Package 'bbsRDM'

May 19, 2019

Type Package

Title Calculate regime detection metrics using bird communities

Version 0.1.0

Author Jessica Burnett

Maintainer Jessica Burnett <jessicaleighburnett@Gmail.com>

Description Calculates regime detection metrics using North American Breeding Bird Survey data across North-South and East-West spatial transects. License: MIT +

Encoding UTF-8

LazyData true

Depends dplyr,
readr,
stringr,
ggplot2,
sp,
raster,
here,
feather,
kedd,
devtools,
caTools,
glue

Imports

RoxygenNote 6.1.1

Suggests knitr,
rmarkdown

VignetteBuilder knitr

R topics documented:

bbsRDM 2
birdsToFeathers 2
calculateMetrics 3
createSamplingGrid 3
funcMass 4
getDataBBS 4

getMilBases 5
GetRegions 5
getRouteInfo 6
GetSpNames 7
GetUnzip 7
hello 8
importDataBBS 8
importResults 9
loadBirdFeathers 9
mergeFunMassBBS 10
mungeSubsetData 10
myTheme 10
saveMyResults 11
sort.year.line 11
subsetByAOU 11

Index **12**

`bbsRDM` *bbsRDM: A package for calculating various 'regime detection' measures.*

Description

The bbsRDM package provides source code for calculating multiple methods that are used in the ecological regime shift literatures. foo, bar and baz.

`birdsToFeathers` *Save BBS dataframe as a feather file*

Description

Save BBS dataframe as a feather file

Usage

```
birdsToFeathers(dataIn, newDir, filename)
```

Arguments

`dataIn`	The BBS data to save.
`newDir`	Where to save the BBS feathers.
`filename`	Name of the new filename (e.g., 'arizona.zip'). This function will replace .zip with .feather

calculateMetrics *Calculate regime detection metrics*

Description

Calculates regime detection metrics across space or time. Calculates distance travelled, Fisher Information, Variance Index, Coefficient of Variation, mean, standard deviation, variance,skewness, and kurtosis. #' @param dataIn A data frame containing columns c(variable, time, value).

Usage

```
calculateMetrics(dataIn, metrics.to.calc = c("distances", "ews"),
  min.samp.sites = 8, direction, yearInd, to.calc = c("EWS", "FI",
  "VI"))
```

Arguments

dataIn	data frame with columns: sortVar (the sorting variable; latitude or longitude), cellID (cell ID for the spatial grid), variable (species), value (count data).
metrics.to.calc	One or more of c("distances", "ews")
min.samp.sites	Minimum number of unique sites in the transect (or unique times along the time series) required to analyze the data. Most metrics can be calculated using three data points, although we do not nrecommend this.
direction	Direction of the analysis (South-North or East-West)

createSamplingGrid *Generate a sampling grid (rectangular) for regions in North America.*

Description

Creates a sampling grid across the continental united states and assign BBS routes to specific a row and column ID.

Usage

```
routes_gridList <- createSamplingGrid(cs = c(1,1))
```

Arguments

cs	Cell size (in degree lat, long). Default is 0.5 degree long by 0.5 degree lat. In this region, 1 deg latitude ~= 69 miles & 1 deg longitude ~= 55 miles. The total length of a BBS route is ~50 miles. Caution when using degrees < 1 by 1 degree as a single route could fal into multiple cells...
bbLat	Min and max (in any order) latitude coordinates for the bounding box. The function removes routes (lat,long) falling outside these coordinates. Default = c(23, 51). See also 'bbLong'.

bbLong	Min and max (in any order) longitude for the bounding box. The function removes routes (lat,long) falling outside these coordinates. Default = c(23, 51). See also 'bbLat'.
country	One or more of c("CA","USA"). If not specified, will keep grid based on both CA and USA.

---

funcMass *Load functional trait and mass data*

Description

Load functional trait and mass data

Usage

```
funcMass(dataWD = paste0(getwd(), "/data"), fxn = T, mass = F)
```

Arguments

dataWD	Where the functional trait and mass dataframes are stored.
fxn	Logical. Retrieves functional trait data (referece).
mass	Logical. Retrieves body mass information (Dunning reference).

---

getDataBBS *Download USGS Breeding Bird Survey data*

Description

This function was adapted from **oharar/rBBS** package.

