab="log(speed m/s)")
legend(8,-3, c("Fitted speed", "Null CI"), col=1:2, lty=1:2)
```

`lincircKern` *Linear-circular kernel fit*

Description

Fits a Von Mises kernel distribution describing a linear variable as a function of a circular predictor.

Usage

```
lincircKern(x, circdat, lindat)
```

Arguments

`x`	Numeric vector of radian values at which to evaluate the distribution.
`circdat`	Numeric vector of radian data matched with `lindat`.
`lindat`	Numeric vector of linear data matched with `circdat`.

Value

A numeric vector of fitted `lindat` values matched with `x`.

References

Xu, H., Nichols, K. & Schoenberg, F.P. (2011) Directional kernel regression for wind and fire data. Forest Science, 57, 343-352.

Examples

```
data(BCIspeed)
i <- BCIspeed$species=="ocelot"
sp <- log(BCIspeed$speed[i])
tm <- BCIspeed$time[i]*2*pi
circseq <- seq(0,2*pi,pi/256)
trend <- lincircKern(circseq, tm, sp)
plot(circseq, trend, type="l")
```

lincircmod-class *An S4 class describing linear-circular relationships.*

Description

An S4 class describing linear-circular relationships.

Slots

`data` Object of class `"data.frame"`, the input data, with columns `lindat` (linear data) and `circdat` (circular data).

`fit` Object of class `"data.frame"`, summary of the model fit, with columns: `x`: A regular ascending sequence from 0 to 2*pi at which other columns evaluated; `fit`: The linear fitted values; `p`: The two tailed probability of observing the fitted values under a random (null) circular distribution; `nullLCL`: The lower confidence limit of the null distribution; `nullUCL`: The upper confidence limit of the null distribution.

plot,actmod-method *Plot activity distribution*

Description

Plot an activity probability distribution from a fitted `actmod` object.

Usage

```
## S4 method for signature 'actmod'
plot(x, xunit = c("hours", "radians"),
  yunit = c("frequency", "density"), dat = c("histogram", "rug", "none"),
  add = FALSE, dcol = 1, lcol = 2, ...)
```

Arguments

`x`	Object of class `actmod`.
`xunit`	Character string defining x-axis unit.
`yunit`	Character string defining y-axis unit.
`dat`	Character string defining whether to plot the data distribution and if so which style to use.
`add`	Logical defining whether to create a new plot (default) or add the probability density to an existing plot (in which case no data are plotted).
`dcol`	Numeric or character defining colour of data lines.
`lcol`	Numeric or character defining colour of PDF lines.
`...`	Additional arguments passed to internal plot call affecting only axes.

plot,lincircmod-method *Plot linear-circular relationship*

Description

Plot linear against circular data along with the fitted and null confidence limit distributions from a fitted `lincircmod` object.

Usage

```
## S4 method for signature 'lincircmod'
plot(x, CircScale = 2 * pi, tlim = c(0, 1),
  fcol = "black", flty = 1, ncol = "red", nlty = 2, ...)
```

Arguments

`x`	Object of class `lincircmod`.
`CircScale`	Single numeric value defining the plotting maximum of the circular scale.
`tlim`	Numeric vector with two elements >=0 and <=1 defining the lower and upper limits at which to plot distributions; default plots the full range.
`fcol, flty, ncol, nlty`	Define line colour (`col`) and type (`lty`) for fitted (`f`) and null (`n`) distributions; input types as for `col` and `lty`, see `par`.
`...`	Additional arguments passed to the inital plot construction, affecting axes and data plot symbols.

rvmkern *Random circular kernel numbers.*

Description

Random numbers drawn from a fitted Von Mises kernel distribution.

Usage

```
rvmkern(n, fit)
```

Arguments

`n`	Integer number of numbers to return.
`fit`	An emprical distribution contained in either a matrix or a fitted `actmod` object (see Details).

Details

If `fit` is a matrix, it should have two columns: [,1] precisely seq(0,2*pi,pi/256) sequence of radian values at which pdf evaluated; [,2] corresponding pdf values.

Value

A numeric vector of radian numbers.

Examples

```
#Matrix input
data(BCItime)
tm <- 2*pi*BCItime$time[BCItime$species=="paca"]
xx <- seq(0,2*pi, pi/256)
rn <- rvmkern(100, cbind(xx, dvmkern(xx, tm)))

#actmod input
fit <- fitact(tm, sample="n")
rvmkern(100,fit)
```

---

show,actmod-method *Show activity level estimate*

Description

Prints the `act` slot (activity level estimate) from an `actmod` object.

Usage

```
## S4 method for signature 'actmod'
show(object)
```

Arguments

`object`	Object of class `actmod`.

---

yfromx *Impute empirical circular distribution.*

Description

Imputes values at given points on an empirical circular distribution.

Usage

```
yfromx(xvals, x, y)
```

Arguments

xvals	Numeric circular values at which to evaluate y
x	Evenly spaced ascending numeric sequence of circular values
y	Empirical numeric output distribution matched with x

Details

Note that x is assumed circular, so first and last y values should be equal. Evaluation points xvals should also be within the range of x.

Value

A numeric vector of y values evaluated at xvals

Examples

```
#Abstract example
x <- seq(0,2*pi,length.out=11)
y <- c(0,1,2,3,4,5,4,3,2,1,0)
yfromx(0:6,x,y)

#BCI data example
#Weighting ocelot activity pattern to correct for variation in speed
data(BCIspeed)
data(BCItime)
#Fit linear-circular model to log(speed)
i <- BCIspeed$species=="ocelot"
lcfit <- fitlincirc(BCIspeed$time[i]*2*pi, log(BCIspeed$speed[i]), reps=50)
#Fit weighted activity model using yfromx to create weights
j <- BCItime$species=="ocelot"
tdat <- BCItime$time[j]*2*pi
w <- 1/yfromx(tdat, lcfit@fit$x, exp(lcfit@fit$fit))
mod <- fitact(tdat, wt=w, sample="none")
plot(mod)
#Oveplot unweighted model for comparison
mod2 <- fitact(tdat, sample="none")
plot(mod2, lcol=3, add=TRUE)
```

Index