

# Package 'letsR'

July 2, 2014

**Type** Package

**Title** Tools for data handling and analysis in macroecology

**Version** 1.0

**Date** 2014-18-05

**Author** Bruno Vilela & Fabricio Villalobos

**Maintainer** Bruno Vilela <brunovilelasilva@hotmail.com>

**Description** R functions for handling and analyzing geographic data, mainly species' geographic distributions (in ESRI shapefile format) and environmental variables (in raster format), as well as species' information related to their description (e.g. taxonomy and year of description) and conservation status (e.g. category of threat, population trend) as provided by the IUCN's RedList database. The package includes functions to create presence-absence matrices based on species distributions and user-defined grid systems, from which several other functions could be applied to generate, for example, species richness rasters and maps, and geographical midpoints of species. In addition, it provides a function to create spatial correlograms of variables, based on the Moran's I index, under the equiprobable or equidistant criterion to define distance classes.

**License** GPL-2

**Imports** maps, raster, XML, geosphere, maptools, sp

**Depends** R (>= 2.10)

**NeedsCompilation** no

**Repository** CRAN

**Date/Publication** 2014-05-19 07:10:39

## R topics documented:

iucn . . . . .	2
lets.addpoly . . . . .	3
lets.addvar . . . . .	4
lets.correl . . . . .	5
lets.field . . . . .	6
lets.iucn . . . . .	7
lets.iucn.ha . . . . .	8
lets.iucn.his . . . . .	9
lets.iucncont . . . . .	10
lets.maplizer . . . . .	11
lets.midpoint . . . . .	12
lets.presab . . . . .	13
lets.presab.birds . . . . .	15
lets.shFilter . . . . .	17
lets.transf . . . . .	18
letsR . . . . .	19
PAM . . . . .	19
Phyllomedusa . . . . .	20
plot.PresenceAbsence . . . . .	20
print.PresenceAbsence . . . . .	21
print.summary.PresenceAbsence . . . . .	22
summary.PresenceAbsence . . . . .	22
temp . . . . .	23
<b>Index</b>	<b>24</b>

---

iucn

*IUCN avaliation for frogs of Phyllomedusa genera*


---

### Description

Result of the function `lets.iucn` applied to Southern American frog genera Phyllomedusa in 2014.

### Usage

```
data(iucn)
```

---

lets.addpoly	<i>Add polygon values to a PresenceAbsence object</i>
--------------	---

---

**Description**

Add polygon values to a PresenceAbsence object.

**Usage**

```
lets.addpoly(x, y, z, onlyvar=F)
```

**Arguments**

x	A PresenceAbsence object.
y	Polygon values to add.
z	Column name of where is located the polygon names to be used.
onlyvar	If TRUE only the matrix object will be returned.

**Value**

The result is a presence-absence matrix of species with the polygons names added as columns at the right-end of the matrix . The Values represent the percentage of the cell covered by the polygon.

**Author(s)**

Bruno Vilela

**See Also**

[lets.presab.birds](#)  
[lets.presab](#)  
[lets.addvar](#)

**Examples**

```
## Not run:  
data(PAM) # Phyllomedusa presence-absence matrix  
data(wrld_simpl) # World map  
Brazil <- wrld_simpl[wrld_simpl$NAME=="Brazil", ] # Brazil (polygon)  
  
# Check where is the variable name (in this case it is in "NAME" which will be my z value)  
names(Brazil)  
  
PAM_pol <- lets.addpoly(PAM, Brazil, "NAME")  
  
## End(Not run)
```

---

`lets.addvar`*Add variables to a PresenceAbsence object*

---

### Description

Add variables (in raster format) to a PresenceAbsence object.

### Usage

```
lets.addvar(x, y, onlyvar=F, fun=mean)
```

### Arguments

<code>x</code>	A PresenceAbsence object.
<code>y</code>	Variables to be added in Raster or RasterStack format.
<code>onlyvar</code>	If TRUE only the matrix object will be returned.
<code>fun</code>	Function used to aggregate values.

