

# Package ‘boris’

September 12, 2019

**Title** Boris' miscellanea

**Description** Miscellaneous functions of boris. It contains general utils,  
model, plot and scientific utils. Probably in future to be splitted into  
multiple packages.

**Version** 0.21

**Maintainer** Gianni Boris Pezzatti <boris.pezzatti@wsl.ch>

**Author** Gianni Boris Pezzatti

**License** GPL-3

**Imports** dplyr, tidyr, tibble, stringr, foreach, zoo, SDMTools, dismo

**Suggests** testthat, roxygen2

**RoxygenNote** 6.1.1

**NeedsCompilation** no

## R topics documented:

accuracy_glm_cross . . . . .	2
accuracy_glm_simple . . . . .	3
accuracy_me_cross . . . . .	3
accuracy_me_simple . . . . .	4
accuracy_simple . . . . .	4
bkr . . . . .	5
bkr_for_tpr . . . . .	5
boris . . . . .	6
calcArea . . . . .	6
calcAreaLim . . . . .	6
cor2df . . . . .	7
cor2df_fire . . . . .	8
cordf . . . . .	9
dep_vars . . . . .	9
evaltext . . . . .	10
filename . . . . .	11
fill_1_na . . . . .	11
fill_na . . . . .	12
formulae . . . . .	12
formulae_cleaned . . . . .	13
fpr . . . . .	14
fpr_for_tpr . . . . .	14

getArgs . . . . .	15
glm_pseudoabsence . . . . .	15
ind_vars . . . . .	16
kfold_seq . . . . .	16
maxent_formula . . . . .	17
me_constants . . . . .	17
me_lambdas . . . . .	18
me_parNum . . . . .	18
me_predict . . . . .	19
mirror_na . . . . .	19
models . . . . .	20
orderfactor . . . . .	20
read_fwf_fixedheader . . . . .	21
replace_na . . . . .	21
resample_meteo_h2d . . . . .	22
rescale01 . . . . .	23
tpr . . . . .	23
vpd . . . . .	24
without_na . . . . .	24
yearplots . . . . .	25

<b>Index</b>	<b>26</b>
--------------	-----------

---

accuracy_glm_cross	<i>Accuracy function to evaluate a gml model for training and test dataset</i>
--------------------	--

---

## Description

Evaluate according to multiple criteria a gml model for training and test dataset

## Usage

```
accuracy_glm_cross(m, abundance, test, depvar_name, abundance_name)
```

## Arguments

m	glm model
abundance	abundances, should be passed in same order as presences and same length
test	dataframe holding test data
depvar_name	name of te dependent variable in the test dataframe
abundance_name	name of te abundance variable in the test dataframe

## Value

a vector of named arguments (n=number data, np=number of presences, )

---

accuracy_glm_simple	<i>Accuracy function to evaluate a glm model</i>
---------------------	--

---

**Description**

Evaluate according to multiple criteria a glm model

**Usage**

```
accuracy_glm_simple(m, p, a, abundance = NULL)
```

**Arguments**

m	glm model
p	presences
a	absences
abundance	abundances, should be passed in same order as presences and same length

**Value**

a vector of named arguments (n=number data, np=number of presences, )

---

accuracy_me_cross	<i>Accuracy function to evaluate a maxent model for training and test dataset</i>
-------------------	---

---

**Description**

Evaluate according to multiple criteria a maxent model ([maxent](#)) for training and test dataset

**Usage**

```
accuracy_me_cross(me, abundance, test = NULL, depvar_name,  
  abundance_name)
```

**Arguments**

me	maxent model
abundance	abundances, should be passed in same order as presences and same length
test	dataframe holding test data
depvar_name	name of te dependent variable in the test dataframe
abundance_name	name of te abundance variable in the test dataframe

**Value**

a vector of named arguments (n=number data, np=number of presences, )

---

accuracy_me_simple	<i>Accuracy function to evaluate a maxent model</i>
--------------------	---

---

**Description**

Evaluate according to multiple criteria a maxent model ([maxent](#))

**Usage**

```
accuracy_me_simple(me, p, a, abundance = NULL)
```

**Arguments**

me	maxent model
p	presences
a	absences
abundance	abundances, should be passed in same order as presences and same length

**Value**

a vector of named arguments (n=number data, np=number of presences, auc=usual auc, auc.bg=auc on background, auc.me=auc for maxent, aicc.me=aicc for maxent )

---

accuracy_simple	<i>Accuracy function to evaluate a presence/absence/background model</i>
-----------------	--