Usage

```
getDataBBS(file,
  dir = "ftp://ftpext.usgs.gov/pub/er/md/laurel/BBS/DataFiles/States/",
  year = NULL, aou = NULL, countrynum = NULL, states = NULL)
```

Arguments

file	One file name including the .zip extension ("stateX.zip"). Preferably download a single state at a time, otherwise run time will take >1 minutes.
dir	URL to the StatesFiles.
year	Vector of years. Default = NULL (all years).
aou	Vector of AOU #s Default = NULL (all species).
countrynum	Vector of country ID #'s. Default = NULL (all countryNums).
states	Vector of state names Default = NULL (all states).

Value

If download successful, a dataframe with the results.

Examples

```
# download all species and years from Nebraska.

## Not run:
NE <- getDataBBS(file = "Nebrask.zip")

## End(Not run)
```

`getMilBases` *Get military installation shapefile from online data repo and save to*

Description

Get military installation shapefile from online data repo and save to

Usage

```
getMilBases(shploc = "http://www.acq.osd.mil/eie/Downloads/DISDI/installations_ranges.zip",
 shpfile = "MIRTA_Points")
```

Arguments

`shploc`	URL location for.zip file
`shpfile`	Name of the file to upload

Value

shp A shapefile with points designating U.S. military bases.

`GetRegions` *Get BBS region names for download route data.*

Description

Read in list of regions (State/Prov/TerrName), from RegionCodes.txt, and then extract list of where the 10-stop data is kept

Usage

```
GetRegions(Dir = "ftp://ftpext.usgs.gov/pub/er/md/laurel/BBS/DataFiles/",
ZipFiles = TRUE, bbsDir = NULL)
```

Arguments

Dir	location of the BBS files. Do not change unless they make major changes.
bbsDir	Location of the folder containing bbs raw data (defined in runthrough.rmd)

---

getRouteInfo *Download route information from USGS server*

Description

This function was adapted from **oharar/rBBS** package.

Usage

```
getRouteInfo(routesFile = "routes.zip",
  routesDir = "ftp://ftpext.usgs.gov/pub/er/md/laurel/BBS/DataFiles/",
  RouteTypeID = 1, Stratum = NULL, BCR = NULL)
```

Arguments

RouteTypeID	One or more numbers indicating route substrate (1=roadside;2=water;3=off-road; Default = 1, roadside only).
Stratum	A vector of BBS physiographic stratum codes by which to filter the routes.
BCR	A vector of Bird Conservation Region codes where by which to filter the routes.
routeDir	Location of the routes.zip folder Should be in DatFiles folder (default).
routeFile	Name of the route information file. Usually "routes.zip".

Value

If download successful, a dataframe with the results.

Examples

```
# download BBS route data.

## Not run:
RouteInfo <- getRouteInfo()

## End(Not run)
```

GetSpNames *Download species names*

Description

Read in list of species names, from SpeciesList.txt, and then extract list of where the data is kept

Usage

```
GetSpNames(Dir = "ftp://ftpext.usgs.gov/pub/er/md/laurel/BBS/DataFiles/")
```

Arguments

Dir	ftp URL for directory with data files

Value

A dataframe

GetUnzip *Downloads and unzips a zip archive*

Description

Downloads and unzips a zip archive

Usage

```
GetUnzip(ZipName, FileName)
```

Arguments

ZipName	file to download
FileName	file to unzip to

Details

Used internally. If ZipName begins with 'http' or 'ftp', then download and unzip to Filename and return as a dataframe. Otherwise, unzip Zipname and return as a data.frame.

Value

A dataframe

hello *Hello, World!*

Description

Prints 'Hello, world!'.

Usage

```
hello()
```

Examples

```
hello()
```

importDataBBS *Download and merge all the count, route, and species information*

Description

This function was adapted from **oharar/rBBS** package.

Usage

```
importDataBBS(file, dir, year = NULL, aou = NULL, countrynum = NULL,
  states = NULL, routesFile = "routes.zip",
  routesDir = "ftp://ftpext.usgs.gov/pub/er/md/laurel/BBS/DataFiles/",
  RouteTypeID = 1, Stratum = NULL, BCR = NULL)
```

Arguments

file	The name of the zipfile to be downloaded from dir
dir	http:// pr ftp:/ link to bbs data location (here: state files)
year	Vector of years. Default = NULL (all years).
aou	Vector of AOU #s Default = NULL (all species).
countrynum	Vector of country ID #'s. Default = NULL (all countryNums).
states	Vector of state names Default = NULL (all states).
RouteTypeID	One or more numbers indicating route substrate (1=roadside;2=water;3=off-road; Default = 1, roadside only).
Stratum	A vector of BBS physiographic stratum codes by which to filter the routes.
BCR	A vector of Bird Conservation Region codes where by which to filter the routes.
zipFileNames	One or more file names ("state.zip"), char vector. Preferably download a single state at a time, otherwise run time will take >1 minutes.
countryName	Vector of country names. Default = NULL (all countryNames).
regions	Vector of regionCodes. Default = NULL (all regions).
routeDir	Location of the routes.zip folder Should be in DatFiles folder (default).
routeFile	Name of the route information file. Usually "routes.zip".