### Value

The result is a presence-absence matrix of species with the variables added as columns at the right-end of the matrix.

### Author(s)

Bruno Vilela

### See Also

[lets.presab.birds](#)

[lets.presab](#)

[lets.addpoly](#)

### Examples

```
## Not run:  
data(temp) # Global mean temperature  
data(PAM) # Phyllomedusa presence-absence matrix  
PAM_temp <- lets.addvar(PAM, temp)  
  
## End(Not run)
```

---

lets.correl	<i>Compute correlogram based on Moran's I</i>
-------------	---

---

**Description**

Compute Moran's correlogram of a variable using a distance matrix.

**Usage**

```
lets.correl(x, y, z, equidistant=FALSE, plot=TRUE)
```

**Arguments**

x	A single variable in vector format or multiple variables in matrix format (as columns).
y	A distance matrix of class <code>matrix</code> .
z	The number of distance classes to use in the correlogram.
equidistant	Logical, if TRUE the classes will be equidistant. If FALSE the classes will have equal number of observations.
plot	Logical, if TRUE the correlogram will be plotted.

**Value**

Returns a matrix with the Moran's I Observed value, Standard Deviation and Expected value. Also the p value of the null model, the mean distance between classes and the number of observations.

**Author(s)**

Bruno Vilela, Fabricio Villalobos, Lucas Jardim & Jose Alexandre Diniz-Filho

**Examples**

```
## Not run:
var <- runif(100) # random variable

# Correlated distance matrix
distan <- matrix(runif(1000), ncol=100, nrow=100)
diag(distan) <- 0
distan[lower.tri(distan)] <- distan[upper.tri(distan)]
distan2 <- as.matrix(dist(var))
distan <- (distan)*(distan2)

moran <- lets.correl(var, distan, 5, equidistant=FALSE, plot=TRUE)

## End(Not run)
```

---

`lets.field`*Create variables based on species geographical ranges*

---

### Description

Create species' attributes based on the attribute values of species co-occurring within their ranges.

### Usage

```
lets.field(x, y, z, weighth=TRUE, count=FALSE)
```

### Arguments

<code>x</code>	A PresenceAbsence object.
<code>y</code>	Species attribute to be considered. It must be a numeric attribute.
<code>z</code>	Species names in the same order as the attributes.
<code>weighth</code>	If TRUE the value is weighted by species' range size, if FALSE the value is the mean of all species that cooccur with the species target.
<code>count</code>	Logical, if TRUE a counting window will open.

### Details

If the species do not co-occur with any other species NaN will be returned.

### Author(s)

Bruno Vilela

### See Also

[lets.presab.birds](#)

[lets.presab](#)

### Examples

```
## Not run:  
data(PAM)  
range <- colSums(PAM$P)[- (1:2)]  
field <- lets.field(PAM, range, PAM$S, weighth=TRUE)  
  
## End(Not run)
```

---

`lets.iucn`*Download species' information from IUCN*

---

**Description**

Get species' information from the IUCN website([http://http://www.iucnedlist.org/](http://www.iucnedlist.org/)) for one or more species.

**Usage**

```
lets.iucn(input, count=FALSE)
```

**Arguments**

<code>input</code>	Character vector with one or more species names, or an object of class <code>PresenceAbsence</code> .
<code>count</code>	Logical, if TRUE a counting window will open.

**Details**

Note that you must be connected to the internet to use this function.

**Value**

Returns a data frame with the Species name, Family, Status, Criteria used to establish the conservation status, Population Status, Year of Description, and the Countries where it occurs. If species do not have information (i.e. have not been evaluated), the result is: NE (Not evaluated).

**Author(s)**

Bruno Vilela

**See Also**

[lets.iucn.ha](#)

[lets.iucn.his](#)

**Examples**

```
## Not run:  
# Single species  
lets.iucn("Pongo pygmaeus")  
  
# Multiple species  
lets.iucn(c("Musonycteris harrisoni", "Ailuropoda melanoleuca", "Cebus flavius"))  
  
## End(Not run)
```

---

`lets.iucn.ha`*Download habitat information for species from IUCN*

---

### Description

Get species' habitat information from IUCN website([http://:http://www.iucnedlist.org/](http://www.iucnedlist.org/)) for one or more species.