---

**Description**

Evaluate according to multiple criteria

**Usage**

```
accuracy_simple(p, a, abundance = NULL)
```

**Arguments**

p	presences
a	absences
abundance	abundances, should be passed in same order as presences and same length

**Value**

a vector of named arguments (n=number data, np=number of presences, auc=auc, auc.bg=auc on the background)

---

bkr	<i>Background proportion</i>
-----	------------------------------

---

**Description**

Background proportion

**Usage**

```
bkr(d, thr, depvar)
```

**Arguments**

d	dataframe
thr	threshold value
depvar	column holding the model output

**Value**

background proportion

---

bkr_for_tpr	<i>Background portion for a given true positive rate</i>
-------------	--

---

**Description**

Background portion for a given true positive rate

**Usage**

```
bkr_for_tpr(d, tp.rate, depvar, occurrence)
```

**Arguments**

d	dataframe
tp.rate	true positive rate
depvar	column holding the model output
occurrence	column holding presence [0/1]

**Value**

background proportion

---

boris	<i>boris</i>
-------	--------------

---

**Description**

boris

---

calcArea	<i>Area under a curve</i>
----------	---------------------------

---

**Description**

Calculates the trapezoid area (boxes+traingles) under the curve  $y=f(x)$

**Usage**

```
calcArea(x, y)
```

**Arguments**

x	vector holding the x values
y	vector holding the y values

**Value**

area under the curve

**See Also**

[calcAreaLim](#)

---

calcAreaLim	<i>Area under a curve</i>
-------------	---------------------------

---

**Description**

Calculates the trapezoid area (boxes+traingles) under the curve  $y=f(x)$  up to a given x limit (xupper), when given

**Usage**

```
calcAreaLim(x, y, xupper = NULL)
```

**Arguments**

x	vector holding the x values
y	vector holding the y values
xupper	x value

## Details

If xupper is not one of the x values, the corresponding y value is calculated using the approx function

## Value

area under the curve

## See Also

[calcArea](#)

---

cor2df	<i>Variable pairs correlated above a threshold</i>
--------	--

---

## Description

This function returns a dataframe with the variable pairs above a given correlation threshold

## Usage

```
cor2df(cor.matrix, threshold = 0.6)
```

## Arguments

cor.matrix	correlation matrix
threshold	correlation threshold

## Details

It is based on the [cor](#) function, but instead of a correlation matrix it returns a dataframe with the pairwise combinations above a threshold.

## Value

a dataframe holding the variable pairs with a correlation higher than the specified threshold

## See Also

[cordf](#)

---

cor2df_fire	<i>Correlations above a threshold, showing aicc's of a logistic according to fire presence</i>
-------------	--

---

## Description

This function returns a dataframe with the variable pairs above a given correlation threshold, and the aic value of a logistic model with fire occurrence

## Usage

```
cor2df_fire(data, vars = NULL, fire, threshold, use = "everything",  
            method = c("pearson", "kendall", "spearman"))
```

## Arguments

data	dataframe with the data
vars	vector of column names or column numbers holding the variables to analyse. If not specified all the columns will be used.
fire	column name of number holding fire presence [0/1]
threshold	correlation threshold
use	an optional character string giving a method for computing covariances in the presence of missing values. This must be (an abbreviation of) one of the strings "everything", "all.obs", "complete.obs", "na.or.complete", or "pairwise.complete.obs"
method	a character string indicating which correlation coefficient is to be computed. One of "pearson" (default), "kendall", or "spearman", can be abbreviated

## Details

It is based on the [cor](#) function, but instead of a correlation matrix it returns a dataframe with the pairwise combinations above a threshold.

## Value

a list with

- cors: dataframe holding the variable pairs with a correlation higher than the specified threshold, and the relative aicc's'
- aiccs: dataframe holding all the aicc's' for all variables

## See Also

[cordf](#)



---

cordf	<i>Correlations above a threshold</i>
-------	---------------------------------------

---

### Description

This function returns a dataframe with the variable pairs above a given correlation threshold

### Usage

```
cordf(data, vars = NULL, threshold = 0.6, use = "everything",  
      method = c("pearson", "kendall", "spearman"))
```

### Arguments

data	dataframe with the data
vars	vector of column names or column numbers holding the variables to analyse. If not specified all the columns will be used.
threshold	correlation threshold
use	an optional character string giving a method for computing covariances in the presence of missing values. This must be (an abbreviation of) one of the strings "everything", "all.obs", "complete.obs", "na.or.complete", or "pairwise.complete.obs"
method	a character string indicating which correlation coefficient is to be computed. One of "pearson" (default), "kendall", or "spearman", can be abbreviated

### Details

It is based on the [cor](#) function, but instead of a correlation matrix it returns a dataframe with the pairwise combinations above a threshold.

