

Package ‘ClimMobTools’

November 9, 2021

Type Package

Title API Client for the 'ClimMob' Platform

Version 0.4.2

URL <https://agrdatasci.github.io/ClimMobTools/>

BugReports <https://github.com/agrdatasci/ClimMobTools/issues>

Description API client for 'ClimMob', an open source software for experimental crowdsourcing citizen science under the 'tricot' approach <<https://climmob.net/>>. Developed by van Etten et al. (2019) <[doi:10.1017/S0014479716000739](https://doi.org/10.1017/S0014479716000739)>, it turns the research paradigm on its head; instead of a few researchers designing complicated trials to compare several technologies in search of the best solutions for the target environment, it enables many participants to carry out reasonably simple experiments that taken together can offer even more information. 'ClimMobTools' enables project managers to deep explore and analyse their 'ClimMob' data in R.

License MIT + file LICENSE

Encoding UTF-8

Depends R (>= 3.5.0), climatrends, PlackettLuce

Imports httr, jsonlite, Matrix, methods, RSpecra, sf, utils

Suggests knitr, rmarkdown, testthat (>= 2.1.0)

Language en-GB

RoxygenNote 7.1.2

VignetteBuilder knitr

NeedsCompilation no

Author Kauê de Sousa [aut, cre] (<<https://orcid.org/0000-0002-7571-7845>>),
Jacob van Etten [aut] (<<https://orcid.org/0000-0001-7554-2558>>),
Brandon Madriz [ctb] (API Client implementation)

Maintainer Kauê de Sousa <kaue.desousa@inn.no>

Repository CRAN

Date/Publication 2021-11-09 19:20:02 UTC

R topics documented:

ClimMobTools	2
getDataCM	3
getProjectProgress	4
getProjectsCM	6
randomise	7
rankTricot	8
rmGeoIdentity	10
seedNeed	11

Index	12
--------------	-----------

ClimMobTools	<i>API Client for the 'ClimMob' platform in R</i>
--------------	---

Description

API client for 'ClimMob', an open source software for experimental crowdsourcing citizen science under the 'tricot' approach <<https://climmob.net/>>. Developed by van Etten et al. (2019) <[doi:10.1017/S0014479716000739](https://doi.org/10.1017/S0014479716000739)>, it turns the research paradigm on its head; instead of a few researchers designing complicated trials to compare several technologies in search of the best solutions for the target environment, it enables many participants to carry out reasonably simple experiments that taken together can offer even more information. 'ClimMobTools' enables project managers to deep explore and analyse their 'ClimMob' data in R.

Author(s)

Kauê de Sousa and Jacob van Etten and Brandon Madriz

See Also

Useful links:

- Development repository: <https://github.com/agrdatasci/ClimMobTools>
- Static documentation: <https://agrdatasci.github.io/ClimMobTools/>
- Report bugs: <https://github.com/agrdatasci/ClimMobTools/issues>
- ClimMob Platform: <https://climmob.net>
- The tricot user guide: <https://hdl.handle.net/10568/109942>

 getDataCM

Get ClimMob data

Description

Fetch the data from a ClimMob project using an application programming interface (API) key

Usage

```

getDataCM(
  key,
  project,
  as.data.frame = TRUE,
  as.text = FALSE,
  server = "climmob3",
  ...
)

## S3 method for class 'CM_list'
as.data.frame(x, ..., tidynames = TRUE, pivot.wider = FALSE)

```

Arguments

key	a character for the user's application programming interface (API) key
project	a character for the project id
as.data.frame	logical, to return a data frame
as.text	logical, to return a text file that can be parsed to json
server	optional, a character to select from which server the data will be retrieved. See details
...	additional arguments passed to methods
x	an object of class CM_list
tidynames	logical, TRUE make clean column names
pivot.wider	logical, if TRUE return a wider object where each tricot package is a row

Details

server: the default server is "climmob" used for clients of <https://climmob.net/climmob3/>, other options are:

"avisa" for clients of <https://avisa.climmob.net/>

"rtb" for clients of <https://rtb.climmob.net/>

"testing" for clients of <https://testing.climmob.net/climmob3/>

Value

An object of class 'CM_list' or a text file or a data.frame with class "CM_df" with the variables:

id	the participant's package id
moment	the data collection moment
variable	the variable name
value	the value for each variable

Author(s)

Kauê de Sousa

See Also

ClimMob website <https://climmbob.net/>

Other GET functions: [getProjectProgress\(\)](#), [getProjectsCM\(\)](#)

