

Package: cesr (via r-universe)

September 18, 2024

Version 1.0.5

Title Trend analysis of Constant Effort Site ringing data

Date 2024-01-19

Author Rob Robinson <rob.robinson@bto.org>

Maintainer Rob Robinson <rob.robinson@bto.org>

Description Functions for reading EuroCES data in Euring format to produce annual indices of abundance, productivity and survival. Designed to primarily analyse data from schemes with 10-12 visits per year following methods used by the BTO. Use `read.ces()` to read in data, `extract.data()` and `index()` to do abundance/productivity trends and `extract.ch()` and `mark.ces()` to estimate survival probabilities.

License GPL-2

LazyLoad yes

ByteCompile yes

ZipData no

Depends data.table, R (>= 3.5.0)

Imports reshape, RMark, maps, utils, Rcpp, mgcv, lme4

Suggests knitr, rmarkdown

LinkingTo Rcpp

VignetteBuilder knitr

Repository <https://btorobrob.r-universe.dev>

RemoteUrl <https://github.com/btorobrob/cesr>

RemoteRef HEAD

RemoteSha 0e4c0909d7556f338353e5d4754f00363497b524

Contents

cesr-package	2
ces.table	4

cesnames	5
check_rings	6
extract.ch	6
extract.counts	8
extract.coverage	9
extract.data	10
extract.sites	11
get.euring	12
index	12
load.boxplot	14
map.ces	15
mark.ces	16
plot.ces	17
plot_trend	19
print.ces	20
readces	21
save.boxplot	22
select.data	24
setceslang	25
siteplot	26
sitereport	27
summary.ces	28
transient_plot	29
ukdata	30
writeces	31
Index	33

cesr-package

Trend analysis of Constant Effort Site ringing data

Description

Facilities for reading EuroCES data in Euring format to produce annual indices of productivity, abundance and survival. Primarily designed to analyse data from schemes with 9-12 visits per year following Peach et al. (1996, 1998) and Robinson et al. (2007).

Details

Package: cesr
Type: Package
Version: 1.0.2
Date: 2023-06-07
License: GPL-2
LazyLoad: yes

Constant Effort Site (CES) ringing data provide a valuable way to monitor changes in abundance, productivity and survival through standardised capture and recapture of individuals. `cesr` provides some facilities to produce annual trends in abundance and productivity from these data. In order to do this `cesr` requires CES data organised in a plain text file according to the agreed Euring format (details of which can be found on the Euring website (http://www.euring.org/internal/ces_research_group/euro_ces_meeting_)).

The key steps are: (i) to read in the data (with `readces`);

(ii) generate a table of which plots are covered in which year (with `extract.coverage`);

(iii) extract data for the species to be analysed (with `extract.data`), and

(iv) generate the annual abundance/productivity indices (with `index`).

Adult survival probabilities can be estimated by first using `extract.ch` then `mark.ces`. Results can be plotted using `plot_trend`. More details on how to do each of these can be found in the introductory manual, type

```
vignette('cesr-intro')
```

to view it.

Author(s)

Maintainer: Rob Robinson <rob.robinson@bto.org>

References

- "Peach, W.J., Buckland, S.T. & Baillie, S.R. (1996) The use of constant effort mist-netting to measure between-year changes in the abundance and productivity of common passerines. *Bird Study* 43:142-156."
- "Peach, W.J., Baillie, S.R. & Balmer, D.E. (1998) Long-term changes in the abundance of passerines in Britain and Ireland as measured by constant effort mist-netting. *Bird Study* 45:257-275."
- "Robinson, R.A., Freeman, S.N., Balmer, D.E. & Grantham, M.J. (2007) Cetti's Warbler: anatomy of an expanding population. *Bird Study* 54:230-235."
- "Robinson, R.A., Julliard, R & Saracco, J.F. (2009) Constant effort: studying avian population processes using standardized ringing. *Ringing & Migration* 24:199-204."

Examples

```
## Not run:
# A sample session might look as follows:
setwd('d:/data/ces') # set the working directory which contains the data file
setceslang('English') # which language for species names?

alldat <- readces('netherlands.csv') # read in the data
plot.nl <- extract.coverage(alldat) # determine which sites covered in which years
sonth.nl <- extract.data(alldat, species=12000, plots=plot.nl) # extract data for Song Thrush
st.fit0 <- index(sonth.nl, year=2008) # fit an annual index
st.fit1 <- index(sonth.nl, trend=30) # fit a simple linear trend

writeces(st.fit0, 'sonthdata.txt') # write annual indices to a file

## End(Not run)
```

ces.table

*Create a table of results for a list of species***Description**

Runs a specified set of models (see details) for a list of species and present the results in a table.

Usage

```
ces.table(cesobj, species=NA, columns=c("A-1", "P-1", "S-1"), base=100, plots=NULL, min.n=100, min.ch=
```

Arguments

cesobj	a CES dataframe typically generated by readces .
species	a numeric vector of Euring species codes, these can be checked using get.euring , if missing use min.n=.
columns	a character vector specifying which models should be applied to each species, see details.
base	which number should indicate no change: 0, 1, or 100?
plots	an object containing plot coverage data created by extract.coverage . If missing one will be created containing all sites and years.
min.n	if species is not specified, include all species with more than this number of captures.
min.ch	for survival models, how many individuals need to be captured for a site to be included, see extract.ch . Can be either an integer, or a vector the same length as species.
ndigits	numeric, how many decimal places to round the results to, see note.
year	the reference year for the comparisons if not the last.
change	logical, for comparison models return the change or the relative value (e.g. +0.2 or 1.2); for trend models TRUE gives the overall change, FALSE the (annual) slope.
visit.corr	logical, include an offset for missed visits in the models, see index .
save.results	logical, save the full model results, or only the annual values. Setting this to TRUE can result in a quite a large object, so it may be helpful to skip these.