### Value

a dataframe holding the variable pairs with a correlation higher than the specified threshold

### See Also

[cor2df](#)

---

dep_vars	<i>Dependent variable</i>
----------	---------------------------

---

### Description

Extract the name of the dependent variable from formula

### Usage

```
dep_vars(formula)
```

**Arguments**

formula                      formula to inspect, either as formula object or string

**Value**

name of the dependent variable

**See Also**

[all.vars](#) from base package to get all variables

---

evaltext	<i>Concatenate and evaluate string expressions in a specified environment</i>
----------	---

---

**Description**

This function allows to write in a shorter form the evaluation of a vector of characters.

**Usage**

```
evaltext(..., envir = parent.frame(), enclos = if (is.list(envir) ||
  is.pairlist(envir)) parent.frame() else baseenv(), sep = "")
```

**Arguments**

...                      character strings holding the code to be evaluated

envir                    the environment in which expr is to be evaluated. May also be NULL, a list, a data frame, a pairlist or an integer as specified to sys.call.

enclos                   Relevant when envir is a (pair)list or a data frame. Specifies the enclosure, i.e., where R looks for objects not found in envir. This can be NULL (interpreted as the base package environment, baseenv()) or an environment.

sep                      separator character to be used as in the [paste](#) function

**Value**

The result of evaluating the object: for an expression vector this is the result of evaluating the last element

---

filename	<i>Filename without extension</i>
----------	-----------------------------------

---

**Description**

Strips the extension form the filename.

**Usage**

```
filename(file)
```

**Arguments**

file	name of the file
------	------------------

**Value**

file name without extension

---

fill_1_na	<i>Fill 1-value gaps in a vector</i>
-----------	--------------------------------------

---

**Description**

Fill gaps of single values with linearization (mean of the adjacent values) or repetition of previous/next value.

**Usage**

```
fill_1_na(x, method = c("linearize", "previous", "next"))
```

**Arguments**

x	numeric vector
method	how to fill in the gaps (default by linearization, otherwise by previous/next value duplication)

**Value**

numeric vector with filled 1-value gaps

---

fill_na	<i>Fill gaps in a vector</i>
---------	------------------------------

---

**Description**

Fill gaps with linearization (mean of the adjacent values) or repetition of previous/next value, by using the [na.approx](#) function.

**Usage**

```
fill_na(x, method = c("linearize", "previous", "next"), maxgap = 2)
```

**Arguments**

x	numeric vector
method	how to fill in the gaps (default by linearization, otherwise by previous/next value duplication)
maxgap	maximum number of consecutive NAs to fill. Any longer gaps will be left unchanged.

**Value**

numeric vector with filled 1-value gaps

---

formulae	<i>Formulae from variable combinations</i>
----------	--

---

**Description**

Build the formulae (as strings) from variable names

**Usage**

```
formulae(formula, dep = NULL, vars = NULL, nullmodelterm = "1",
  minsize = 1, maxsize = NULL)
```

**Arguments**

formula	formula with all the terms (beyond optimal model)
dep	name of the dependent variable. If the formula is specified, this argument is not considered.
vars	character vector with the names of the independent variables (without nullmodel term). If the formula is specified, this argument is not considered.
nullmodelterm	to specify in case of an always required fixed term (should not be included in the vars)
minsize	minimum size of the formula (number of independent variables)
maxsize	maximum size of the formula (number of independent variables). NULL means unrestricted.

**Value**

character vector hold the strings of the generated formulae.