Value

A dataframe

`importResults` *Load the regime detection metric results (.feathers)*

Description

Load the regime detection metric results (.feathers)

Usage

```
importResults(resultsDir, myPattern, subset.by = NULL,
  metrics.keep = NULL)
```

Arguments

`resultsDir`	Where the results are stored.
`myPattern`	Pattern for loading results files. Name of the subdirectory ("distances", "ews").
`subset.by`	One or more patterns by which to filter file names for import the data. Can be used to import South-North transects only (subset.by = "South-North"), or to import all within a single year (subset.by == "year1987"). Default = NULL will import all files in the directory. Multiple example = c("1979","South-North")
`metrics.keep`	If specified will keep only the metrics specified.

Details

Used after running calculate_distanceTravelled()? to make results available for visualization?

Value

A dataframe

`loadBirdFeathers` *Load the BBS data feathers into R.*

Description

Load the BBS data feathers into R.

Usage

```
loadBirdFeathers(newDir, filename)
```

Arguments

`newDir`	Where the BBS feathers are saved.
`filename`	Name of the feather filename (e.g., 'arizona.zip' or 'arizona'). This function will replace .zip with .feather when necessary.

mergeFunMassBBS *Merge functional group and mass data for species with BBS counts*

Description

Merge functional group and mass data for species with BBS counts

Usage

```
mergeFunMassBBS(bbsData, funMass, printMissing = T)
```

Arguments

bbsData	The input bbsData.
funMass	The funMass list (from 'funcMass()').
printMissing	Logical. Prints to screen the missing species.

mungeSubsetData *Munge the subsetting data*

Description

munge the subbsetted data

Usage

```
mungeSubsetData(df)
```

Arguments

df	A data frame

myTheme *Plotting themes for BBS RDM*

Description

sets a plotting theme for plots

Usage

```
myTheme()
```

saveMyResults *Save results from distance travelled*

Description

Writes the results of distance travelled to file as .feather.

Usage

```
saveMyResults(results, resultsDir, metricInd)
```

Arguments

results	A data frame or list element with columns 'time', 'metricType', and 'metricValue' for either the EWS or Distance results.
resultsDir	Where to save the feather.
metricInd	One of 'distances' or 'ews'. Used in outfile name.

sort.year.line *Plot a single transect over multiple years, with one metricTYpe.*

Description

Plot a single transect over multiple years, with one metricTYpe.

Usage

```
## S3 method for class 'year.line'
sort(df, metric.ind, year.ind, dirID.ind, direction,
  scale = TRUE, center = TRUE, min.data = 5)
```

subsetByAOU *Subset the BBS data by species, functional traits, and/or body mass.*

Description

Subset the BBS data by species, functional traits, and/or body mass.

Usage

```
subsetByAOU(myData, subset.by = c("remove.fowl", "remove.shorebirds",
  "remove.shoreWaderFowl"))
```

Arguments

myData	A data frame including the column "aou".
subset.by	One or more of 'remove.fowl' (removes waterfowl), "remove.shorebirds" (removes shorebirds and waders), 'remove.shoreWaderFowl' (removes shorebirds, waders, and fowl).
mass	Logical. Retrieves body mass information (Dunning reference).

Index

∗Topic **bbs,**
createSamplingGrid, 3
∗Topic **routes**
createSamplingGrid, 3

bbsRDM, 2
bbsRDM-package *(*bbsRDM*)*, 2
birdsToFeathers, 2

calculateMetrics, 3
createSamplingGrid, 3

funcMass, 4

getDataBBS, 4
getMilBases, 5
GetRegions, 5
getRouteInfo, 6
GetSpNames, 7
GetUnzip, 7

hello, 8

importDataBBS, 8
importResults, 9

loadBirdFeathers, 9

mergeFunMassBBS, 10
mungeSubsetData, 10
myTheme, 10

saveMyResults, 11
sort.year.line, 11
subsetByAOU, 11