

Package ‘biogeo’

October 12, 2022

Type Package

Title Point Data Quality Assessment and Coordinate Conversion

Version 1.0

Date 2016-03-10

Author Mark Robertson

Maintainer Mark P. Robertson <markrobertsonsa@gmail.com>

Description Functions for error detection and correction in point data quality datasets that are used in species distribution modelling. Includes functions for parsing and converting coordinates into decimal degrees from various formats.

LazyData TRUE

License GPL (>= 3)

URL <http://onlinelibrary.wiley.com/doi/10.1111/ecog.02118/abstract>

Depends R (>= 3.1.0)

Imports raster, stringr, maptools, vegan, sp

Suggests dismo (>= 0.9-3)

NeedsCompilation no

Repository CRAN

Date/Publication 2016-04-08 10:43:59

R topics documented:

biogeo-package	3
addmainfields	3
alternatives	4
alternatives2	6
alternativesenv	7
checkdatastr	9
coord2numeric	10
dat	10
datm	11

dd2dmslat	12
dd2dmslong	13
dem	14
dms2dd	15
dmsabs	16
dmsparse	16
dmsparsefmt	17
duplicatesexclude	18
edat	19
elevcheck	20
env2stack	21
errorcheck	22
fieldsmissing	23
finddecimals	24
fmtcheck	25
gbifdat	26
geo2envid	27
geo2envpca	28
getextent	30
getformat	31
getletter	31
keepmainfields	32
missingcoords	33
missingvalsexclude	34
modified	35
modifiedtoday	35
msk60	36
nearestcell	37
outliers	38
parsecoords	39
places	40
plotsetup	40
points2shape	41
pointsworld	42
precisioncheck	43
precisionenv	44
quickclean	45
quickrich	46
renamefields	47
richness	48
richnessmap	49
rjack	50
sep	51
speciescount	51
substdmm	52
uniqueformats	53
wclim	54
world	55

biogeo-package	<i>Point data quality assessment and coordinate conversion</i>
----------------	--

Description

Functions for error detection and correction in point data quality datasets that are used in species distribution modelling. Includes functions for parsing and converting coordinates into decimal degrees from various formats.

Details

Package: biogeo

Type: Package

Version: 1.0

Date: 2015-08-27

License: LGPL-2

Depends: "R (>= 3.1.0)", "sp (>=1.0-15)", "raster (>= 2.2-31)", "rgdal (>=0.8-16)", "stringr (>= 0.6.2)", "maptools (>= 0.8-3)

Author(s)

Mark Robertson

Maintainer: Mark P. Robertson <mrobertson@zoology.up.ac.za>

References

see tutorial document

See Also

maptools, raster, dismo

addmainfields	<i>Adds required fields to a dataframe containing coordinates for points</i>
---------------	--

Description

Several functions require a particular set of columns with specific names. The required fields are: ID, x, y, Species, x_original, y_original, Correction, Modified, Reason and Exclude. This function adds these columns to the dataframe.

Usage

```
addmainfields(dat, species)
```

Arguments

`dat` a dataframe containing coordinates for a set of points (collection localities)
`species` the name of the field containing the species names

Value

a dataframe with the required fields added

Author(s)

Mark Robertson and Vernon Visser

See Also

`checkdatastr`, `renamefields`, `keepmainfields`

Examples

```
dat1<-dat[,1:4]
addmainfields(dat1, species="Species")
```

alternatives	<i>Determine where an incorrect point record should be placed by showing alternative positions for that point based on common errors in datasets.</i>
--------------	---

Description

This is an interactive plot. The alternative positions are determined by: transposing the x- and y-coordinates, changing the sign on x-coordinate, changing the sign on y-coordinate, changing signs on both coordinates, transposing degrees and minutes, transposing the coordinates but not their signs.

Usage

```
alternatives(dat, group1 = "Species", group2 = "",
world, rst, locality = "", pos = "bottomleft",
ext = c(-180, 180, -60, 90))
```

Arguments

dat	a dataframe containing fields with the following names: ID, x, y, Species, x_original, y_original, Correction, Modified, Exclude
group1	this is usually the column of species names (default = Species)
group2	this is a second grouping variable
world	a shapefile of the world, where the column containing the country names must be "NAMES"; see data(wrld_simpl)
rst	a raster (see raster package)
locality	a column in the dataframe containing the locality description for the point
pos	position of the legend when group2 is used (e.g. bottomleft)
ext	The extent, which can be specified as c(xmin, xmax, ymin, ymax) default extent c(-180, 180, -90, 90). Alternatively if ext="p" then the extent will be calculated from the coordinates of the points in the dataset.

Details

The user starts by clicking on a record of interest. Then alternative positions for that record are displayed using purple point symbols. All other records for that particular species are indicated in black. The user then clicks on the position of the correct record, or back on the originally selected record to exclude it. If none of the alternative points are correct then the stop button should be selected (top left of plot screen). The identifier (ID) of the record is displayed next to the point and its coordinates and species name are displayed at the top of the map. Once a new position for the point is selected then the new coordinates for that point are displayed at the bottom of the map. When a record is changed then all records with identical x- and y-coordinates will also be changed in the same way. This is because several different species may have been collected at the same locality.

Value

An interactive plot is produced and the x- and y-coordinates are updated according to the selection of an alternative point by the user. The original values of the x- and y-coordinates will be written into the fields x_original and y_original. The date and time that the record was modified will be written into the field called Modified. The type of correction will be recorded as a number in the field called Corrected.

Author(s)

Mark Robertson and Vernon Visser

See Also

alternatives2, alternativesenv

Examples

```
## Not run:
dem<-raster(dem,xmn=-180, xmx=180, ymn=-60, ymx=90)
dat<-alternatives(dat,group1="Species",group2="",world,dem,locality="",pos="bottomleft",ext="p")
## End(Not run)
```

alternatives2	<i>Determine where an incorrect point record should be placed by showing alternative positions for that point based on common errors in datasets.</i>
---------------	---

Description

This function is similar to alternatives but the difference is that the user specifies a species so that only records for that species are shown in the plot.

Usage

```
alternatives2(dat, g1, group1 = "Species", group2 = "",
world, rst, locality = "", pos = "bottomleft",
ext = c(-180, 180, -60, 90))
```

Arguments

dat	a dataframe containing fields with the following names: ID, x, y, Species, x_original, y_original, Correction, Modified, Exclude
g1	the name of the species to select
group1	this is usually the column of species names (default = Species)
group2	this is a second grouping variable
world	a shapefile of the world, where the column containing the country names must be "NAMES"; see data(wrld_simpl)
rst	a raster (see raster package)
locality	a column in the dataframe containing the locality description for the point
pos	position of the legend when group2 is used (e.g. bottomleft)
ext	The extent, which can be specified as c(xmin, xmax, ymin, ymax) default extent c(-180, 180, -90, 90). Alternatively if ext="p" then the extent will be calculated from the coordinates of the points in the dataset.

Details

The user starts by clicking on a record of interest. Then alternative positions for that record are displayed using purple point symbols. The user then clicks on the position of the correct record, or back on the originally selected record to exclude it. If none of the alternative points are correct then the stop button should be selected (top left of plot screen). The identifier (ID) of the record is displayed next to the point and its coordinates and species name are displayed at the top of the

map. Once a new position for the point is selected then the new coordinates for that point are displayed at the bottom of the map. When a record is changed then all records with identical x- and y-coordinates will also be changed in the same way. This is because several different species may have been collected at the same locality.