Details

Models are specified using three characters: (i) the parameter ('A'dult and 'J'juvenile numbers, 'P'roductivity and 'S'urvival, (ii) the model type, currently either '-' to compare to previous years or '/' for a trend, and (iii) the number of (previous) years over which the comparison or trend should be calculated. Thus, for example. "A-5" would compare the current year's estimate of adult captures to the previous 5 years, see [index](#) for more details on this. The default is to compare with the previous year ("-1").

For change ('-') models the estimates will be either be the index relative to the base value (so -0.15, 0.85, or 85.0 for base = 0, 1, 100 respectively) or the relevant change (if change=TRUE then -0.15, -0.15, 15.0 in those instances). For trend ('/') models the estimate will be a slope (mean annual change), on either the log (abundance) or logit (survival/productivity) scale (and might be expressed as either 0.07 or 7.00, for example, depending on whether or not base=100).

A simple table will be printed with the change/trend estimate (along with confidence limits and an indication of significance) for each model (columns) and species (rows). These numbers are returned as a dataframe, see 'value' below.

Value

a list with three components: table - a dataframe that mimics the printed table, so in 'wide' format; estimates - a dataframe in 'long' format, one row per entry; results - a named list of the model objects, in the form "T_xxxxx_mmm", where xxxxx is the Euring code number, mmm the model specification.

Note

ndigits is strict, the returned 'estimates' dataframe will be rounded to this level as well.

Author(s)

Rob Robinson

See Also

[index](#)

Examples

```
## Not run:
# compare adult numbers in the current yaer to the previous year,
# survival with the previous five years and a 10-year trend in productivity
results <- ces.table(ukdata, columns=c('A-0', 'S-5', 'P/10'))

## End(Not run)
```

cesnames

Species names translation

Description

A dataset containing species names in different languages for use by [setceslang](#)

Usage

```
data(ukdata)
```

Format

A data frame with 666 species names in English ad 11 other languages.

check_rings	<i>Check ringing details</i>
-------------	------------------------------

Description

Currently just checks for rings used on multiple species

Usage

```
check_rings(cesdata)
```

Arguments

cesdata	A CES data object
---------	-------------------

Details

Checks where multiple species are associated with a single number number. Scheme is included as part of the ring number.

Value

A dataframe containing a column of ring numbers and the species that they were ringed as; NULL if none are found. Only those rings which have apparently been put on more than one species are included.

Author(s)

Rob Robinson

extract.ch	<i>Get adult capture history data for an individual species</i>
------------	---

Description

Creates a list with individual capture histories for birds from a CES object created by readces for a survival analysis using mark.ces

Usage

```
extract.ch(cesdata, species=0, plots=NULL, late=FALSE, group=NA, exclude=NULL, min.n=1, min.yrs=3)
```

Arguments

cesdata	a CES object created by readces
species	a numeric Euring species code (including final zero which would indicate race)
plots	an object containing plot coverage data created by extract.coverage.
late	should late visits be considered in site selection? In UK (at least) most adults are caught in first 6 visits, so defaults to FALSE (ie a site is included if, say, 4 of the first 6 visits are made). Set to TRUE to require late visits to be considered also, which will results in more site-year combinations being considered as 'missing'
group	a character string indicating a grouping variable for survival, such as 'sex'. Group levels should be alphabetic only (i.e. no digits (0-9) or symbols please; these confuse MARK).
exclude	one or more levels of 'group' to exclude, for example, '-' for unsexed birds
min.n	minimum total number of captures for a site to be included, may be useful with large datasets to aid convergence or memory allocation
min.yrs	minimum number of years for which a site has operated to be included

Details

Creates a '10' style capture history for each ringed bird (if a bird moves site it is treated as a new individual). Missing visits are preserved and handled by MARK. The additional capture within the first season is entered by shifting the first capture back a 'year' and entering a 1 subsequently (ie in the first actual year of capture) if the bird is recaptured within that year, 0 otherwise. Birds initially caught as juveniles are included if they are subsequently caught as adults (with the juvenile captures removed).

Value

A list of six items

chdata	A data frame with columns: ring, ch and site
begin.time	A four digit integer indicating the first year from which observations come
years	An integer indicating the number of years from which observations come
group	a list consisting of name - character string indicating a grouping variable for survival or NA, and levels - a vector of the group levels
spp	Euring species code
spp.name	Name of the species in whichever language has been set by setceslang

Note

The matching of site names assumes that the order in which they appear in x is the original one, avoid sorting this dataframe. If histories are extracted by group, the [mark.ces](#) will always estimate survival by group, to avoid this you will need to re-extract the data without a grouping variable.

Author(s)

Rob Robinson

See Also[mark.ces](#)**Examples**

```
data(ukdata)
plots <- extract.coverage(ukdata)
robin.dat <- extract.ch(ukdata, species=10990, plots=plots)
```

`extract.counts`*Get species data from a CES dataset*

Description

Extracts catch data for a particular species from a CES object, usually called through `extract.data`.

Usage

```
extract.counts(cesobj, species=0, age=0, plots=NULL)
```

Arguments

<code>cesobj</code>	A dataframe containing CES data usually generated by <code>/codereadces</code>
<code>species</code>	Species for which to extract data. This should be a standard Euring numeric code, for example <i>Turdus merula</i> would be 11870 (note the last digit, to indicate race, should always be zero). If <code>species=0</code> , summarises catch total across all species.
<code>age</code>	Euring age code 3 for juveniles and 4 for adults, any other value is an error.
<code>plots</code>	An object indicating which sites were covered in each year. Normally this would be generated by <code>extract.coverage</code> , if it is absent the coverage data will be extracted silently.

Value

A dataframe containing the number of birds caught at each site in each year. Usually with an identifier for the species to which the data refer.