---

formulae_cleaned	<i>Formulae from variable combinations without correlated variables</i>
------------------	---

---

**Description**

Build the formulae (as strings) from variable names

**Usage**

```
formulae_cleaned(formula, dep = NULL, vars = NULL,
  nullmodelterm = "1", minsize = 1, maxsize = NULL, data,
  threshold = 0.6, use = "everything", method = c("pearson",
  "kendall", "spearman"))
```

**Arguments**

formula	formula with all the terms (beyond optimal model)
dep	name of the dependent variable. If the formula is specified, this argument is not considered.
vars	character vector with the names of the independent variables (without nullmodel term). If the formula is specified, this argument is not considered.
nullmodelterm	to specify in case of an always required fixed term (should not be included in the vars)
minsize	minimum size of the formula (number of independent variables)
maxsize	maximum size of the formula (number of independent variables). NULL means unrestricted.
data	dataframe holding the dataset with the column names corresponding to vars
threshold	correlation threshold
use	an optional character string giving a method for computing covariances in the presence of missing values. This must be (an abbreviation of) one of the strings "everything", "all.obs", "complete.obs", "na.or.complete", or "pairwise.complete.obs"
method	a character string indicating which correlation coefficient is to be computed. One of "pearson" (default), "kendall", or "spearman", can be abbreviated

**Value**

list of

- formulae: character vector hold the strings of the generated formulae
- vars: list of variables names combinations

---

fpr	<i>False positive rate (1- specificity)</i>
-----	---

---

**Description**

False positive rate (1- specificity)

**Usage**

fpr(d, thr, depvar, occurrence)

**Arguments**

d	dataframe
thr	threshold value
depvar	column holding the model output
occurrence	column holding presence [0/1]

**Value**

false positive rate (1- specificity)

---

fpr_for_tpr	<i>False positive rate for a given true positive rate</i>
-------------	---

---

**Description**

False positive rate for a given true positive rate

**Usage**

fpr\_for\_tpr(d, tp.rate, depvar, occurrence)

**Arguments**

d	dataframe
tp.rate	true positive rate
depvar	column holding the model output
occurrence	column holding presence [0/1]

**Value**

false positive rate

---

`getArgs`*Extract and Load command line arguments into session*

---

**Description**

Loads the command line arguments supplied when this R session was invoked into the session environment.

**Usage**

```
getArgs()
```

**Value**

Nothing

---

`glm_pseudoabsence`*GLM with pseudoabsences*

---

**Description**

Wrapper around `glm` to use a pseudoabsence approach, substituting absences (zeros's) with the prevalence (mean occurrence).

**Usage**

```
glm_pseudoabsence(formula, family = gaussian, data, ...)
```

**Arguments**

<code>formula</code>	formula with dependent and independent variables
<code>family</code>	a description of the error distribution and link function to be used in the model. This can be a character string naming a family function, a family function or the result of a call to a family function. (See <a href="#">family</a> for details of family functions.)
<code>data</code>	dataframe holding the data
<code>...</code>	additional parameters to pass to <code>glm</code>

**Value**

glm model

**See Also**

[glm](#)

---

ind_vars	<i>Independent variable(s)</i>
----------	--------------------------------

---

**Description**

Extract the name of the independent variable(s) from formula

**Usage**

```
ind_vars(formula, simplify = F)
```

**Arguments**

formula	formula to inspect, either as formula object or string
simplify	if terms such as $I(x^2)$ should be retained as variable or not

**Value**

name of the independent variable(s)

**See Also**

[all.vars](#) from base package to get all variables

---

kfold_seq	<i>Sequential k-fold partitioning</i>
-----------	---------------------------------------

---

**Description**

Modified version of the [kfold](#) function, that returns subsequent (not random) folds (consistent among runs)

**Usage**

```
kfold_seq(x, k = 5, by = NULL)
```

**Arguments**

x	a vector, matrix, data.frame, or Spatial object
k	number of groups
by	Optional argument. A vector or factor with sub-groups (e.g. species). Its length should be the same as the number of records in x

**Value**

a vector with group assignments

**See Also**

[kfold](#)



---

maxent_formula	<i>Maxent with formula</i>
----------------	----------------------------

---

**Description**

Wrapper for Maxent to use a formula instead of data and presences

**Usage**

```
maxent_formula(formula, data, ...)
```

**Arguments**

formula	formula with dependent and independent variables
data	dataframe holding the data
...	additional parameters to pass to maxent

**Value**

Maxent model

**See Also**

[maxent](#)

---

me_constants	<i>Constants of a maxent model</i>
--------------	------------------------------------

---

**Description**

Extract the constants of a maxent model ([maxent](#))

**Usage**

```
me_constants(m)
```

**Arguments**

m	either a maxent model or a character vector containing the lines of a maxent lambda file
---	--

**Value**

dataframe holding the constants (what, value)

---

me_lambdas	<i>Extract lambda file values</i>
------------	-----------------------------------

---

**Description**

Fill a data frame with the lambda file values of a maxent model ([maxent](#))