Value

An interactive plot is produced and the x- and y-coordinates are updated according to the selection of an alternative point by the user. The original values of the x- and y-coordinates will be written into the fields `x_original` and `y_original`. The date and time that the record was modified will be written into the field called `Modified`. The type of correction will be recorded as a number in the field called `Corrected`.

Author(s)

Mark Robertson and Vernon Visser

See Also

`alternatives`, `alternativesenv`

Examples

```
## Not run:
dem<-raster(dem,xmn=-180, xmx=180, ymn=-60, ymx=90)
dat<-alternatives2(dat,g1="Species A",group1="Species",group2="",
world,dem,locality="",pos="bottomleft",ext="p")
## End(Not run)
```

<code>alternativesenv</code>	<i>Determine where an incorrect point record should be placed by showing alternative positions for that point based on common errors in datasets.</i>
------------------------------	---

Description

View alternative points in an environmental space

Usage

```
alternativesenv(dat, g1, group1 = "Species", ev, vars,
world, xname = "", yname = "", rst, locality = "",
ext = c(-180, 180, -60, 90))
```

Arguments

<code>dat</code>	a dataframe containing fields with the following names: ID, x, y, Species, x_original, y_original, Correction, Modified, Exclude and the values of selected environmental variables
<code>g1</code>	the name of the species to select
<code>group1</code>	this is usually the column of species names
<code>ev</code>	a raster stack of environmental variables
<code>vars</code>	a character vector of two environmental variable names that will be used to define the environmental space. Values of these variables must be present in dat.
<code>world</code>	a shapefile of the world, where the column containing the country names must be "NAMES"; see data(wrld_simpl)
<code>xname</code>	x-axis label for the environmental space
<code>yname</code>	y-axis label for the environmental space
<code>rst</code>	a raster
<code>locality</code>	the name of the column of dat containing locality descriptions
<code>ext</code>	The extent, which can be specified as c(xmin, xmax, ymin, ymax) default extent c(-180, 180, -90, 90). Alternatively if ext="p" then the extent will be calculated from the coordinates of the points in the dataset.

Details

This is an interactive plot of geographical and environmental space. Alternative positions for records selected in the geographical space are plotted in geographical space and environmental space.

Value

An interactive plot is produced and the x- and y-coordinates are updated according to the selection of an alternative point by the user. The original values of the x- and y-coordinates will be written into the fields x_original and y_original. The date and time that the record was modified will be written into the field called Modified. The type of correction will be recorded as a number in the field called Corrected.

Author(s)

Mark Robertson and Vernon Visser

See Also

alternatives, alternatives2, geo2envid, geo2envpca

Examples

```
## Not run:
fd<-system.file(package="biogeo")
foldenv<-file.path(fd,"inst","extdata", fsep = .Platform$file.sep)
ev<-env2stack(foldenv, vars = "", fext="bil")
dem<-raster(dem,xmn=-180, xmx=180, ymn=-60, ymx=90)
plotsetup(6,6)
g1="Species U"
vars=c("bio1","bio12")
d5<-alternativesenv(edat,g1,group1="Species",ev,vars,world,
xname="Annual Mean Temperature",yname="Annual Precipitation",
dem,locality="LocalityName",ext="p")
## End(Not run)
```

checkdatastr

Checks data structure

Description

Checks the data structure to see which of the required columns are missing from the dataframe

Usage

```
checkdatastr(dat)
```

Arguments

`dat` a dataframe containing several columns, including the x- and y-coordinates

Details

Several functions require a particular set of columns with specific names. The required fields are: ID, x, y, Species, x_original, y_original, Correction, Modified, Reason and Exclude. Required fields that are missing can be added using `addmainfields`.

Value

returns a dataframe of two columns, one containing the names of the required fields (Field) and the other (Present) indicating whether or not that field is present (TRUE or FALSE)

Author(s)

Mark Robertson and Vernon Visser

See Also

`addmainfields`, `renamefields`, `keepmainfields`

Examples

```
data(dat)
a<-checkdatastr(dat)
```

coord2numeric	<i>Converts coordinates that are factors into numeric values</i>
---------------	--

Description

Converts coordinates that are factors into numeric values

Usage

```
coord2numeric(xn)
```

Arguments

xn	coordinate
----	------------

Value

coordinates as numeric

Author(s)

Mark Robertson

Examples

```
xn<-as.factor(c("-25.345", "35.187", "-34.563"))
coord2numeric(xn)
```

dat	<i>Species collection records dataset</i>
-----	---

Description

A collection records for a number of insect species (Species A to Species U) containing common errors. The errors were inserted into the dataset to illustrate the use of the functions in the package.

Usage

```
data(dat)
```

Format

A data frame with 1694 observations on the following 11 variables.

ID a numeric vector - unique identifiers

Species a character vector - species names (Species A to Species U)

Country a character vector - country of collection

x a numeric vector - x-coordinate in decimal degrees

y a numeric vector - y-coordinate in decimal degrees

LocalityName a character vector - name of locality of collection

x_original a logical vector - original x-coordinate

y_original a logical vector - original y-coordinate

Correction a character vector - a number associated with a particular correction

Modified a character vector - date and time the record was modified

Exclude a numeric vector - values of one indicate that the record should be excluded, zero if to be included

Reason a character vector - indicates the reason for excluding the record

Examples

```
data(dat)
head(dat)
```

datm	<i>A dataset for a marine species</i>
------	---------------------------------------

Description

A dataset for a marine species

Usage

```
data("datm")
```

Format

A data frame with 500 observations on the following 10 variables.

ID a numeric vector - unique identifiers

Species a factor with levels - the name of the species

x a numeric vector - x-coordinate in decimal degrees

y a numeric vector - y-coordinate in decimal degrees

x_original a logical vector - original x-coordinate

y_original a logical vector - original y-coordinate

Correction a factor with levels - a number associated with a particular correction

Modified a factor with levels - date and time the record was modified

Exclude a numeric vector - values of one indicate that the record should be excluded, zero if to be included

Reason a factor with levels - reason that record was excluded

Source

Data originally obtained from GBIF www.gbif.org, with certain fields removed and the species name replace with "Marine example"

Examples

```
data(datm)
```

dd2dmslat	<i>Convert decimal degree coordinates for latitude into degrees, minutes and seconds</i>
-----------	--

Description

Convert decimal degree coordinates for latitude into degrees, minutes and seconds

Usage

```
dd2dmslat(decdeg)
```

Arguments

decdeg a vector of decimal degrees for latitude

Value

a dataframe with degrees, minutes and seconds in separate columns

Author(s)

Mark Robertson

References

a dataframe with degrees, minutes, seconds and N or S in separate columns

See Also

dd2dmslong

Examples

```
data(dat)
dd2dmslat(dat$y)
```

dd2dmslong	<i>Convert decimal degree coordinates for longitude into degrees, minutes and seconds</i>
------------	---

Description

Convert decimal degree coordinates for longitude into degrees, minutes and seconds

Usage

```
dd2dmslong(decdeg)
```

Arguments

decdeg a vector of decimal degrees for longitude

Value

a dataframe with degrees, minutes, seconds and E or w in separate columns

Author(s)