### Usage

```
lets.iucn.ha(input, count=FALSE)
```

### Arguments

<code>input</code>	Character vector with one or more species names, or an object of the Presence-Absence class.
<code>count</code>	Logical, if TRUE a counting window will open.

### Details

Note that you must be connected to the internet to use this function.

### Value

A data frame with species names in the first column and the habitats in the remaining columns, '1' if species is present in that habitat and '0' otherwise.

### Author(s)

Bruno Vilela

### See Also

[lets.iucn](#)

[lets.iucn.his](#)

### Examples

```
## Not run:  
# Single species  
lets.iucn.ha("Pongo pygmaeus")  
  
# Multiple species  
lets.iucn.ha(c("Musonycteris harrisoni", "Ailuropoda melanoleuca", "Cebus flavius"))  
  
## End(Not run)
```



---

lets.iucn.his                      *Download historical conservation status of species from IUCN*

---

## Description

Get species conservation status through history (i.e. from 1980 to present date available) from the IUCN website([http://:http://www.iucnedlist.org/](http://http://www.iucnedlist.org/)) for one or more species.

## Usage

```
lets.iucn.his(input, count=FALSE)
```

## Arguments

input	character vector with one or more species names, or an object of class Presence-Absence.
count	Logical, if TRUE a counting window will open.

## Details

Note that you must be connected to the internet to use this function.

## Value

A data frame with the species names in the first column rows and the years (1980 - present) in the remaining columns, the code represents the species' conservation status (see the IUCN website for details). If species do not have information (i.e. have not been evaluated), the result is: NE (Not evaluated).

## Author(s)

Bruno Vilela

## See Also

[lets.iucn.ha](#)  
[lets.iucn](#)

## Examples

```
## Not run:  
# Single species  
lets.iucn.his("Panthera onca")  
  
# Multiple species  
lets.iucn.his(c("Rhincodon typus", "Ailuropoda melanoleuca"))  
  
## End(Not run)
```

---

`lets.iucncont`*Transform IUCN conservation status to continuous values*

---

**Description**

Transform IUCN conservation status to continuous values from 0 to 5.

**Usage**

```
lets.iucncont(x, dd=NA, ne=NA)
```

**Arguments**

<code>x</code>	A vector or a matrix containing IUCN codes to be transformed.
<code>dd</code>	The value to be attributed to DD (data-deficient) species, the default option is NA.
<code>ne</code>	The value to be attributed to NE (not-evaluated) species, the default option is NA.

**Details**

EX and EW = 5  
EN = 4  
CR = 3  
VU = 2  
NT = 1  
LC = 0  
DD = NA  
NE = NA

**Value**

Returns a vector/matrix with continuous values from 0 to 5.

**Author(s)**

Bruno Vilela

**See Also**

[lets.iucn](#)

**Examples**

```
## Not run:
#Vector transformation
status <- sample(c("EN","VU", "NT", "CR", "DD", "LC", "EX"), 30, replace=TRUE)
transV <- lets.iucncont(status)

#matrix transformation
data(IUCN)
transM <- lets.iucncont(iucn)

## End(Not run)
```

---

lets.maplizer	<i>Create a matrix summarizing species' attributes</i>
---------------	--

---

**Description**

Summarize species attributes per cell in a presence-absence matrix.

**Usage**

```
lets.maplizer(x, y, z, func=mean, ras=FALSE)
```

**Arguments**

x	A PresenceAbsence object.
y	Species attribute to be considered.
z	Species names in the same order as the attributes.
func	function to summarize the attribute.
ras	If TRUE the raster file will be returned together with the matrix

**Value**

Return a matrix with coordinates and the attributes summarized.