Author(s)

Rob Robinson

See Also[extract.data](#)

extract.coverage	<i>Identify CES sites covered in each year</i>
------------------	--

Description

Creates a list which identifies which CES sites are covered in each year and the number of missing visits.

Usage

```
extract.coverage(cesdata, early=NA, late=NA, min.visits=1, all.visits=0, exclude=list(years=NULL, sites=NULL))
```

Arguments

cesdata	A CES object created by readces
early, late	Numeric vectors of length 2, the first giving the number of visits in the first/second period, the second the number of visits required in each for the site to be included. They will be reversed if necessary.
min.visits	The minimum number of visits that should be made in the season. An alternative to specifying early/late.
all.visits	The number of visits that should be made in the season. If missing will default to the highest visit number in the dataset with a warning (see details).
exclude	A list with one or two components listing sites or years that should be excluded from consideration.

Details

Creates a list containing information on coverage of sites in each year. For a site to be considered covered in a given year a certain number of visits (min.visits) must be made. Most adults will be caught earlier in the season, most juveniles later; thus an additional constraint is imposed in that a certain number (y) of the first x visits, and similarly of later visits, should also be made. This can be specified using early=c(x,y) and late=c(x,y); for example, in Britain, four of the first six and four of the last six visits are required so these would be specified as c(6,4). This has the advantage that if a site is only covered in the early period of the year, say, it can still be used for calculating adult abundance. Although normally the specification is likely to be the same for both periods, it doesn't have to be (e.g. if you have an odd number of visits), and the periods can overlap if required.

It is wise to specify all.visits explicitly, if not the highest visit number will be taken from the dataset. This can cause problems if, say, most sites do nine visits, but one site has done ten in one year, meaning that only that one season is identified as 'complete' and greatly inflating the number of 'missing' visits. The function will try to warn if it thinks this has happened.

Value

A list containing three objects

sites	A dataframe consisting of site details including: habitat, latitude, longitude, net length and first/last year of operation.
-------	--

years	A dataframe consisting of a site by year matrix and identifying whether early and late periods were covered.
coverage	A dataframe identifying number of visits in each period and whether each site/year should be included in the trend calculations.
missing.visits	A dataframe identifying which visits are missing across sites and years
all.visits	An integer specifying the number of visits that should be made.
min.visits	An integer specifying the minimum number of visits that should be made.

Note

The function calculates coverage from the catch data, thus if no birds are caught on a visit that visit will not be registered; hopefully this will be an infrequent occurrence!

Author(s)

Rob Robinson

Examples

```
data(ukdata)
sites.cov<-extract.coverage(ukdata)
```

extract.data	<i>Get data on an individual species</i>
--------------	--

Description

Creates dataframes with capture data for adult and juvenile birds from a CES object created by readces

Usage

```
extract.data(cesdata, species=0, plots=NULL, min.yrs=1)
```

Arguments

cesdata	a CES object created by readces.
species	Euring species code (including the final zero which would indicate race).
plots	an object containing plot coverage data created by extract.coverage.
min.yrs	sites covered in fewer than min.yrs will be ignored.

Value

A list of up to four items

spp Euring species code
 spp.name Name of the species in whichever language has been set by setceslang

Either or, more usually, both of the following may be present

ad.data data frame with summary of adult captures
 jv.data data frame with summary of juvenile captures

Author(s)

Rob Robinson

Examples

```
data(ukdata)
plots <- extract.coverage(ukdata)
robin.dat <- extract.data(ukdata, species=10990, plots=plots)
```

extract.sites	<i>Extracts CES site information</i>
---------------	--------------------------------------

Description

Generates a table of site information

Usage

```
extract.sites(cesdata, exclude=NULL)
```

Arguments

cesdata A CES object created by readces
 exclude A vector of sitenames to be excluded

Details

Currently mainly used to generate data for [map.ces](#).

Value

A dataframe with site (the internal numeric identifier), sitename, habitat, lat and long (in decimal degrees), netlength, number of years operated (nyears), first and last year of operation, whether the site is current, and the number of missing years in the data run.

Author(s)

Rob Robinson

Examples

```
data(ukdata)
sites <- extract.sites(ukdata)
```

get.euring	<i>Find a Euring code number</i>
------------	----------------------------------

Description

Search for the Euring code number of a species.

Usage

```
get.euring(string)
```

Arguments

string A character string containing (part of) a species name, in any language.

Value

A dataframe containing the code and scientific names of species that contain the search string.

Author(s)

Rob Robinson

index	<i>Calculate indices of abundance and productivity from CES data</i>
-------	--

Description

Takes a CES object created by `extract.data` and, depending on what data are available, creates indices of the number of adults and juveniles caught and the ratio of the two, as an index of productivity.

Usage

```
index(cesdata, year=-1, begin=0, smooth=FALSE, random=FALSE, trend=0, compare=0, verbose=FALSE, visit.
```

Arguments

<code>cesdata</code>	A CES dataset create by <code>extract.data</code>
<code>year</code>	A year to act as reference year, i.e. in which the index is value is 1
<code>begin</code>	The first year for which the trend should be calculated from. May be useful if a species is not common in the first years of the dataset, but becomes so subsequently.
<code>visit.corr</code>	Logical, include the correction for missing visits when calculating indices?
<code>cl</code>	Width of confidence limits. 0.95 is default, 0.83 limits don't overlap with (approximate) probability 0.95, so may be useful when comparing (or graphing) values of the index in specific years.
<code>smooth</code>	An indicator value to determine if smoothed trends should be calculated. These are fitted with <code>mgcv : gam</code> so using a thin-plate spline and the degree of 'wiggleness' will be determined by the data.
<code>random</code>	An indicator value to determine if site and year should instead be fitted as random effects ($\sim(1 site) + (1 year) + (1 site:year)$), with both site and year as factors (only).
<code>trend</code>	An integer to indicate how many years to calculate a linear trend over. The last year of the trend will always be the last year in the dataset. If trend is greater than the number of years available then a simple linear trend will be fitted.
<code>compare</code>	An integer to indicate how many years to compare the last year to. Thus if <code>compare=5</code> and the last year is 2008 then the index value for 2008 will be compared to the average values for 2003-2007 (which will be set to 1). Earlier years will be modelled as annual fixed-effects.
<code>verbose</code>	Logical. If TRUE then the annual estimates will be printed, if FALSE then a summary test will be printed.