Mark Robertson

See Also

dd2dmslat

Examples

```
data(dat)
dd2dmslong(dat$x)
```

dem

*Digital elevation model 10 minute spatial resolution***Description**

A digital elevation model at 10 minute spatial resolution

Usage

```
data("dem")
```

Format

The format is: Formal class 'RasterLayer' [package "raster"] with 12 slots ..@ file :Formal class '.RasterFile' [package "raster"] with 13 slots@ name : chr "c:\projects\biogeo\datasets\alt.bil"@ datanotation: chr "INT2S"@ byteorder : chr "little"@ nodatavalue : num -Inf@ NAchanged : logi FALSE@ nbands : int 1@ bandorder : chr "BIL"@ offset : int 0@ toptobottom : logi TRUE@ blockrows : int 1@ blockcols : int 2160@ driver : chr "gdal"@ open : logi FALSE ..@ data :Formal class 'SingleLayerData' [package "raster"] with 13 slots@ values : logi(0)@ offset : num 0@ gain : num 1@ inmemory : logi FALSE@ fromdisk : logi TRUE@ isfactor : logi FALSE@ attributes: list()@ haveminmax: logi TRUE@ min : num -353@ max : num 6241@ band : int 1@ unit : chr ""@ names : chr "alt" ..@ legend :Formal class 'RasterLegend' [package "raster"] with 5 slots@ type : chr(0)@ values : logi(0)@ color : logi(0)@ names : logi(0)@ colortable: logi(0) ..@ title : chr(0) ..@ extent :Formal class 'Extent' [package "raster"] with 4 slots@ xmin: num -180@ xmax: num 180@ ymin: num -60@ ymax: num 90 ..@ rotated : logi FALSE ..@ rotation:Formal class '.Rotation' [package "raster"] with 2 slots@ geotrans: num(0)@ transfun:function () ..@ ncols : int 2160 ..@ nrows : int 900 ..@ crs :Formal class 'CRS' [package "sp"] with 1 slots@ projargs: chr "+proj=longlat +ellps=WGS84 +towgs84=0,0,0,0,0,0,0 +no_defs" ..@ history : list() ..@ z : list()

Details

A digital elevation model obtained from worldclim

Source

www.worldclim.org

References

Hijmans, R.J., S.E. Cameron, J.L. Parra, P.G. Jones and A. Jarvis, 2005. Very high resolution interpolated climate surfaces for global land areas. *International Journal of Climatology* 25: 1965-1978.

Examples

```
data(dem)
```

dms2dd	<i>Converts coordinates in degrees, minutes, seconds into decimal degrees</i>
--------	---

Description

Converts coordinates in degrees, minutes, seconds into decimal degrees. These can be latitude or longitude.

Usage

```
dms2dd(dd, mm, ss, ns)
```

Arguments

dd	degrees of latitude or longitude
mm	minutes of latitude or longitude
ss	seconds of latitude or longitude
ns	letters (N,S,E,W)

Value

returns decimal degrees

Author(s)

Mark Robertson

See Also

dd2dmslat, dd2dmslong, dmsparse, dmsabs

Examples

```
dd<-c(23, 45, 34)
mm<-c(45, 34, 22)
ss<-c(2, 56, 10)
ns<-c("E", "W", "N")
dms2dd(dd, mm, ss, ns)
```

dmsabs	<i>Separates a coordinate string into degrees, minutes and seconds when there are no delimiters</i>
--------	---

Description

A format string is used (fmt) to indicate which values in the input string are associated with degrees, minutes and seconds.

Usage

```
dmsabs(coordstr, fmt)
```

Arguments

coordstr	a coordinate string without delimiters
fmt	a format string where d indicates degrees, m indicates minutes and s indicates seconds, L indicates a letter (N,S,E,W)

Value

returns a dataframe with degrees, minutes, seconds and letter in separate columns

Author(s)

Mark Robertson

See Also

dmsparse

Examples

```
dmsabs("234513S", "ddmmsL")
dmsabs("23_45_13_S", "dd_mm_ss_L")
```

dmsparse	<i>Parse coordinate strings into separate degree, minute and second fields</i>
----------	--

Description

Separates (parse) coordinates into separate columns, provided that there are delimiters between the degrees, minutes and seconds e.g. 27 27'E. There can be several different formats and delimiters in the input.

Usage

```
dmparse(dat, x = "long", y = "lat", id = "ID")
```

Arguments

dat	A dataframe containing columns for latitude, longitude and a unique identifier (ID)
x	The name of the longitude column (default long)
y	The name of the latitude column (default lat)
id	A column containing a unique identifier for that row. The name of the field must be "ID".

Value

A dataframe containing the input columns and new columns for degrees, minutes, seconds and decimal degrees for latitude and longitude

Author(s)

Mark Robertson and Vernon Visser

See Also

dms2dd, getformat, parsecoords, dmsabs

Examples

```
## Not run:  
head(places)  
p<-dmparse(places,x='long',y='lat',id='id')  
## End(Not run)
```

dmparsefmt

Parse coordinate string using a format string

Description

Parse coordinate string into degrees, minutes and seconds using a format string

Usage

```
dmparsefmt(x, fmt)
```

Arguments

x	a character vector of coordinates
fmt	a format string specifying the format to be used e.g. 'dd.mm L'

Value

a dataframe with the degrees, minutes and seconds of the coordinate in separate columns

Author(s)

Mark Robertson

See Also

parsecoords

Examples

```
x<-c('44.25 E', '21.20 E', '14.03 E')
dmsparsefmt(x,fmt="dd.mm L")
```

duplicatesexclude *Exclude duplicate point records per species per grid cell*

Description

When there is more than one point record per grid cell for a species then the duplicates will be excluded. The grid cell size is specified by res.

Usage

```
duplicatesexclude(dat,res)
```

Arguments

dat	a dataframe containing fields with the following names: ID, x, y, Species and Exclude
res	the spatial resolution in minutes

Value

a dataframe in which values in the column Exclude are set to 1 when records represent duplicates for that species in a particular grid cell.

Author(s)

Mark Robertson and Vernon Visser

See Also

addmainfields, renamefields, keepmainfields, checkdatastr

Examples

```
a<-duplicatesexclude(dat,res=10)
```

edat

Species collection records dataset and environmental variables data

Description

A collection records for two insect species (Species T and Species U) containing common errors. The errors were inserted into the dataset to illustrate the use of the functions in the package. The dataset also contains values of 19 environmental variables.

Usage

```
data(edat)
```

Format

A data frame with 45 observations on the following 30 variables.

ID a numeric vector - unique identifiers

Species a character vector - species names

Country a character vector - country of collection

x a numeric vector - x-coordinate in decimal degrees

y a numeric vector - y-coordinate in decimal degrees

LocalityName a character vector - name of locality of collection

x_original a logical vector - original x-coordinate

y_original a logical vector - original y-coordinate

Correction a character vector - a number associated with a particular correction

Modified a character vector - date and time the record was modified

Exclude a numeric vector - values of one indicate that the record should be excluded, zero if to be included

Reason a character vector - indicates the reason for excluding the record

elev a numeric vector - elevation in meters

bio1 a numeric vector - values for Annual Mean Temperature

bio10 a numeric vector - values for Mean Temperature of Warmest Quarter

bio11 a numeric vector - values for Mean Temperature of Coldest Quarter

bio12 a numeric vector - values for Annual Precipitation

bio13 a numeric vector - values for Precipitation of Wettest Month

bio14 a numeric vector - values for Precipitation of Driest Month

bio15 a numeric vector - values for Precipitation Seasonality

bio16 a numeric vector - values for Precipitation of Wettest Quarter

bio17 a numeric vector - values for Precipitation of Driest Quarter

bio18 a numeric vector - values for Precipitation of Warmest Quarter
 bio19 a numeric vector - values for Precipitation of Coldest Quarter
 bio2 a numeric vector - values for Mean Diurnal Range
 bio3 a numeric vector - values for Isothermality
 bio4 a numeric vector - values for Temperature seasonality
 bio5 a numeric vector - values for Max Temperature of Warmest Month
 bio6 a numeric vector - values for Min Temperature of Coldest Month
 bio7 a numeric vector - values for Temperature Annual Range
 bio8 a numeric vector - values for Mean Temperature of Wettest Quarter
 bio9 a numeric vector - values for Mean Temperature of Driest Quarter

Details

values for these variables were obtained from Worldclim.org 10 minute spatial resolution bioclimatic data

Source

Environmental variable data obtained from Worldclim.org

Examples

```
data(edat)
head(edat)
```

elevcheck

Elevation check

Description

Compares the recorded elevation values for records with values extracted from a digital elevation model and determines whether there is a mismatch in values.