Examples

```
## Not run:  
  
# This function will not work without an API key  
# the user API key can be obtained once a free ClimMob account  
# is created via https://climmbob.net/  
  
my_key <- "add_your_key"  
my_project <- "my_climmbob_project"  
  
data <- getDataCM(key = my_key, project = my_project)  
  
## End(Not run)
```

getProjectProgress *Get project progress*

Description

Fetch the progress of a ClimMob project

Usage

```
getProjectProgress(key, project, server = "climmbob3", ...)
```

Arguments

key	a character for the user's application programming interface (API) key
project	a character with the id of one or more projects
server	optional, a character to select from which server the data will be retrieved. See details
...	additional arguments passed to methods. See details

Details

server: the default server is "climmbob" used for clients of <https://climmbob.net/climmbob3/>, other options are:

"avisa" for clients of <https://avisa.climmbob.net/>

"rtb" for clients of <https://rtb.climmbob.net/>

"testing" for clients of <https://testing.climmbob.net/climmbob3/>

Value

A data frame with the ClimMob projects

project_id	the project unique id
name	the project name
moment	either the design, registration or data collection
number_obs	number of observations collected in a given moment
last_activity	last activity of the given moment

Author(s)

Kauê de Sousa

See Also

ClimMob website <https://climmbob.net/>

Other GET functions: [getDataCM\(\)](#), [getProjectsCM\(\)](#)

Examples

```
## Not run:  
# This function will not work without an API key  
# the user API key can be obtained once a free ClimMob account  
# is created via https://climmbob.net/  
  
my_key <- "add_your_key"  
  
my_project <- "project_id"  
  
getProjectProgress(my_key, my_project)
```

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

```
getProjectsCM          Get ClimMob projects
```

Description

Fetch the status of ClimMob projects

Usage

```
getProjectsCM(key, server = "climmob3", ...)
```

Arguments

key	a character for the user's application programming interface (API) key
server	optional, a character to select from which server the data will be retrieved. See details
...	additional arguments passed to methods. See details

Details

server: the default server is "climmob" used for clients of <https://climmob.net/climmob3/>, other options are:

"avisa" for clients of <https://avisa.climmob.net/>

"rtb" for clients of <https://rtb.climmob.net/>

"testing" for clients of <https://testing.climmob.net/climmob3/>

Value

A data frame with the ClimMob projects

project_id	the project unique id
name	the project name
country	ISO code for the country where the project was implemented
status	the current status
creation_date	the project's creation date
intended_participants	the number of participants the project intended to register
intended_participants	the number of participants the project intended to register
registered_participants	the number of participants registered
last_registration_activity	number of days since the submission of the last registration

Author(s)

Kauê de Sousa

See Also

ClimMob website <https://climmob.net/>

Other GET functions: [getDataCM\(\)](#), [getProjectProgress\(\)](#)

Examples

```
## Not run:
# This function will not work without an API key
# the user API key can be obtained once a free ClimMob account
# is created via https://climmob.net/

my_key <- "add_your_key"

getProjectsCM(key = my_key)

## End(Not run)
```

randomise

Randomised group of items

Description

Set a randomised group of items for crowdsourcing citizen science. Generate designs for ranking of options. It is designed for tricot trials specifically (comparing 3 options), but it will also work with comparisons of any other number of options. The design strives for approximate A optimality, this means that it is robust to missing observations. It also strives for balance for positions of each option. Options are equally divided between first, second, third, etc. position. The strategy is to create a "pool" of combinations that does not repeat combinations and is A-optimal. Then this pool is ordered to make subsets of consecutive combinations also relatively balanced and A-optimal

Usage

```
randomise(
  npackages,
  itemnames,
  ncomp = 3,
  availability = NULL,
  proportions = NULL,
  ...
)
```

Arguments

<code>npackages</code>	an integer for the number of trial packages to be produced
<code>itemnames</code>	a character for the name of items tested in the project
<code>ncomp</code>	an integer for the number of items to be assigned to each package
<code>availability</code>	optional, a vector with integers indicating the number of packages available for each <i>itemnames</i>
<code>proportions</code>	optional, a numeric vector with the desired proportions for each <i>itemnames</i>
<code>...</code>	additional arguments passed to methods

Value

A dataframe with the randomised design

Author(s)

Jacob van Etten

Examples

```
ncomp <- 3
npackages <- 100
itemnames <- c("apple","banana","grape","mango", "orange", "kiwi", "pineapple")
availability <- c(50, 50, 150, 150, 150, 150, 20)

table(unlist(randomise(ncomp = ncomp,
                      npackages = npackages,
                      itemnames = itemnames)))

table(unlist(randomise(ncomp = ncomp,
                      npackages = npackages,
                      itemnames = itemnames,
                      availability = availability)))
```

rankTricot

Build Plackett-Luce rankings from tricot dataset

Description

Create an object of class "rankings" from tricot data.