Details

Uses data from sites covered in more than one year.

Value

A list with some or all of the following components

<code>ad.results</code>	Model fit information for adult birds
<code>jv.results</code>	Model fit information for juvenile birds
<code>ad.results</code>	Model fit information for adult birds

Each of these three consists of three components: `model` - a `glm` object containing the model fit; `parms` - a dataframe containing the annual year parameter estimates; and `test` - a list containing either the results of statistical tests generated by using `compare` or `trend`, or additional model information if `smooth=` or `random=TRUE`.

<code>model.type</code>	a list containing the type of model fitted (<code>type</code>), the reference year (<code>refyear</code>) and the number of years over which any trend or comparison are fitted (<code>nyrs</code>)
<code>spp</code>	Euring species code
<code>spp.name</code>	Name of the species in whichever language has been set by <code>setceslang</code>

Author(s)

Rob Robinson

Examples

```
data(ukdata)
plots <- extract.coverage(ukdata)
robin.dat <- extract.data(ukdata, species=10990, plots=plots)
result <- index(robin.dat, verbose=TRUE)
res.lin <- index(robin.dat, trend=5)
```

`load.boxplot`*Load CES boxplot dataset*

Description

Loads a datafile containing summary boxplot data to use as reference for site-specific plots

Usage

```
load.boxplot(filename=NULL, name='bxpdata')
```

Arguments

filename	the name of the file containing the summarised data, typically ending in 'year.Rdata' or '_speciesxxxx.RData'
name	the name of the object to store the boxplot data in

Value

A list with three components:

ds	the (subsetting) dataset
ds.bxp	data for the boxplot in summary form
ds.attr	some useful attributes

Note

No error checking is done so make sure you load the right file!

Author(s)

Rob Robinson

See Also[save.boxplot](#), [siteplot](#)

map.ces

*Map CES Sites***Description**

Plot CES sites on a simple outline map

Usage

```
map.ces(cesobj, type='c', xlim=c(-20,30), ylim=c(35,70), pch=21, col=c('white','red'), file=NULL, width,
```

Arguments

cesobj	A CES object with site information, usually generated by <code>extract.sites()</code>
type	How to identify sites: currently 'c' for current/non-current, 'n' for number of years operating, 'h' for habitat, and 's' for site-specific colours
xlim	Minimum and maximum longitude bounds, note R will make this 'pretty'
ylim	Minimum and maximum latitude bounds, note R will make this 'pretty'
pch	A numeric value (21-25) identifying symbol type, see <code>?pch</code> , but 21=circles, 22=squares
col	A vector of colors to fill points, see <code>type</code>
lwd	if <code>pch=21:25</code> , weight of point borders, useful for highlighting e.g. current sites
cex	size of points
file	A valid filename to plot the map to, identified by extension, current options recognised are .png, .jpg (or .jpeg), .tif (or .tiff) and .ps
width, height	size of map (normally in pixels)
units	Units for the size, use 'mm' if outputting to pdf/postscript file
...	Additional arguments passed to <code>map</code> , such as <code>border</code> and <code>lty</code> that control the color and line type of the polygon borders if <code>fill=TRUE</code> is specified.

Details

The default is to color points by whether they are current (second element of `col`) or historic - the latest year present is taken to represent the 'current' year. If `type='n'` then the plot will depend on the number of colors, two will separate sites below/above the median number of years operated; three the bottom quartile, the middle 50 percent and the upper quartile (approximately) and four will split by quartile. Any expansion of 'Current', 'No Years', 'Habitat' should work (only the first letter is significant and case is not important).

Value

None

Author(s)

Rob Robinson

Examples

```
data(ukdata)
sites<-extract.sites(ukdata)
map.ces(sites)
```

mark.ces

*Estimate adult survival from CES data.***Description**

Fit CJS-with-transients model to estimate annual adult survival from CES data.

Usage

```
mark.ces(cesobj, exclude=NULL, type='+', trend=0, constant=0, compare=0, cleanup=TRUE)
```

Arguments

cesobj	A CES object with capture histories, typically created by extract.ch
exclude	A vector of sites to exclude from the survival modelling
type	How to model group-specific survivals, either additively '+' (the default) or in interaction ':', this requires the groups to have been specified in extract.ch
trend	An integer to indicate how many years to calculate a linear trend over. The last year of the trend will always be the last year in the dataset. If trend is greater than the number of years available then a simple linear trend will be fitted.
constant	An integer indicating the last n years over which survival will be modelled as constant.
compare	An integer indicating the how many n previous years to compare the last year's estimate to.
cleanup	MARK creates a series of output files for each survival model fitted. If cleanup=FALSE, these will be stored in the sub-directory 'markfiles', if cleanup=TRUE they will be deleted. x\$model\$output indicates which files are associated with the survival model stored in x.

Details

Calls MARK to fit a CJS-with-transients model for data extracted by [extract.ch](#), see which for details of data inclusion. Currently, fits the a model in which probability of residency id constant across sites and years, survival is year-dependent and recapture rates are site dependent (with that within the first year being a fixed proportion of the annual rate).

If a grouping variable was specified when the data were extracted, survivals will be modelled in relation to this variable, either additively or interactively, according to 'type'.

To fit a time-constant model ('phi(.)' in Mark notation) set 'constant' to be a number equal to the number of capture intervals in the data. If a number larger than that is given for 'constant' or 'compare' then it will be adjusted accordingly (to n or n-1).