Usage

```
elevcheck(dat, dem, elevc = "elevation", diff = 50)
```

Arguments

dat	A dataframe containing the required fields, including: ID, x, y, Species, x_original, y_original, Correction, Modified, Reason and Exclude. Required fields that are missing can be added using addmainfields.
dem	A digital elevation model as a raster object.
elevc	The field containing the recorded elevation (or depth) values in meters.
diff	A threshold value for determining a mismatch between the recorded elevation and the elevation extracted from the digital elevation model. If values exceed this value then they are flagged as being mismatches.

Value

A dataframe of two fields. Mismatches are indicated as ones in elevMismatch and the elevation values extracted from the digital elevation model are in demElevation.

Author(s)

Mark Robertson

See Also

errorcheck, quickclean

Examples

```
## Not run:
gb <- keepmainfields(gbifdat,ID='',Species='species',x='decimallongitude',y='decimallatitude',
  others=c('gbifid','elevation')) # Convert example data to biogeo format
gb <- gb[ gb$Species=='Heterotheca villosa', ] # Keep data for only one species
dem<-raster(dem,xmn=-180, xmx=180, ymn=-60, ymx=90)
gba<-elevcheck(gb,dem,elevc="elevation",diff=50)

## End(Not run)
```

env2stack

Read environmental variable rasters

Description

Reads environmental variable rasters from a selected folder into a raster stack

Usage

```
env2stack(foldenv, vars = "", fext)
```

Arguments

foldenv	a folder containing the environmental variables
vars	names of specific variables that should be selected
fext	file extension e.g. "bil", "grd" or "asc"

Value

a raster stack containing the environmental variables

Author(s)

Mark Robertson

See Also

wclim, extract, alternativesenv

Examples

```
fd<-system.file(package="dismo")
foldenv<-file.path(fd,"ex", fsep = .Platform$file.sep)
ev<-env2stack(foldenv, vars = c("bio1","bio12","bio5","bio6"), fext="grd")
```

errorcheck

Identifies errors in a dataset of point records

Description

It searches for mismatches between country names in the dataset and those extracted using the point records. Records for which there are no environmental data (based on the object rst) are indicated with a value of one in the field called wrongEnv. Low precision records are indicated by a value of one in the field lowprec. Environmental outliers are indicated by a value of one in a field beginning with the name of the environmental variable and ending either in "_e" for records assessed using boxplot statistics (e.g. bio1_e) or ending in "_j" for records assessed using the reverse jackknife procedure. The recorded elevation values for records (specified with a field name in elevc) are compared to digital elevation model values (which are returned in the field demElevation) and indicated as a mismatch if they exceed the value specified in the parameter called diff (the difference in metres).

Usage

```
errorcheck(world, dem, dat, countries="", countryfield="NAME",
vars=c("bio1", "bio12", "bio5", "bio6"), res=10,elevc="",diff=50)
```

Arguments

world	A shapefile of the world, with the column containing the country names. See data(wrld_simpl)
dem	a digital elevation model raster
dat	A dataframe containing the required fields, including: ID, x, y, Species, x_original, y_original, Correction, Modified, Reason and Exclude. Required fields that are missing can be added using addmainfields.
countries	The name of the field in dat with country names.
countryfield	The name of the field in the shape file containing the country names. In wrld_simpl this field is "NAME".
vars	The names of the environmental variables to be used.
res	The spatial resolution of the grid to be created for identifying duplicate records. This value is given in minutes.
elevc	A field containing elevation or depth values for points.

`diff` The difference between the elevation recorded and the elevation extracted from the DEM. Absolute differences that are greater than this value will be indicated as elevation mismatches.

Details

The field called "error" will contain a value of one if there are any values of one in CountryMismatch, CountryMismatch, wrongEnv or any of the outlier fields. The field called "spperr" will contain ones for all records of a species for which there are one or more errors.

Value

An error is returned if any of the fields are missing.

Author(s)

Mark Robertson and Vernon Visser

See Also

`addmainfields`, `fieldsmissing`, `renamefields`, `env2stack`, `precisioncheck`, `quickclean`, `quickrich`

Examples

```
## Not run:
dem<-raster(dem,xmn=-180, xmx=180, ymn=-60, ymx=90)
d8 <- errorcheck(world, dem, dat=edat, countries="Country", countryfield="NAME",
vars=c("bio1", "bio12", "bio5", "bio6"), res=10,elevc="elev",diff=50)

## End(Not run)
```

<code>fieldsmissing</code>	<i>Determines whether any of the required fields are missing from the dataframe</i>
----------------------------	---

Description

Several functions require a particular set of columns with specific names. The required fields are: ID, x, y, Species, x_original, y_original, Correction, Modified, Reason and Exclude. Required fields that are missing can be added using `addmainfields`.

Usage

```
fieldsmissing(dat, fields)
```

Arguments

`dat` a dataframe containing several columns, including the x- and y-coordinates
`fields` a character vector of the field names that are required

Value

An error is returned if any of the fields are missing

Note

this function is used in pointsworld and several other functions

Author(s)

Mark Robertson and Vernon Visser

See Also

checkdatastr, addmainfields

Examples

```
data(dat)
fieldsmissing(dat, fields=c("ID", "x", "y"))
```

finddecimals

Find coordinates that are in decimal degrees

Description

Finds indices for coordinates that are in decimal degrees in a dataframe

Usage

```
finddecimals(dat, x = "x", y = "y")
```

Arguments

dat	A dataframe containing latitude and longitude coordinates in separate columns
x	the name of the column containing the x-coordinate (longitude)
y	the name of the column containing the y-coordinate (latitude)

Value

index

Author(s)

Mark Robertson and Vernon Visser

See Also

dmsparse

Examples

```
finddecimals(places,x='long',y='lat')
```

fmtcheck	<i>Coordinate string format check</i>
----------	---------------------------------------

Description

checks that a coordinate string is not NA, does not have an empty string or if has no numbers i.e. if it is a valid coordinate

Usage

```
fmtcheck(x)
```

Arguments

x a coordinate string

Details

used in dmsparse

Value

returns a value of 1 if the coordinate has either an NA, an empty string or has no numbers or returns a zero if not

Author(s)

Mark Robertson

See Also

dmsparse

Examples

```
a<-fmtcheck(x="no coordinate")
a<-fmtcheck(x="23_45S")
```

`gbifdat`*A dataset of records from GBIF*

Description

A dataset of records from the Global Biodiversity Information Facility (GBIF)

Usage

```
data(gbifdat)
```

Format

A data frame with 311 observations on the following 16 variables.