**Usage**

```
me_lambdas(m)
```

**Arguments**

m	either a maxent model or a character vector containing the lines of a maxent lambda file
---	--

**Value**

dataframe with the lambda values (what, lambda, min, max)

---

me_parNum	<i>Count the number of Maxent parameters (with lambda!=0)</i>
-----------	---

---

**Description**

Counts the number of parameters (with lambda!=0) of a maxent model ([maxent](#))

**Usage**

```
me_parNum(m)
```

**Arguments**

m	maxent model
---	--------------

**Value**

number of parameters

---

me_predict	<i>Predict new values of a maxent model</i>
------------	---

---

**Description**

Calculates new values of a maxent model ([maxent](#))

**Usage**

```
me_predict(m, data)
```

**Arguments**

m	either a maxent model or a character vector containing the lines of a maxent lambda file
data	data frame holding the data

**Value**

predictions

---

mirror_na	<i>Fill gaps in a dataframe with data from another dataframe</i>
-----------	--

---

**Description**

Function replacing NA values in a dataframe with sequentially corresponding data from another dataframe of the same length and with same column names.

**Usage**

```
mirror_na(to, from, colnames, case.sensitive = T)
```

**Arguments**

to	dataframe holding the NA values to replace
from	dataframe holding the values to replace the NA's
colnames	character vector with the names of the columns
case.sensitive	logical indicating if column names are considered according to case or not

**Value**

dataframe with replaces NA's

---

models	<i>Models from formulae</i>
--------	-----------------------------

---

**Description**

Build a list of models according to different formulae, a prefix and a suffix

**Usage**

```
models(prefix, formulae, suffix, envir = parent.frame(1))
```

**Arguments**

prefix	string representation of the model prefix (e.g. "glm(")
formulae	list of formulae as string expressions
suffix	string representation of the model suffix (e.g. ", family="binomial")"
envir	environment in which to evaluate the model expression. By default evaluates in the environment that calls this function

**Value**

a list of models

---

orderfactor	<i>Change levels order</i>
-------------	----------------------------

---

**Description**

This function changes the order of the levels of a factor

**Usage**

```
orderfactor(x, neworder, ordered = is.ordered(x), ...)
```

**Arguments**

x	factor
neworder	numeric or character vector specifying the new order of the levels
ordered	logical specifying if the factor will be ordered or not (defaults to input factor class)
...	other parameters to be passed to factor function (labels, exclude)

**Value**

factor with levels order changed according to specifications

---

read_fwf_fixedheader	<i>Reads a fixed width formatted data with the header in the same format</i>
----------------------	--

---

### Description

The base function `read.fwf` can read fixed width formatted data, however when including an header, this needs to have another format (e.g. tab-separated, as specified by the `sep` argument). This function allows to read data with the header specifically in the same fixed width format as the data.

### Usage

```
read_fwf_fixedheader(file, widths, ...)
```

### Arguments

<code>file</code>	name of the file.
<code>widths</code>	integer vector, giving the widths of the fixed-width fields (of one line).
<code>...</code>	further arguments to be passed to <code>read.fwf</code> .

### Value

A data.frame as produced by `read.fwf` which is called internally.

---

replace_na	<i>Replace NA's</i>
------------	---------------------

---

### Description

Substitutes NA values in the given vector, dataframe, matrix or list

### Usage

```
replace_na(data, value)
```

### Arguments

<code>data</code>	vector, dataframe, list or matrix
<code>value</code>	replacement value

### Value

a data structure with the given value instead of NA's

---

resample_meteo_h2d	<i>Resample a data.frame with meteorological data with hourly interval to a daily interval, allowing the specification at which time to cut the day (e.g. can be summarized for noon to noon).</i>
--------------------	--

---

### Description

Resample a data.frame with meteorological data with hourly interval to a daily interval, allowing the specification at which time to cut the day (e.g. can be summarized for noon to noon).

### Usage

```
resample_meteo_h2d(h, time_h = 24, timevar, varnames,  
  aggregation = c("sample", "sum", "mean", "max", "min"), na.rm = F,  
  add_suffix = F)
```

### Arguments

h	a data.frame holding the hourly data
time_h	hour value at which to cut the hourly data.frame to build summaries (e.g. 12 for noon to noon). Defaults to 24.
timevar	name of the column holding the date-time information (in POSIX numeric format).
varnames	character vector holding the names of the columns holding the variables to be processed.
aggregation	character vector holding the types of aggregations to perform on the selected variables.
na.rm	boolean to specify if aggregation function should consider or skip NA's.
add_suffix	boolean to specify if the variable names should be completed with the specification of the aggregation.