Value

A list consisting of model (the output returned by RMark), its AIC and (corrected) number of parameters (npar), and dataframes called residency, survival, recapture and recap1 with the parameter estimate, standard error and upper and lower confidence limits (as ucl, lcl) for each parameter.

Note

Requires MARK (<http://www.phidot.org/software/mark/downloads/index.html>) to be installed (and the MarkPath variable to be set if it is not in the default location, see ?mark)

Author(s)

Rob Robinson, with thanks to Christian Kamplicher

Examples

```
## Not run:
data(ukdata)
plots <- extract.coverage(ukdata)
robin.dat <- extract.ch(ukdata, species=10990, plots=plots)
robin.res <- mark.ces(robin.dat)

## End(Not run)
```

plot.ces

Plot CES Objects

Description

Methods to quickly plot CES objects of different types.

Usage

```
## S3 method for class 'ces'
plot(x, sites = FALSE, graph = "X", N=20, sitelist=NULL, col=c('red', 'blue'), ...)
```

Arguments

x	A CES object
sites	Logical. If TRUE, for objects created by <code>extract.data()</code> , plots a summary of number of sites recording a species, if FALSE plots summary of number of captures
graph	Graph type for plotting results of <code>index()</code> : currently 'a' for adults, 'j' for juveniles, 'p' for productivity and 'x' for all three, letters may be lower or upper case.
N	Integer. If plotting an object created by <code>readces()</code> the number of species/sites to be included.
sitelist	Character vector. If plotting an object created by <code>extract.ch()</code> sites to be included, otherwise all are.
col	Character vector. If plotting an object created by <code>extract.ch()</code> colors to plot non-capture and capture occasions respectively.
...	Additional arguments, see details

Details

Plots an appropriate summary graph (or graphs) for a ces object.

For 'raw' data (created by `readces`), a summary of captures by year, visit, site and species (for the top N sites/species only) with bars indicating the inter-quartile range across years.

For 'plot' data (created by `extract.coverage`), the number of sites operating in each year, and the number of sites operating for a particular number of years.

For 'count' data (created by `extract.data`), the number of adults and juveniles caught of the species in each year (black bars indicate the number of expected birds had there been no missing visits) and the cumulative number of birds caught where sites are ranked from those catching the most to fewest.

For abundance/productivity results (created by `index`) graphs of the annual values and parameter estimates; a single trend can be produced using the 'graph=' argument, but use `plot.trend()` for prettier output. Will also accept 'ylab' and 'title'.

For capture history ('ch') data (created by `extract.ch`), a plot of the capture histories, grouped by site, each line represents a group of individuals with the same encounter history, colored by whether they were caught or not on each occasion; darker lines indicate more frequent histories. Also accepts xlab, ylab, cex (on the y-axis, possibly to fit more site labels in) and main.

Value

None

Note

In the ch plots, because birds are possibly caught twice in their first year, 'Occasion' (the x-axis) is not quite the same as Year.

Author(s)

Rob Robinson, the ch plot was inspired by one of Marc Kery's.

See Also

[plot_trend](#)

Examples

```
data(ukdata)
plots <- extract.coverage(ukdata)
robin.dat <- extract.data(ukdata, species=10990, plots=plots)
plot(robin.dat)
```

plot_trend

Plot CES trend

Description

Plots an estimated trend from CES data either to the screen or to a file.

Usage

```
plot_trend(x, type, group=NULL, file = NULL, width=480, height=480, units='px', main=FALSE, lty=c(1,2))
```

Arguments

x	A CES object
type	Which trend to plot? Can be either 'Adult' or 'Juvenile' Abundance, 'Productivity' or 'Survival'. Only the first letter needs to be given, in either lower or upper case.
group	If survival trends have been calculated by group, the group level to be plotted.
file	A valid file name, ending in either '.png', '.jpg', '.jpeg', '.svg', '.tif', '.tiff', '.ps' or '.pdf' to produce a file of the appropriate type. If file=NULL (the default) the trend will be plotted to the screen.
width	The width of the device.
height	The height of the device.
units	The units in which height and width are given. Can be px (pixels, the default), in(ches), or mm.
main	should the plot have a title? If a value other than TRUE/FALSE is given then that will be used as the title.
lty, lwd, lcol	a vector of length 1 or 2, if length 2 the first entry will be used for the trend line, the second for the confidence limits.
line	if required, the value (e.g. 1) at which a horizontal reference line will be drawn.

`rlty, rcol, rlwd` if a reference line is specified, the line type, color and width to use.
`annual` logical, plot the annual estimates too?
`pch, pcol, pcex` if `annual=TRUE`, symbol, color and size to use in plotting points.
`...` additional arguments passed to `plot`, such as `bty`, `ylab` or `ylim`.

Details

Plots a single CES trend to a file, or to the screen if no file is not specified. PNG format is small and lossless and so good for images of this type. Invisibly returns a dataframe with the annual values used to create the graph. If you wish a finer degree of control over the appearance of the graph, best import this data to your favourite graphing package, or [plot](#) this data directly in R.

Value

If assigned to an object, a dataframe with the following columns: `'years'`, `'parm'`, `'se'`, `'index'`, `'lcl'`, `'ucl'`.

Note

If you get an error about "figure margins too large", you need to either increase the height and width (of the file, or make the plotting window larger), or check you have the correct units.

Author(s)

Rob Robinson

Examples

```
##---- Should be DIRECTLY executable !! ----
##-- ==> Define data, use random,
##-- or do help(data=index) for the standard data sets.
```

print.ces

Print methods for CES objects

Description

Largely a placeholder currently, but handles summary tables gracefully

Usage

```
## S3 method for class 'ces'
print(x, ...)
```

Arguments

`x` A CES data object
`...` Other arguments, currently they will have effect!

Details

Largely a placeholder currently, but handles summary tables gracefully.