`gbifid` a numeric vector - gbif identifier

`family` a character vector - family of the taxon

`genus` a character vector - genus name

`species` a character vector - species name

`infraspecificepithet` a character vector - infraspecific epithet

`taxonrank` a character vector - taxon rank

`scientificname` a character vector - scientific name

`countrycode` a character vector - country code

`decimallatitude` a numeric vector - latitude

`decimallongitude` a numeric vector - longitude

`elevation` a numeric vector - elevation in meters

`elevationaccuracy` a numeric vector - elevation accuracy

`depth` a numeric vector - depth

`depthaccuracy` a numeric vector - depth accuracy

`taxonkey` a numeric vector - taxon key

`specieskey` a numeric vector - species key

Source

www.gbif.org

Examples

```
data(gbifdat)
```

geo2envid	<i>Interactive plot to explore points in geographical and environmental space</i>
-----------	---

Description

An interactive plot with options to select to explore points in the geographical or environmental space. The environmental space is defined by the values of two environmental variables.

Usage

```
geo2envid(edat, g1, group1 = "Species", group2 = "",
world, xc = "AP", yc = "AMT", xname = "Annual Precipitation (mm)",
yname = "Mean Annual Temperature", showrecord = "",
ext = c(-180, 180, -60, 90))
```

Arguments

edat	a dataframe containing fields with the following names: ID, x, y, Species, x_original, y_original, Correction, Modified, Exclude and the values of selected environmental variables
g1	the name of the species to select
group1	this is usually the column of species names
group2	a second grouping variable
world	a shapefile of the world, where the column containing the country names must be "NAMES"; see data(wrld_simpl)
xc	the name of the environmental variable to be used on the x-axis of the environmental space
yc	the name of the environmental variable to be used on the y-axis of the environmental space
xname	x-axis label for the environmental space
yname	y-axis label for the environmental space
showrecord	the ID of a selected record to be shown on the map
ext	The extent, which can be specified as c(xmin, xmax, ymin, ymax) default extent c(-180, 180, -90, 90). Alternatively if ext="p" then the extent will be calculated from the coordinates of the points in the dataset.

Details

The selected records are marked with a red dot and ID numbers are shown. Records that are considered to be outliers can be excluded by selecting the record in the environmental space. A menu with various options is produced.

Value

Interactive plot

Note

plotsetup should be run first

Author(s)

Mark Robertson and Vernon Visser

See Also

plotsetup, geo2envpca, alternatives, alternativesenv, wclim

Examples

```
## Not run:
plotsetup(6,6)
ed<-geo2envid(edat,"Species U","Species","",world,xc="bio12",
yc="bio1",xname="Ann. Precip.",yname="Ann. Mean Temp.",
showrecord="",ext="p")
## End(Not run)
```

geo2envpca

Interactive plot to explore points in geographical and environmental space

Description

An interactive plot with options to select to explore points in the geographical or environmental space. The environmental space is defined by the values of two principal components from a principal components analysis on several environmental variables.

Usage

```
geo2envpca(edat, g1, group1 = "Species", group2 = "",
world, scaling = 1, vars = c("AMT", "AP", "MTCM", "MTWM",
"PWQ", "PCQ"), showrecord = "", ext = c(-180, 180, -60, 90))
```

Arguments

edat	a dataframe containing fields with the following names: ID, x, y, Species, x_original, y_original, Correction, Modified, Exclude and the values of selected environmental variables
g1	the name of the species to select
group1	this is usually the column of species names

group2	a second grouping variable
world	a shapefile of the world, where the column containing the country names must be "NAMES"; see data(wrld_simpl)
scaling	a value of 1 or 2 for the type of scaling of the PCA space
vars	a character vector of environmental variable names that will be used to define the environmental space. Values of these variables must be present in edat.
showrecord	the ID of a selected record to be shown on the map
ext	The extent, which can be specified as c(xmin, xmax, ymin, ymax) default extent c(-180, 180, -90, 90). Alternatively if ext="p" then the extent will be calculated from the coordinates of the points in the dataset.

Details

The selected records are marked with a red dot and ID numbers are shown. Records that are considered to be outliers can be excluded by selecting the record in the environmental space. A menu with various options is produced.

Value

Interactive plot

Note

plotsetup should be run first

Author(s)

Mark Robertson and Vernon Visser

See Also

plotsetup, geo2envid, alternatives, alternativesenv, wclim

Examples

```
## Not run:
plotsetup(6,6)
ed<-geo2envpca(edat,"Species U",group1="Species",group2="",
world,scaling=1,vars=c("bio1","bio12","bio5","bio6","bio14"),
showrecord="1981",ext="p")
## End(Not run)
```

`getextent`*Get the extent of an object*

Description

Calculates the extent (x- and y-limits in decimal degrees)

Usage

```
getextent(x, y, ext)
```

Arguments

<code>x</code>	x-coordinates in decimal degrees
<code>y</code>	y-coordinates in decimal degrees
<code>ext</code>	<code>ext</code> can be a raster from the raster package; an extent object from the raster package; "p" which means that the extent will be derived from the x- and y-coordinates in the input.

Value

<code>x1m</code>	x-limits
<code>y1m</code>	y-limits
<code>beyond</code>	a vector with ones where points are beyond extent and zeros where they are within the extent

Author(s)

Mark Robertson

See Also

`pointsworld`

Examples

```
data(dat)
e<-getextent(dat$x,dat$y,"p")
```

getformat	<i>Get formats of coordinates</i>
-----------	-----------------------------------

Description

Finds the formats for a character vector of coordinates

Usage

```
getformat(x)
```

Arguments

x a character vector of coordinates

Value

a format string in which 0 is returned if numeric, "." if decimal, "L" if N,S,E,W; or "*" for any other character

Author(s)

Mark Robertson

See Also

dmsabs, dmsparse

Examples

```
getformat("22.15.32S")
```

getletter	<i>Find letters in coordinates</i>
-----------	------------------------------------

Description

Finds letters (N, S, W, E) in a character vector or coordinates

Usage

```
getletter(x)
```

Arguments

x a character vector of coordinates

Value

a dataframe with the coordinate and letter (N, S, E, W) in separate columns

Note

This function is used in dmsparse

Author(s)

Mark Robertson

See Also

dmsparse

Examples

```
x<-c("25 30 15S", "34 45 12E")
getletter(x)
```

keepmainfields

Keep main fields in a dataframe.

Description

Select main fields required by biogeo.

Usage

```
keepmainfields(dat, ID = "", Species = "", x = "", y = "", others = "")
```

Arguments

dat	A dataframe containing records including an identifier field, a species name field, x- and y-coordinates fields and any other fields.
ID	The unique identifier field
Species	The field containing the species names
x	x-coordinate in decimal degrees
y	y-coordinate in decimal degrees
others	Names of other fields that should be retained in the dataframe

Value

A dataframe containing the selected fields.

Author(s)

Vernon Visser and Mark Robertson

See Also

addmainfields, renamefields, checkdatastr

Examples

```
names(gbifdat)
dat3 <- keepmainfields(gbifdat, Species='species', x='decimallongitude', y='decimallatitude')
```

missingcoords

Finds indices of records in a dataset where the coordinates are missing

Description

Finds indices of records in a dataset where the coordinates are missing

Usage

```
missingcoords(x, y)
```

Arguments

x a column of x-coordinates
y a column of y-coordinates

Value

index corresponding with row numbers in the input data frame

Author(s)

Mark Robertson

See Also

dmsparse

Examples

```
a<-missingcoords(dat$x, dat$y)
```

missingvalsexclude *Excludes records with missing values for points*

Description

Excludes records with missing values for points (i.e. points falling in the sea for terrestrial species)

Usage

```
missingvalsexclude(rst, dat)
```

Arguments

rst	a raster
dat	a dataframe containing fields with the following names: ID, x, y, Species and Exclude

Value

a dataframe in which records with points that have missing values extracted from the raster input (rst) have values in the column Exclude are set to 1.