**Author(s)**

Bruno Vilela

**See Also**

[lets.presab](#)

[lets.presab.birds](#)

## Examples

```
## Not run:
data(PAM)
trait <- runif(32)
resu <- lets.maplizer(PAM, trait, PAM$$, ras=TRUE)
head(resu$Matrix)
plot(resu$raster) ; map(add=T)

## End(Not run)
```

---

lets.midpoint	<i>Species' geographical range midpoint</i>
---------------	---

---

## Description

Calculate species' geographical range midpoint from a presence-absence matrix.

## Usage

```
lets.midpoint(pam, planar=FALSE)
```

## Arguments

pam	A presence-absence matrix (sites in the rows and species in the columns, with the first two columns being longitude and latitude coordinates, respectively), or an object of class PresenceAbsence.
planar	Logical, if FALSE the coordinates are in Longitude/Latitude. If TRUE the coordinates are planar.

## Value

A matrix containing the species' names and coordinates (longitude [x], latitude [y]) of species' midpoints.

## Author(s)

Fabricio Villalobos & Bruno Vilela

## See Also

[lets.presab](#)

[lets.presab.birds](#)

**Examples**

```
## Not run:
data(PAM)
mid <- lets.midpoint(PAM)

## End(Not run)
```

lets.presab

*Shapefiles into a presence-absence matrix***Description**

Convert species shapefiles into a presence-absence matrix.

**Usage**

```
lets.presab(shapes, xmn=-180, xmx=180, ymn=-90, ymx=90, resol=1,
remove.cells=TRUE, remove.sp=TRUE, show.matrix=FALSE,
crs=CRS("+proj=longlat +datum=WGS84"), cover=0, presence=NULL,
origin=NULL, seasonal=NULL, count=FALSE)
```

**Arguments**

shapes	Object of class SpatialPolygonsDataFrame (see function <a href="#">readShapePoly</a> to open these files). Species name should be in a column (within the .DBF table of the shapefile) called BINOMIAL or binomial.
xmx	Maximum longitude used to construct the grid in which the matrix will be based (i.e. the [gridded] geographic domain of interest)
xmn	Minimum longitude used to construct the grid in which the matrix will be based (i.e. the [gridded] geographic domain of interest)
ymx	Maximum latitude used to construct the grid in which the matrix will be based (i.e. the [gridded] geographic domain of interest)
ymn	Minimum latitude used to construct the grid in which the matrix will be based (i.e. the [gridded] geographic domain of interest)
resol	Numeric vector of length 1 or 2 to set the grid resolution.
remove.cells	Logical, if TRUE the final matrix will not contain cells in the grid with a value of zero (i.e. sites with no species present).
remove.sp	Logical, if TRUE the final matrix will not contain species that do not match any cell in the grid.
show.matrix	Logical, if TRUE only the presence-absence matrix will be shown.
crs	Character representing the PROJ.4 type description of a Coordinate Reference System (map projection).
cover	Percentage of the cell covered by the shapefile that will be considered for presence (values between 0 and 1).

presence	A vector with the code numbers for the presence type to be considered in the process (for IUCN spatial data <a href="http://www.iucnredlist.org/technical-documents/spatial-data">http://www.iucnredlist.org/technical-documents/spatial-data</a> ).
origin	A vector with the code numbers for the origin type to be considered in the process (for IUCN spatial data).
seasonal	A vector with the code numbers for the seasonal type to be considered in the process (for IUCN spatial data).
count	Logical, if TRUE a counting window will open.

### Details

The function creates the presence-absence matrix based on a raster file. Depending on the cell size, extension used and number of species it may require a lot of memory, and may take some time to process it. Thus, during the process, if count argument is set TRUE, a counting window will open so you can see the progress (i.e. in what polygon the function is working). Note that the number of polygons is not the same as the number of species that you have (i.e. a species may have more than one polygon/shapefiles).