Author(s)

Rob Robinson

readces

Read in CES data

Description

Reads in CES data from a text file in EuroCES format

Usage

```
readces(file, visits='std', group.race=TRUE, fix=FALSE, winter=FALSE, verbose=FALSE)
readBTO(file, species)
```

Arguments

file	A valid filename
visits	Either 'std' to include only standard (numbered up to 13) visits, 'all' to do no checking, or a (character) vector listing allowed visit identifiers.
group.race	Aggregate records of different races (i.e. those where the last digit of the Euring code is not zero)?
fix	Logical, should presumed errors be fixed? See details.
winter	Logical, if TRUE days will be numbered from July 1, and year will be adjusted downwards by 1 for visits in January to June, so Jan 1 1985 will become day 185 of 1984.
verbose	Logical, produce a file listing records with (currently species/age/sex) errors to be fixed?
species	A numeric Euring species code

Details

For details of the file specification see the CES pages of the Euring website, the function will attempt to match similar column names; a warning will be generated if it detects anything amiss (not guaranteed!). Some of these can be safely ignored (e.g. the 'surprising species' one) if everything appears to be in order.

The function does some basic error-checking and will try to give informative messages where it detects a problem. Where records are deleted or values set to NA (missing) a warning will be given. If fix=TRUE then the function will change records as follows: - sex of all encounters for an individual will be assigned the commonest sex recorded for that species (if the sex on encounter is recorded as unknown, or the other sex). - records aged as 3 in any year other than that of first

encounter will be changed to 4. - sites with multiple coordinates will be assigned their average location (across rows) - rows will be ordered by site, followed by year, visit, species and ring number where possible the function will indicate how many records have been fixed.

Value

A dataframe with (at a minimum) the following columns: countryID (where the scheme is), site-name(as referred by the scheme), site (a sequential numeric identifier), habitat, visit (number), day, month, year, netlength, ring, species, sex, age, race, julian (day), lat, long. Unrecognised columns in the datafile will be removed. Note, 'ring' will be a combination of scheme and ring, separated by an underscore ('_'), any '.' in the ring number will be removed.

Latitude and longitude are in decimal degrees (rounded to 3 d.p. which is ~ 0.1km, and always towards the south and west, i.e. the bottom-left of the grid cell). Race will usually be the same as species (i.e. a Euring number), unless particular races were identified in the data file (e.g. as for *Motacilla alba alba/yarrelli*, see `group.race=`). The column 'julian' gives the day of year from 1 Jan (including 29 Feb in leap years).

Note

The CES programs expect birds aged as either juvenile (3) or adult (4), the function will recode ages where it can, 2 is also allowed; records with other ages (0, 1) will be deleted. Expect some warnings, some are for information only - they do not necessarily indicate an error.

`readBTO()` is provided for reading in BTO '.all' fixed format files, variables not included in this format (eg habitat, NetLength) are simply set to NA.

Author(s)

Rob Robinson

References

Euring CES pages: <https://euring.org/research/ces-europe/>

Examples

```
## Not run:
ukdata <- readces("d:/data/ces/ukdata.csv", fix=TRUE)

## End(Not run)
```

save.boxplot

Save CES boxplot dataset

Description

Creates a datafile containing summary boxplot data to use as reference for site-specific plots

Usage

```
save.boxplot(data=NULL, annual=TRUE, species=TRUE, year=0, filename='bxp', ...)
```

Arguments

data	a CES dataset created by readces
annual	logical, if TRUE create boxplot summary data by year
species	logical, if TRUE create boxplot summary data by species
year	numeric, if species=TRUE, the year for which to create the summary
filename	a character string to use as the filestem, see details
...	further arguments passed to siteplot, possibly one or more of effort=, age=, subset=

Details

Saves the necessary data to create a boxplot summarising site-specific captures using siteplot. If annual=TRUE, the filename will expand to 'filename_year.RData', if species=TRUE then the data will be saved to 'filename_speciesxxxx.RData', where xxxx is the year. If year is not specified then the last year present in the dataset will be assumed. Data may be reloaded by using load.boxplot.

Value

None. Used for its side effect of creating one or two (.RData) files.

Author(s)

Rob Robinson

See Also

[siteplot](#), [load.boxplot](#)

Examples

```
##---- Should be DIRECTLY executable !! ----  
##-- ==> Define data, use random,  
##-- or do help(data=index) for the standard data sets.
```

`select.data`*Subsets a CES dataset*

Description

Select sites from a CES data object, either from a list of specified sites or habitat.

Usage

```
select.data(x, sites = 0, habitat = "")
```

Arguments

<code>x</code>	A CES data object, typically generated by <code>readces()</code>
<code>sites</code>	A vector of site codes
<code>habitat</code>	Two-letter code identifying habitat required. Generally the available codes are 'RB' Reedbed, 'WS' Wet Scrub, 'DS' Dry/Thorn Scrub and 'WD' Woodland but certain countries may have other scheme-specific codes

Details

An easy way of subsetting a CES data object to select particular sites. Experienced users may prefer to do this directly.

Value

A ces data object

Author(s)

Rob Robinson

See Also

[readces](#)

Examples

```
data(ukdata)
sitelist = c(401, 403, 405, 407, 408, 414, 417, 447, 463) # select sites with complete data
complete = select.data(ukdata, sites=sitelist)

scrub = select.data(ukdata, habitat='DS')
```

setceslang	<i>Set language for printing species names</i>
------------	--

Description

Allows species names in outputs to be rendered in one of 14 languages.

Usage

```
setceslang(lang)
```

Arguments

lang	A string indicating a valid language, see details below. If no language is set most functions will default to using the scientific name
------	---

Details

Valid languages are "Latin", "Danish", "Dutch", "English", "Finnish", "French", "German", "Italian", "Norwegian", "Polish", "Portuguese", "Spanish", "Swedish". This is set by a call to `options()$ceslang` to check which language is currently set.