Author(s)

Mark Robertson and Vernon Visser

See Also

duplicatesexclude, addmainfields, renamefields, keepmainfields, checkdatastr

Examples

```
## Not run:  
dem<-raster(dem,xmn=-180, xmx=180, ymn=-60, ymx=90)  
missingvalsexclude(dem, dat)  
## End(Not run)
```

modified	<i>Identify records that were modified between two dates</i>
----------	--

Description

Identifies records that were modified between two dates

Usage

```
modified(dat, d1, d2)
```

Arguments

dat	a dataframe containing fields with the following names: ID, x, y, Species, x_original, y_original, Correction, Modified, Exclude
d1	the start date e.g. "01-07-2014 12:11:12"
d2	the end date e.g. "05-07-2014 12:11:12"

Value

index of row numbers for records with a date in the Modified column of the dataframe (dat) that falls between the dates specified in d1 and d2

Author(s)

Mark Robertson

See Also

modifiedtoday

Examples

```
modified(dat, "01-07-2014 12:11:12", "05-07-2014 12:11:12")
```

modifiedtoday	<i>Identify records that were modified today</i>
---------------	--

Description

Identifies records that were modified on the current date

Usage

```
modifiedtoday(dat)
```

Arguments

`dat` a dataframe containing fields with the following names: ID, x, y, Species, x_original, y_original, Correction, Modified, Exclude

Value

index of row numbers for records with a date in the Modified column of the dataframe (`dat`) that correspond with today's date

Author(s)

Mark Robertson

See Also

`modified`, `alternatives`, `alternatives2`

Examples

```
f<-modifiedtoday(dat)
dat[f,]
```

`msk60`

msk60

Description

A vector of index numbers indicating the grid cells with environmental data in 1 degree (60 minute) dataset

Usage

```
data("msk60")
```

Format

The format is: atomic [1:16216] 2304 2305 2306 2307 2308 ... - attr(*, "na.action")=Class 'exclude'
int [1:37784] 1 2 3 4 5 6 7 8 9 10 ...

Details

Generated by reclassifying the Worldclim 10 minute data to 30 minute spatial resolution

Source

www.worldclim.org

Examples

```
data(msk60)
```

nearestcell	<i>Assigns points that fall into the sea to the centre of the nearest adjacent coastal grid cell for terrestrial species and to nearest sea cell for marine species</i>
-------------	---

Description

Assigns points that fall in the sea to the centre nearest adjacent coastal grid cell. It ignores points that do not have an adjacent coastal grid cell.

Usage

```
nearestcell(dat, rst)
```

Arguments

dat	a dataframe containing fields with the following names: ID, x, y, Species, x_original, y_original, Correction, Modified, Reason, Exclude
rst	a raster

Value

dat	a dataframe in which the new coordinates are assigned to x and y. The original values for the x- and y-coordinates are assigned to x_original and y_original.
moved	a dataframe with the identifiers (ID), x- and y-coordinates (x and y) for the coordinates that were modified

Author(s)

Mark Robertson and Veron Visser

See Also

missingvaluesexclude, addmainfields

Examples

```
## Not run:
dem<-raster(dem,xmn=-180, xmx=180, ymn=-60, ymx=90)
a<-nearestcell(dat, dem)
## End(Not run)
```

`outliers`*Detects outliers using values extracted from an environmental variable*

Description

Calculates the outliers using the reverse jackknife procedure (see `rjack`) and boxplot statistics (using `boxplot.stats`). If the value lies 1.5 times beyond the length of the box in the boxplot then it is considered to be an outlier. This function is used by `errorcheck`.

Usage

```
outliers(rid, species, dups, ev)
```

Arguments

<code>rid</code>	row identifier created in <code>errorcheck</code>
<code>species</code>	column of species names
<code>dups</code>	a column of zeros and ones, where ones indicate duplicates
<code>ev</code>	the values of the environmental variable

Value

values of 1 if records is an outlier and zero if not in two columns. The first column is for the boxplot method and the second is for the reverse jackknife method.

Note

This function is only applied to a species if there are 10 or more records for that species.

Author(s)

Mark Robertson

See Also

`errorcheck`, `boxplot.stats`, `rjack`

Examples

```
rid<-1:20
species<-rep("Species A",20)
dups=rep(0,20)
ev<-c(rnorm(19,mean=20,sd=1),40)
a<-outliers(rid, species, dups, ev)
```

`parsecoords`*Parse coordinates into separate fields*

Description

Parse coordinates into separate fields from a character vector using a format string. This is a manual version of `dmsparse`.

Usage

```
parsecoords(dat, d1, fmtstr)
```

Arguments

<code>dat</code>	a character vector of coordinates
<code>d1</code>	a summary of the various coordinate formats obtained from <code>uniqueformats</code>
<code>fmtstr</code>	a format string for the different formats present e.g. "dd.mm", "dd.mm.ss"

Value

a dataframe containing degrees, minutes, seconds, letters (N,S,E,W) and decimal degrees in separate columns

Note

In most cases `dmsparse` would be preferable

Author(s)

Mark Robertson

See Also

`dmsparse`, `uniqueformats`

Examples

```
fmtstr<-c("dd mm.m", "dd mm ss.ss", "dd mm ss")
uf<-uniqueformats(places$long[1:3])
px<-parsecoords(places$long[1:3],uf,fmtstr)
```

places *A dataset of localities*

Description

A dataset of localities with coordinates in different formats.

Usage

```
data(places)
```

Format

A data frame with 20 observations on the following 4 variables.

id a numeric vector - a unique identifier

Place a character vector - the name of the place

long a character vector - longitude in various formats

lat a character vector - latitude in various formats

Examples

```
head(places)
```

plotsetup *Produces devices for plotting*

Description

Plot windows are used by geo2envid and geo2envpca

Usage

```
plotsetup(xi, yi)
```

Arguments

xi dimension of the x-axis of the plot window in inches

yi dimension of the y-axis of the plot window in inches

Value

produces three plot windows

Author(s)

Mark Robertson

See Also

geo2envid2, geo2envpca

Examples

```
plotsetup(6,6)
```

points2shape	<i>Save data to a points shape file</i>
--------------	---

Description

Save data to a points shape file with a geographic projection

Usage

```
points2shape(dat, x, y, fn)
```

Arguments

dat	a dataframe containing x- and y-coordinates and any other data
x	name of x-coordinates column in dat
y	name of y-coordinates column in dat
fn	the name of the file to be create (should have a .shp extension)

Value

a point shape file

Author(s)

Mark Robertson

See Also

pointsworld

Examples

```
points2shape(dat, "x", "y", fn="pointshape.shp")
```

pointsworld

Plots point records on a world map using their latitude and longitude

Description

Plots points on a world map. Records that fall outside of country boundaries appear in red and records inside country boundaries appear in blue.