### Value

The result is an object of class PresenceAbsence with the following objects:

**Presence-Absence Matrix:** A matrix of species' presence(1) and absence(0) information. The first two columns contain the longitude (x) and latitude (y) of the cells' centroid (from the gridded domain used);

**Richness Raster:** A raster containing species richness data;

**Species name:** A character vector with species' names contained in the matrix.

\*But see the option argument `show.matrix`.

### Author(s)

Bruno Vilela & Fabricio Villalobos

### See Also

[plot.PresenceAbsence](#)  
[lets.presab.birds](#)  
[lets.shFilter](#)

### Examples

```
## Not run:
data(Phyllomedusa) # Spatial distribution polygons of south american frogs of genus Phyllomedusa.
PAM <- lets.presab(Phyllomedusa, xmn=-93, xmx=-29, ymn= -57, ymx=15)
summary(PAM)
plot(PAM) # Species richness map
plot(PAM, name="Phyllomedusa nordestina") # Map of the specific species

## End(Not run)
```

---

lets.presab.birds      *Shapefiles to presence/absence matrix by folder location*

---

## Description

Convert species' shapefiles into a presence-absence matrix. This function is specially designed to work with BirdLife Intl. shapefiles (<http://www.birdlife.org>).

## Usage

```
lets.presab.birds(path, xmn=-180, xmx=180, ymn=-90, ymx=90, resol=1,
remove.cells=TRUE, remove.sp=TRUE, show.matrix=FALSE,
crs=CRS("+proj=longlat +datum=WGS84"), cover=0, presence=NULL,
origin=NULL, seasonal=NULL, count=FALSE)
```

## Arguments

path	Path location of folders with one or more species' shapefiles.
xmx	Maximum longitude used to construct the grid in which the matrix will be based (i.e. the [gridded] geographic domain of interest)
xmn	Minimum longitude used to construct the grid in which the matrix will be based (i.e. the [gridded] geographic domain of interest)
ymx	Maximum latitude used to construct the grid in which the matrix will be based (i.e. the [gridded] geographic domain of interest)
ymn	Minimum latitude used to construct the grid in which the matrix will be based (i.e. the [gridded] geographic domain of interest)
resol	Numeric vector of length 1 or 2 to set the grid resolution.
remove.cells	Logical, if TRUE the final matrix will not contain cells in the grid with a value of zero (i.e. sites with no species present).
remove.sp	Logical, if TRUE the final matrix will not contain species that do not match any cell in the grid.
show.matrix	Logical, if TRUE only the presence-absence matrix will be shown.
crs	Character representing the PROJ.4 type description of a Coordinate Reference System (map projection).
cover	Percentage of the cell covered by the shapefile that will be considered for presence (values between 0 and 1).
presence	A vector with the code numbers for the presence type to be considered in the process (for IUCN spatial data <a href="http://www.iucnredlist.org/technical-documents/spatial-data">http://www.iucnredlist.org/technical-documents/spatial-data</a> ).
origin	A vector with the code numbers for the origin type to be considered in the process (for IUCN spatial data).
seasonal	A vector with the code numbers for the seasonal type to be considered in the process (for IUCN spatial data).
count	Logical, if TRUE a counting window will open.

## Details

The function creates the presence-absence matrix based on a raster file. Depending on the cell size, extension used and number of species it may require a lot of memory, and may take some time to process it. Thus, during the process, if count argument is set TRUE, a counting window will open so you can see the progress (i.e. in what polygon the function is working). Note that the number of polygons is not the same as the number of species that you have (i.e. a species may have more than one polygon/shapefiles).

## Value

The result is an object of class PresenceAbsence with the following objects:

**Presence-Absence Matrix:** A matrix of species' presence(1) and absence(0) information. The first two columns contain the longitude (x) and latitude (y) of the cells' centroid (from the gridded domain used);

**Richness Raster:** A raster containing species richness data;

**Species name:** A vector with species' names contained in the matrix.

\*But see the option argument `show.matrix`.