Value

Sets global option 'ceslang'

Warning

Names are 'sticky', that is when an object is created, for example with `extract.data`, `cesr` uses the language set at the time and this cannot easily be changed subsequently.

Note

Names are case sensitive, that is they need to be capitalised.

Author(s)

Rob Robinson

Examples

```
setceslang("English")
```

 siteplot

Summary plot of site-specific captures

Description

Generate boxplot of captures for all sites, by year or species, with those from one site highlighted.

Usage

```
siteplot(data, site=0, year=0, visit=FALSE, splist=NULL, nspecies=20, nyears=NA, age=0, subset=NULL, e
```

Arguments

data	either a dataframe generated by readces or a boxplot summary created by previously running siteplot, see details
site	character, the site to highlight
year	integer, specify a year to create a summary by species or visit
visit	logical, summarise catches within a year by visit number? If visit=FALSE (the default), catches will be summarised by species
splist	character, if a year is specified, optionally either a vector of Euring codes, or 'all' (not generally advised)
nspecies	numeric, if a year is specified and splist not, then plot the commonest 'n' species on the site
nyears	numeric, limit the number of years plotted to the number that the site operated, with this as a minimum. For example, if nyears=10 and the site has operated for 15 years, 15 bars (and dots) will be shown, but if the site has only operated for 5 years, then 10 bars will be plotted.
age	numeric, either 3 or 4 to select only juvenile or adult captures
subset	a logical expression to subset the data
effort	logical, should the number of captures be divided by netlength on the site?
ylab	character, a label for the number axis
margin	if year is specified, either a single number for the width of the left margin, or a vector of length 4 specifying each margin width, see details
pch	the symbol with which the site counts are plotted
pt.cex	size of the symbol with which the site counts are plotted
pt.col	colour of the symbol with which the site counts are plotted
plot	logical, should a plot be produced. For larger datasets, creating the summary data first without plotting may be helpful, see details
...	additional arguments passed to bxp for drawing the boxplot, such as 'main' (for a title) or 'whisklty' (change the whisker line type)

Details

Generates a boxplot of captures per site, with those from 'site' highlighted using an overlain point. Data may be summarised either annually (if year=0), or by species (if a year is specified). In the former case the boxes will be vertical, in the latter horizontal. The width of the margin(s) can be changed (by using margin=) if there is not enough room for the species names.

For larger datasets, creating the boxplot summaries may take some time. In this case it is advised to run the function with plot=FALSE to create the data for the boxplot. This can then be supplied to data=, meaning that only the individual site counts need to be extracted and plotted each time.

Value

A list with three components:

ds	the (subsetting) dataset
ds.bxp	data for the boxplot in summary form
ds.attr	some useful attributes

Note

It is assumed that this function will be used for current sites, so the nyears argument only counts backwards from the current year.

Author(s)

Rob Robinson

Examples

```
data(ukdata)
uk.bxp <- siteplot(ukdata, site='656')
siteplot(uk.bxp, site='420')
```

sitereport

Site reporting summary

Description

Generate a simple summary report (as MS Word documents) of captures at a site.

Usage

```
sitereport(cesdata, sitelist=NULL, year=0, template_file=NULL, outdir=getwd(), no.species=12, nyears
```

Arguments

cesdata	a dataframe generated by readces
sitelist	character, a vector of sitenames for which to generate the reports; NULL and "all" are equivalent
year	specify a year to create a summary by species
template_file	the name of an RMarkdown template file
outdir	where to save the reports to
nyears	numeric, limit the number of years plotted to the number that the site operated, with this as a minimum. For example, if nyears=10 and the site has operated for 15 years, 15 bars (and dots) will be shown, but if the site has only operated for 5 years, then 10 bars will be plotted.
no.species	numeric, maximum number of species to include in the barplots.
verbose	logical, print a message when each site has been processed?

Author(s)

Rob Robinson

Examples

```
## Not run:
data(ukdata)
sitereport(ukdata, sitelist=c('656','420'))

## End(Not run)
```

summary.ces

Summarizing CES objects

Description

Methods to summarize CES objects

Usage

```
## S3 method for class 'ces'
summary(object, age=0, sp.order='alpha', ads=FALSE, jvs=FALSE, prod=FALSE, df=FALSE, nrow=6, silent=FA
```

Arguments

object	A CES object
ads, jvs, prod	logical, if TRUE and x contains index results, print summary statistics for adult/juvenile abundance/productivity only.
age	Euring age code - either 3 juveniles or 4 for adults

<code>sp.order</code>	Should species be displayed in order of the 'alpha'bet, decreasing 'count' or, if neither specified, 'taxon'omic (Voous) order.
<code>df</code>	logical, if TRUE and x is CES data object created by <code>readces</code> , print traditional column summary statistics.
<code>nrow</code>	integer, if x is a species summary object, print this many rows, ordered by highest count.
<code>silent</code>	logical, if TRUE suppress printing of the species counts. May be useful if you just want to generate a table of species counts for further manipulation.
<code>...</code>	arguments passed to other methods.

Details

`summary()` will call the appropriate method as follows: If given an object created by `readces` it will print a table of number of each species caught (only those of 'age' if that argument is given) in 'sp.order' If given an object created by `extract.data` it will provide a simple summary of the number of birds caught (or only those of 'age' if that argument is given) If given an object created by `extract.coverage` it will provide a summary of the number of plots operated in each year and the number of years plots have been operating If given an object created by `index` it will provide summary statistics of the fit

Normally one will want to summarise results for both age-classes and productivity measure, but the facility is provided to do this for only one age-class if desired.

Value

Invisibly, a dataframe (or a list of dataframes) containing the numerical data.

Author(s)

Rob Robinson

Examples

```
data(ukdata)
summary(ukdata)
```

<code>transient_plot</code>	<i>Visualise capture frequency</i>
-----------------------------	------------------------------------

Description

CES sites may have an excess of individuals that are caught only once, and so potentially transient to the site. This function plots individuals by whether they were captured subsequently or not.