Usage

```
pointsworld(world, dat, x, y, ext = c(-180, 180, -90, 90))
```

Arguments

world	a shapefile of the world, where the column containing the country names must be "NAMES"; see data(wrld_simpl)
dat	a dataframe containing the x- and y-coordinates in decimal degrees and a unique identifier for each record called ID
x	the name of the x-coordinate column in dat
y	the name of the y-coordinate column in dat
ext	The extent, which can be specified as c(xmin, xmax, ymin, ymax) default extent c(-180, 180, -90, 90). Alternatively if ext="p" then the extent will be calculated from the coordinates of the points in the dataset.

Value

a map of the world showing the points

Author(s)

Mark Robertson

See Also

geo2envid, alternatives, alternatives2, alternativesenv, errorcheck

Examples

```
## Not run:
dev.new(width=7,height=7)
a<-pointsworld(world, dat, x="x", y="y", ext = c(-180, 180, -90, 90))
a<-pointsworld(world, dat, x="x", y="y", ext = "p")
a<-pointsworld(world, dat, x="x", y="y", ext = c(10, 40, -35, -20))

## End(Not run)
```

precisioncheck	<i>Check the precision of the coordinates</i>
----------------	---

Description

Checks the precision of the coordinates to determine whether the coordinate fall exactly at the centre or exactly on top left corner of a grid cell of a particular spatial resolution e.g. 30 minute.

Usage

```
precisioncheck(dat, x = "x", y = "y", s, e)
```

Arguments

dat	a dataframe containing fields with the following names: ID, x, y, Species, x_original, y_original, Correction, Modified, Exclude
x	name of the x-coordinate in decimal degrees
y	name of the y-coordinate in decimal degrees
s	start spatial resolution in minutes e.g. 10 min
e	end spatial resolution in minutes e.g. 30 min

Details

Grid cells of sizes corresponding to the start spatial resolution an increasing by 5 minutes up to the end spatial resolution will be considered.

Value

returns the contents of dat and includes columns for each spatial resolution e.g. p10m for 10 minute spatial resolution. These columns contain values of 1 if the record is considered to have a low precision and zero otherwise. A column called preci contains values of 1 if any of the other columns tested have a value of one.

Note

In most cases the values should be set as s=10 and e=30

Author(s)

Mark Robertson

See Also

errorcheck

Examples

```
precisioncheck(dat, x = "x", y = "y", s=10, e=30)
```

precisionenv

Check precision of records

Description

Determines whether any records have a lower precision than that of the selected raster file

Usage

```
precisionenv(dat, rst, x = "x", y = "y")
```

Arguments

dat	A dataframe containing the required biogeo fields (see checkdatastr)
rst	A raster
x	x-coordinate in decimal degrees
y	y-coordinate in decimal degrees

Value

A dataframe containing the field envpreci, with ones when the point records have a lower precision than the raster and zero otherwise.

Author(s)

Vernon Visser and Mark Robertson

See Also

precisioncheck, errorcheck, quickclean

Examples

```
## Not run:
dem<-raster(dem,xmn=-180, xmx=180, ymn=-60, ymx=90)
datpce <- precisionenv(dat, dem, x='x', y='y')
datpce[datpce$envpreci==1,] #View records with possible precision problems

## End(Not run)
```

 quickclean

Automated data cleaning

Description

Automated data cleaning. Performs a country mismatch check if the country field is specified, it performs a check to determine if the records are at the appropriate precision for the spatial resolution, it assigns point records to the nearest cell containing environmental data (using `nearestcell`) and removes records that are in the wrong environment. It flags duplicate records per species per grid cell.

Usage

```
quickclean(world,dat, ID = "ID", Species = "Species", x = "x",
y = "y", countries = "", others = "", res, msk, ext)
```

Arguments

<code>world</code>	a shapefile of the world, where the column containing the country names must be "NAMES"; see <code>data(wrld_simpl)</code>
<code>dat</code>	A dataframe containing the required biogeo fields (see <code>checkdatastr</code>)
<code>ID</code>	The unique identifier field
<code>Species</code>	The field containing the species names
<code>x</code>	x-coordinate in decimal degrees
<code>y</code>	y-coordinate in decimal degrees
<code>countries</code>	A field containing country names
<code>others</code>	Names of other fields that should be retained in the dataframe.
<code>res</code>	Spatial resolution for the richness map specified in minutes
<code>msk</code>	A mask index of the same spatial resolution as <code>res</code>
<code>ext</code>	The extent for the map. This can be <code>ext="p"</code> to use the point data to calculate the extent. It could be an <code>Extent</code> object from the Raster package or a vector containing the following: <code>minx, maxx, miny, maxy</code> .

Value

Returns a dataframe containing the identifiers (`ID`), species names (`Species`), x-coordinate (`x`), y-coordinate (`y`), a unique cell index (`indx`), and duplicates (`dups`). All records containing errors in the input dataframe are removed.

Author(s)

Mark Robertson

See Also

errorcheck, nearestcell, elevcheck, quickrich

Examples

```
dat2<-quickclean(world,dat,ID='ID',Species='Species',x='x',y='y',
countries = '',others='',res=60,msk=msk60,ext="")
```

quickrich

Perform error checks and produce richness map

Description

Performs error checks on the data (using quickclean) and removes records that contain errors before producing a species richness map

Usage

```
quickrich(dat,world, ID = "ID", Species = "Species", x = "x",
y = "y", countries = "", others = "", res, msk, ext)
```

Arguments

dat	A dataframe containing the required biogeo fields (see checkdatastr)
ID	A field of unique identifiers
world	a shapefile of the world, where the column containing the country names must be "NAMES"; see data(wrld_simpl)
Species	A field of species names
x	x-coordinate in decimal degrees
y	y-coordinate in decimal degrees
countries	A field containing country names (optional). If the countries field is specified then a check for mismatches is done between the countries in the country field and those extracted from a world map. It is advisable not to specify this parameter for marine species.
others	The names of other fields to be included
res	Spatial resolution of the richness map (in minutes)
msk	A mask index of the same spatial resolution as res
ext	The extent for the map. This can be ext="p" to use the point data to calculate the extent. It could be an Exent object from the Raster package or a vector containing the following: minx, maxx, miny, maxy.

Value

A raster of species richness

Author(s)

Mark Robertson

See Also

richness, richnessmap, quickclean

Examples

```
ex1 <- c(15,35,-36,-23) # set the extent
rich<-quickrich(dat,world,ID='ID',Species='Species',x='x',y='y',
countries = "",others='',res=60,msk=msk60,ext=ex1)
```

renamefields

Rename particular fields in a dataframe

Description

Several functions require a particular set of columns in the dataframe with specific names. If the columns in the dataframe do not have these names then they can be renamed. The fields that can be renamed include: ID (unique identifiers), x (x-coordinate), y (y-coordinate), Species (the species names column)

Usage

```
renamefields(dat, ID = "ID", x = "x", y = "y", Species = "Species")
```

Arguments

dat	a dataframe containing the point records dataset
ID	the identifiers column
x	the x-coordinates (longitude) in decimal degrees
y	the y-coordinates (latitude) in decimal degrees
Species	the species names

Value

a dataframe in which the selected fields have been renamed

Author(s)

Mark Robertson

See Also

checkdatastr, addmainfields

Examples

```
dat<-data.frame(places,Speciesnames="")
a<-renamefields(dat, ID = "id", x = "long", y = "lat", Species = "Speciesnames")
```

richness

*Produced a species richness map***Description**

Produce a species richness map from point records, without the use of a raster.