## Author(s)

Bruno Vilela & Fabricio Villalobos

## See Also

[plot.PresenceAbsence](#)

[lets.presab](#)

[lets.shFilter](#)

## Examples

```
## Not run:  
# Constructing a Presence/Absence matrix for birds  
#(this will not work if do not change the path for the folder were you kept the spatial data)  
PAM <- lets.presab.birds("YOURPATH/YOURPATH/BIRDS")  
plot(PAM) # Species richness map
```

```
## End(Not run)
```



---

lets.shFilter	<i>Shapefiles filtering</i>
---------------	-----------------------------

---

**Description**

Filter species shapefiles by origin, presence and seasonal type (following IUCN types).

**Usage**

```
lets.shFilter(shapes, presence=NULL, origin=NULL, seasonal=NULL)
```

**Arguments**

shapes	Object of class SpatialPolygonsDataFrame (see function readShapePoly to open this files).
presence	A vector with the code numbers for the presence type to be maintained.
origin	A vector with the code numbers for the origin type to be maintained.
seasonal	A vector with the code numbers for the seasonal type to be maintained.

**Details**

Presence codes: (1) Extant, (2) Probably Extant, (3) Possibly Extant, (4) Possibly Extinct, (5) Extinct (post 1500) & (6) Presence Uncertain.

Origin codes: (1) Native, (2) Reintroduced, (3) Introduced, (4) Vagrant & (5) Origin Uncertain.

Seasonal codes: (1) Resident, (2) Breeding Season, (3) Non-breeding Season, (4) Passage & (5) Seasonal Occurrence Uncertain.

More info in the shapefiles' metadata.

**Value**

The result is the shapefile(s) filtered according to the selected types. If the filters remove all polygons, the result will be NULL.

**Author(s)**

Bruno Vilela

**See Also**

[plot.PresenceAbsence](#)

[lets.presab](#)

[lets.presab.birds](#)

---

lets.transf	<i>Transform values of a vector</i>
-------------	-------------------------------------

---

**Description**

Transform each element of a vector.

**Usage**

```
lets.transf(x, y, z, NUMERIC=TRUE)
```

**Arguments**

x	A vector to be transformed.
y	levels to be transformed.
z	The value to be attributed to each level (same order as y).
NUMERIC	logical, if TRUE z will be considered numbers.

**Value**

Return a vector with changed values.

**Author(s)**

Bruno Vilela

**Examples**

```
## Not run:
status <- sample(c("EN","VU", "NT", "CR", "DD", "LC"), 30, replace=TRUE)
TE <- "Threatened"
NT <- "Non-Threatened"
new <- c(TE, TE, NT, TE, "Data Deficient", NT)
old <- c("EN","VU", "NT", "CR", "DD", "LC")
statustrans <- lets.transf(status, old, new, NUMERIC=FALSE)

## End(Not run)
```

---

`letsR`*Tools for data handling and analysis in macroecology.*

---

### Description

R functions for handling and analyzing geographic data, mainly species' geographic distributions (in ESRI shapefile format) and environmental variables (in raster format), as well as species' information related to their description (e.g. taxonomy and year of description) and conservation status (e.g. category of threat, population trend) as provided by the IUCN's RedList database. The package includes functions to create presence-absence matrices based on species distributions and user-defined grid systems, from which several other functions could be applied to generate, for example, species richness rasters and maps, and geographical midpoints of species. In addition, it provides a function to create spatial correlograms of variables, based on the Moran's I index, under the equiprobable or equidistant criterion to define distance classes. The letsR package is in continuous development and suggestions are more than welcome! We hope you enjoy it and find it useful.

### Details

Package: `letsR`  
Type: Package  
Version: 1.0  
Date: 2014-05-05  
License: GPL-2

### Author(s)

Bruno Vilela <brunovilelasilva@hotmail.com>  
Fabricio Villalobos <fabricio.villalobos@gmail.com>

---

`PAM`*PresenceAbsence object for frogs of Phyllomedusa genera*

---

### Description

PresenceAbsence object obtained using the function `lets.presab` in the Geographic distribution of the Southern American frog genera *Phyllomedusa*.