Usage

```
transient_plot(cesobj, col=c('blue', 'red'), border='gray10', leg.size=0.8, ...)
```

Arguments

<code>cesobj</code>	A CES object created by <code>extract.ch</code>
<code>col</code>	Character vector. Colors to plot re-captured and not re-captured individuals respectively.
<code>border</code>	Character. Color for the border of the bars.
<code>leg.size</code>	Numeric. Size of the legend text.
<code>...</code>	Additional parameters passed to <code>legend</code> .

Details

CES sites may have an excess of individuals that are caught only once, and so potentially transient to the site. This function plots individuals by whether they were captured subsequently or not, specifically whether they were captured twice in their first year (and so likely to be resident), captured only once in their first year (but caught again in a later year, and so also resident), and individuals caught only once. These latter may be transients (individuals passing through, or living on the very edge of the catching area, so with low capture probability) or resident individuals that happened not to be recaptured. Two barplots are produced, one classifying individuals by their year of first capture, and the second by site. Transients have the potential to bias survival estimates low as, once they leave the site (and so can no longer be captured), the models see them as dead. The function to calculate survival (`mark.ces`) tries to estimate the proportion of these individuals in the sample.

Value

A list with three elements. (i) `years` - the matrix necessary to draw the years barplot, (ii) `sites` - the matrix necessary to draw the sites barplot, and (iii) the species name.

Author(s)

Rob Robinson

ukdata

A sample CES dataset

Description

A sample CES dataset from the UK of two species (*Erithacus rubecula* and *Turdus philomelos*) from 19 sites created by `readces`

Usage

`data(ukdata)`

Format

A data frame with 4,879 observations on the following 15 variables.

countryID a factor with levels GBT - the scheme identifier using Euring three-letter codes
 habitat a factor with levels DS RB WD WS - for Dry/Thorn Scrub, Reedbed, Woodland and Wet Scrub respectively
 visit numeric - 1:12 for UK
 day numeric
 month numeric - 1:12
 year numeric - four digits
 netlength numeric - length of net erected (in metres)
 scheme a factor with levels GBT - of ring, allows for foreign controls
 ring a factor with levels 012083 012098 012100 ...
 species numeric - Euring numeric, aggregates races so will end in zero
 sex a factor with levels F M U
 age numeric - either 3 (juvenile) or 4 (adult)
 sitename a site identifier, numeric in this case, may be character for other schemes, identical to siteID
 site numeric - a site identifier, with sites numbered 1:n
 race numeric - Euring code as in original datafile, may specify races
 julian numeric - ordinal day (in year) of capture
 lat numeric - latitude in decimal degrees
 long numeric - longitude in decimal degrees, East of the Greenwich meridian is positive

Examples

```
data(ukdata)
summary(ukdata)
plot(ukdata)
```

writeces

Write CES data to a file

Description

Provides a wrapper function to write CES data objects to a comma-separated text file

Usage

```
writeces(cesobj, file = "", na.string = ".", age = 0, ads = FALSE, jvs = FALSE, prod = FALSE, verbose=TRUE)
```

Arguments

cesobj	a CES object
file	A valid filename
na.string	Character(s) to use to represent missing values in the output file
age	Euring age code - 3 for juvenile or 4 for adult
ads, jvs, prod	logical, if TRUE write out data for adult/juvenile abundance/productivity only.
verbose	logical, if TRUE write out individual visit data writing plot data.

Details

Write an object of class CES to a comma-separated text file. Use argument 'age' to just write juvenile or adult data. If outputting the results of a model fit from `index`, normally one will want to summarise results for both age-classes and productivity measure, but the facility is provided to do this for one age-class only if desired (by setting `ads`, `jvs` or `prod` equal to TRUE respectively).

Author(s)

Rob Robinson

Examples

```
## Not run:  
data(ukdata)  
robin.dat=extract.data(ukdata,species=10990)  
writeces(robin.dat, file="d:\data\robin.csv")  
  
## End(Not run)
```


Index

- * **IO**
 - readces, 21
 - writces, 31
 - * **attribute**
 - summary.ces, 28
 - * **datasets**
 - cesnames, 5
 - ukdata, 30
 - * **data**
 - load.boxplot, 14
 - * **dplot**
 - siteplot, 26
 - * **hplot**
 - map.ces, 15
 - plot.ces, 17
 - plot_trend, 19
 - siteplot, 26
 - * **manip**
 - extract.ch, 6
 - extract.counts, 8
 - extract.coverage, 9
 - extract.data, 10
 - extract.sites, 11
 - select.data, 24
 - * **models**
 - index, 12
 - mark.ces, 16
 - * **package**
 - cesr-package, 2
 - * **regression**
 - index, 12
 - mark.ces, 16
 - * **sysdata**
 - setceslang, 25
 - * **utilities**
 - check_rings, 6
 - print.ces, 20
 - transient_plot, 29
- ces.table, 4
- cesnames, 5
- cesr (cesr-package), 2
- cesr-package, 2
- check_rings, 6
- extract.ch, 3, 4, 6, 16
- extract.counts, 8
- extract.coverage, 3, 4, 9
- extract.data, 3, 8, 10
- extract.sites, 11
- get.euring, 4, 12
- index, 3–5, 12
- load.boxplot, 14, 23
- map.ces, 11, 15
- mark.ces, 3, 7, 8, 16
- plot, 20
- plot.ces, 17
- plot_trend, 3, 19, 19
- print.ces, 20
- readBTO (readces), 21
- readces, 3, 4, 21, 24
- save.boxplot, 14, 22
- select.data, 24
- setceslang, 5, 25
- siteplot, 14, 23, 26
- sitereport, 27
- summary.ces, 28
- transient_plot, 29
- ukdata, 30
- writces, 31