Usage

```
richness(dat, res = 10, option = "richness", buf = 5, ext = "")
```

Arguments

dat	A dataframe containing the required biogeo fields (see checkdatastr)
res	Spatial resolution for the richness map specified in minutes
option	Species richness or number of records per grid cell
buf	A buffer (in minutes) around the outer points to define the extent
ext	The extent for the map. This can be ext="p" to use the point data to calculate the extent. It could be an Exent object from the Raster package or a vector containing the following: minx, maxx, miny, maxy.

Value

A raster containing either species richness per grid cell or number of records per grid cell.

Author(s)

Mark Robertson

See Also

quickrich, richnessmap

Examples

```
ex1 <- c(15,35,-36,-23) # specify the extent
rich<-richness(dat,res=20,option="richness",buf=5,ext=ex1)
```

`richnessmap`*Map of the number of species or number of records per grid cell*

Description

Creates a raster map of the number of species or number of records per grid cell

Usage

```
richnessmap(dat, rst, option = "richness")
```

Arguments

<code>dat</code>	a dataframe containing fields with the following names: ID, x, y, Species, x_original, y_original, Correction, Modified, Exclude
<code>rst</code>	a raster object
<code>option</code>	"richness" gives species richness and "records" gives total number of records per grid cell

Details

grid cells are based on rst

Value

a raster object containing species richness or numbers of records per grid cell

Author(s)

Mark Robertson

See Also

`pointsworld`

Examples

```
## Not run:  
dem<-raster(dem,xmn=-180, xmx=180, ymn=-60, ymx=90)  
rich<-richnessmap(dat,dem,option="richness")  
nrec<-richnessmap(dat,dem,option="records")  
  
## End(Not run)
```

`rjack`*Outlier detection using the Reverse Jackknife*

Description

Implements the Reverse Jackknife procedure as described by Chapman (2005). Used in outliers.

Usage

```
rjack(d)
```

Arguments

`d` values of an environmental variable extracted from points

Details

This function is based on the Reverse Jackknife method described by Chapman (2005)

Value

indices of values that are outliers

Note

The implementation was based on Chapman (2005) and not the more conservative implementation in DivaGIS

Author(s)

Mark Robertson

References

Chapman, A.D. (2005) Principles and Methods of Data Cleaning - Primary Species and Species-Occurrence Data, version 1.0. Report for the Global Biodiversity Information Facility, Copenhagen.

See Also

outliers, errorcheck

Examples

```
x<-c((rnorm(20,mean=20,sd=2)),40)
a<-rjack(x)
```

sep	<i>Separate coordinates into degrees, minutes and seconds</i>
-----	---

Description

Separates coordinates into degrees, minutes and seconds (used in dmsparse)

Usage

```
sep(dx)
```

Arguments

dx coordinates in a character vector

Value

a dataframe containing degrees, minutes and seconds in separate columns

Author(s)

Mark Robertson

Examples

```
x<-data.frame(c="23_14_15.2",L="S")
sep(x)
```

speciescount	<i>Count number of records per species</i>
--------------	--

Description

Counts number of records per species in a dataframe

Usage

```
speciescount(dat, orderby = "Species")
```

Arguments

dat a dataframe containing fields with the following names: ID, x, y, Species and Exclude

orderby the output can be ordered by Species, ntot or nuniq

Value

a dataframe including the species names, total number of records for that species (ntot) and number of records for that species with duplicates per grid cell removed (nuniq)

Author(s)

Mark Robertson

See Also

duplicatesexclude, missingvaluesexclude, addmainfields, renamefields, checkdatastr

Examples

```
speciescount(dat, orderby = "Species")
speciescount(dat, orderby = "ntot")
```

substdmm

Swap degrees and minutes in a coordinate

Description

Converts the decimal degrees coordinate to degrees, minutes and seconds then swaps the degrees and minutes and recalculates the decimal degrees

Usage

```
substdmm(dc)
```

Arguments

dc decimal degrees data

Details

used in alternatives, alternatives2 and alternativesenv

Value

a numeric vector of decimal degrees

Author(s)

Mark Robertson

See Also

alternatives, alternatives2, alternativesenv

Examples

```
x<- -25.5432
a<-substdmm(x)
```

uniqueformats	<i>List unique coordinate formats</i>
---------------	---------------------------------------

Description

Lists the unique formats of the coordinates in a character vector

Usage

```
uniqueformats(x2)
```

Arguments

x2 a character vector of coordinates

Value

unique coordinate formats as generated for getformat

Author(s)

Mark Robertson

See Also

dmsabs

Examples

```
x<-c("123_23.345W", "23d15m32.0S", 34.3456, 45.5432)
uniqueformats(x)
```

wclim *Returns Worldclim bioclimatic variable names*

Description

Returns Worldclim bioclimatic variable names. See www.worldclim.org

Usage

```
wclim(f = 1:19, full = F)
```

Arguments

f	the variable numbers for which names are required
full	if full is TRUE then all columns are returned including an abbreviation for the variable and a correction factor

Details

Some of the temperature variables from Worldclim were multiplied by 10 for storage reasons. The correction is the value that the bioclimatic variable should be divided by to get it back into degrees centigrade.

Value

a dataframe including the bio name e.g. BIO05, an abbreviation of the name, the name of the variable and a correction factor.

Author(s)

Mark Robertson

References

Hijmans, R.J., S.E. Cameron, J.L. Parra, P.G. Jones and A. Jarvis, 2005. Very high resolution interpolated climate surfaces for global land areas. *International Journal of Climatology* 25: 1965-1978.

See Also

`env2stack`

Examples

```
wclim()  
wclim(1:19, full=TRUE)  
wclim(c(1,5,6,12))
```

world	<i>World country polygons</i>
-------	-------------------------------

Description

The dataset comes from the maptools package from wrld_simpl. See data(wrld_simpl) in maptools.

Usage

```
data(world)
```

Format

see details in maptools

Source

maptools package http://mappinghacks.com/data/TM_WORLD_BORDERS_SIMPL-0.2.zip

Examples

```
data(world)
```

Index

* datasets

- dat, 10
- datm, 11
- dem, 14
- edat, 19
- gbifdat, 26
- msk60, 36
- places, 40
- world, 55

* package

- biogeo-package, 3

- addmainfields, 3
- alternatives, 4
- alternatives2, 6
- alternativesenv, 7

- biogeo (biogeo-package), 3
- biogeo-package, 3

- checkdatastr, 9
- coord2numeric, 10

- dat, 10
- datm, 11
- dd2dmslat, 12
- dd2dmslong, 13
- dem, 14
- dms2dd, 15
- dmsabs, 16
- dmsparse, 16
- dmsparsefmt, 17
- duplicatesexclude, 18

- edat, 19
- elevcheck, 20
- env2stack, 21
- errorcheck, 22

- fieldsmissing, 23
- finddecimals, 24

- fmtcheck, 25

- gbifdat, 26
- geo2envid, 27
- geo2envpca, 28
- getextent, 30
- getformat, 31
- getletter, 31

- keepmainfields, 32

- missingcoords, 33
- missingvalsexclude, 34
- modified, 35
- modifiedtoday, 35
- msk60, 36

- nearestcell, 37

- outliers, 38

- parsecoords, 39
- places, 40
- plotsetup, 40
- points2shape, 41
- pointsworld, 42
- precisioncheck, 43
- precisionenv, 44

- quickclean, 45
- quickrich, 46

- renamefields, 47
- richness, 48
- richnessmap, 49
- rjack, 50

- sep, 51
- speciescount, 51
- substdmm, 52

- uniqueformats, 53

wclim, [54](#)
world, [55](#)