Usage

```
rankTricot(data, items, input, group = FALSE, additional.rank = NULL, ...)
```


Arguments

data	a data.frame with columns specified by items and input values
items	a character or numerical vector for indexing the column(s) containing the item names in data
input	a character or numerical vector for indexing the column(s) containing the values in data to be ranked
group	logical, if TRUE return an object of class "grouped_rankings"
additional.rank	optional, a data frame for the comparisons between tricot items and the local item
...	additional arguments passed to methods. See details

Details

full.output: logical, to return a list with a "rankings", a "grouped_rankings" and the ordered items

Value

a PlackettLuce "rankings" object, which is a matrix of dense rankings

Author(s)

Kauê de Sousa and Jacob van Etten, with ideas from Heather Turner

References

van Etten J., et al. (2019). Experimental Agriculture, 55(S1), 275–296. doi: [10.1017/S0014479716000739](https://doi.org/10.1017/S0014479716000739)

See Also

[rankings](#)

Examples

```
# beans data where each observer compares 3 varieties randomly distributed
# from a list of 11 and additionally compares these 3 varieties
# with their local variety
library("PlackettLuce")
data("beans", package = "PlackettLuce")

# first build rankings with only tricot items
# and return an object of class 'rankings'
R <- rankTricot(data = beans,
                items = c(1:3),
                input = c(4:5))

head(R)

#####
```

```

# pass the comparison with local item as an additional rankings, then
# each of the 3 varieties are compared separately with the local item
# and return an object of class grouped_rankings
G <- rankTricot(data = beans,
               items = c(1:3),
               input = c(4:5),
               group = TRUE,
               additional.rank = beans[c(6:8)])

head(G)

```

rmGeoIdentity	<i>Remove geographical identity</i>
---------------	-------------------------------------

Description

Build a buffer around the a set of geographical coordinates and take a random point around the buffer. The function is used to omit the precise location of tricot participants but keeping a close distance to its agro-environment

Usage

```
rmGeoIdentity(lonlat, dist = 0.015, nQuadSegs = 2L, ...)
```

Arguments

lonlat	a data.frame or matrix with geographical coordinates long lat
dist	numeric, buffer distance for all <i>lonlat</i>
nQuadSegs	integer, number of segments per quadrant
...	further arguments passed to sf methods

Examples

```

xy <- matrix(c(11.097799, 60.801090,
              11.161298, 60.804199,
              11.254428, 60.822457),
            nrow = 3, ncol = 2, byrow = TRUE)

rmGeoIdentity(xy)

# the function handles NAs by keeping then
# in a logic vector to reconstruct the matrix
xy2 <- matrix(c(11.097799, 60.801090,
               NA, NA,
               11.161298, 60.804199,
               11.254428, 60.822457,
               11.254428, NA),

```

```
nrow = 5, ncol = 2, byrow = TRUE)

rmGeoIdentity(xy2)
```

seedNeed	<i>Required seed amount in a tricot project</i>
----------	---

Description

Calculate the required amount of seeds (or other technology) required for a triadic comparison of technologies (tricot) project.

Usage

```
seedNeed(npackages = 100, ncomp = 3, nitems = 10, nseeds = 0.15, unit = "kg")
```

Arguments

npackages	an integer for the number of trial packages to be produced
ncomp	an integer for the number of items to be assigned to each package
nitems	number of items tested in the project
nseeds	an integer for the metric of seeds each bag receives
unit	optional, a character specifying the metric unit used

Value

a dataframe with required number of seeds

Author(s)

Kauê de Sousa

Examples

```
# allocate 0.2 kg of seeds per variety in a project with 500
# participants and 14 varieties
seedNeed(npackages = 500,
         ncomp = 3,
         nitems = 14,
         nseeds = 0.2)

# allocate 100 seedlings per variety in a project with 400
# participants, 8 varieties and 3 comparisons between varieties
seedNeed(npackages = 400,
         ncomp = 3,
         nitems = 9,
         nseeds = 100,
         unit = "unit")
```

Index

* GET functions

getDataCM, [3](#)

getProjectProgress, [4](#)

getProjectsCM, [6](#)

as.data.frame.CM_list (getDataCM), [3](#)

ClimMobTools, [2](#)

ClimMobTools-package (ClimMobTools), [2](#)

getDataCM, [3](#), [5](#), [7](#)

getProjectProgress, [4](#), [4](#), [7](#)

getProjectsCM, [4](#), [5](#), [6](#)

randomise, [7](#)

randomize (randomise), [7](#)

rankings, [9](#)

rankTricot, [8](#)

rmGeoIdentity, [10](#)

seedNeed, [11](#)

sf, [10](#)