### Usage

```
data(PAM)
```

**Source**

IUCN - <http://www.iucnredlist.org/>. 2014.

---

Phyllomedusa                      *Geographic distribution of Phyllomedusa genera*

---

**Description**

Geographic distribution of the Southern American frog genera Phyllomedusa. Data was modified from IUCN (<http://www.iucnredlist.org/>, downloaded in 05/2014). There are 32 species and 46 polygons.

**Usage**

```
data(Phyllomedusa)
```

**Source**

IUCN - <http://www.iucnredlist.org/>. 2014.

---

`plot.PresenceAbsence`    *Plot an object of class PresenceAbsence*

---

**Description**

Plots species richness map from an object of class PresenceAbsence or an specific species map.

**Usage**

```
## S3 method for class 'PresenceAbsence'
plot(x, name=NULL, world=TRUE, ...)
```

**Arguments**

<code>x</code>	an object of class PresenceAbsence (see function presab).
<code>name</code>	you can specify a species to be plotted instead of the species richness map.
<code>world</code>	if TRUE a map of political divisions (countries) is added to the plot.
<code>...</code>	Other plot parameters.

**Author(s)**

Bruno Vilela

### See Also

[lets.presab](#)

[lets.presab.birds](#)

### Examples

```
## Not run:  
data(PAM)  
plot(PAM)  
plot(PAM, name="Phyllomedusa atelopoides")  
plot(PAM, name="Phyllomedusa azurea")  
  
## End(Not run)
```

---

*print.PresenceAbsence* *Print for object of class PresenceAbsence*

---

### Description

Print for objects of class PresenceAbsence.

### Usage

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

### Arguments

x                    an object of class PresenceAbsence (see function presab).  
...                   Other print parameters.

### Author(s)

Bruno Vilela

---

```
print.summary.PresenceAbsence
```

*Print summary for object of class PresenceAbsence*

---

### Description

Print summary for objects of class PresenceAbsence.

### Usage

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

### Arguments

x                    an object of class PresenceAbsence (see function presab).  
...                   Other print parameters.

### Author(s)

Bruno Vilela

---

```
summary.PresenceAbsence
```

*Summary for object of class PresenceAbsence*

---

### Description

Summary for objects of class PresenceAbsence.

### Usage

```
## S3 method for class 'PresenceAbsence'  
summary(object, ...)
```

### Arguments

object                an object of class PresenceAbsence (see function presab).  
...                    Other summary parameters.

### Author(s)

Bruno Vilela

---

`temp`*Avarege temperature raster for the world*

---

**Description**

Avarege temperature raster for the world in 10 min of resolution. Data was modified from WorldClim (<http://worldclim.com/>, downloaded in 05/2014).

**Usage**

```
data(temp)
```

**Source**

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.

# Index

## \*Topic **package**

letsR, [19](#)

iucn, [2](#)

lets.addpoly, [3, 4](#)

lets.addvar, [3, 4](#)

lets.correl, [5](#)

lets.field, [6](#)

lets.iucn, [2, 7, 8–10](#)

lets.iucn.ha, [7, 8, 9](#)

lets.iucn.his, [7, 8, 9](#)

lets.iucncont, [10](#)

lets.maplizer, [11](#)

lets.midpoint, [12](#)

lets.presab, [3, 4, 6, 11, 12, 13, 16, 17, 19, 21](#)

lets.presab.birds, [3, 4, 6, 11, 12, 14, 15, 17, 21](#)

lets.shFilter, [14, 16, 17](#)

lets.transf, [18](#)

letsR, [19](#)

letsR-package (letsR), [19](#)

PAM, [19](#)

Phyllomedusa, [19, 20](#)

plot.PresenceAbsence, [14, 16, 17, 20](#)

print.PresenceAbsence, [21](#)

print.summary.PresenceAbsence, [22](#)

readShapePoly, [13](#)

summary.PresenceAbsence, [22](#)

temp, [23](#)