### Value

a new data.frame with daily timestep and the selected variables and aggregations .

### See Also

[kfold](#)

---

rescale01	<i>Rescale a vector of numbers between 0 and 1</i>
-----------	--

---

**Description**

This function rescales the values of a numeric vector between 0 and 1

**Usage**

```
rescale01(x, na.rm = FALSE)
```

**Arguments**

x	numeric vector to rescale
na.rm	logical indicating whether missing values should be removed

**Value**

numeric vector with rescaled values

---

tpr	<i>True positive rate (sensitivity)</i>
-----	---

---

**Description**

True positive rate (sensitivity)

**Usage**

```
tpr(d, thr, depvar, occurrence = presence)
```

**Arguments**

d	dataframe
thr	threshold value
depvar	column holding the model output
occurrence	column holding presence [0/1]

**Value**

true positive rate (sensitivity)

---

vpd	<i>Vapour pressure deficit</i>
-----	--------------------------------

---

**Description**

Calculate vapour pressure deficit from temperature and humidity

**Usage**

```
vpd(T, H)
```

**Arguments**

T	numeric vector with air temperature values [C]
H	numeric vector with air humidity values [%]

**Value**

vapour pressure deficit

---

without_na	<i>Remove NA's</i>
------------	--------------------

---

**Description**

Return the given dataframe without the rows where one of the independent variables (extracted from formula) are NA

**Usage**

```
without_na(data, selection)
```

**Arguments**

data	dataframe with the data
selection	column names or formula with dependent and independent variables

**Details**

This is useful to link abundances to the model datasets, since the built models internally exclude those rows

**Value**

A dataframe without NA's in the columns holded by independent variables of the formula



---

yearplots

---

*Multiple plots of daily data over years*

---

**Description**

Plots summaries of daily data in a yearly plot

**Usage**

```
yearplots(data, vars, year = year, doy = doy, what = c("data",  
  "mean", "na"), rows = floor(length(vars)/cols + 1), cols = 4)
```

**Arguments**

data	dataframe with data
vars	character vector with variable (columns) names holding the data to summarise
year	name of the column holding the year
doy	name of the column holding the doy [1-366]
what	what to plot: all values, means over the years or presence of NA's
rows	number of rows for the multiple plots. Defaults according to the number of
cols	number of columns for the multiple plots. Defaults to 4

**Value**

plot

# Index

accuracy\_glm\_cross, [2](#)  
accuracy\_glm\_simple, [3](#)  
accuracy\_me\_cross, [3](#)  
accuracy\_me\_simple, [4](#)  
accuracy\_simple, [4](#)  
all.vars, [10](#), [16](#)  
  
bkr, [5](#)  
bkr\_for\_tpr, [5](#)  
boris, [6](#)  
boris-package (boris), [6](#)  
  
calcArea, [6](#), [7](#)  
calcAreaLim, [6](#), [6](#)  
cor, [7–9](#)  
cor2df, [7](#), [9](#)  
cor2df\_fire, [8](#)  
cordf, [7](#), [8](#), [9](#)  
  
dep\_vars, [9](#)  
  
evaltext, [10](#)  
  
family, [15](#)  
filename, [11](#)  
fill\_1\_na, [11](#)  
fill\_na, [12](#)  
formulae, [12](#)  
formulae\_cleaned, [13](#)  
fpr, [14](#)  
fpr\_for\_tpr, [14](#)  
  
getArgs, [15](#)  
glm, [15](#)  
glm\_pseudoabsence, [15](#)  
  
ind\_vars, [16](#)  
  
kfold, [16](#), [22](#)  
kfold\_seq, [16](#)  
  
maxent, [3](#), [4](#), [17–19](#)  
maxent\_formula, [17](#)  
me\_constants, [17](#)  
me\_lambdas, [18](#)  
  
me\_parNum, [18](#)  
me\_predict, [19](#)  
mirror\_na, [19](#)  
models, [20](#)  
  
na.approx, [12](#)  
  
orderfactor, [20](#)  
  
paste, [10](#)  
  
read.fwf, [21](#)  
read\_fwf\_fixedheader, [21](#)  
replace\_na, [21](#)  
resample\_meteo\_h2d, [22](#)  
rescale01, [23](#)  
  
tpr, [23](#)  
  
vpd, [24](#)  
  
without\_na, [24](#)  
  
yearplots, [25](#)