Package 'bdclean'

April 11, 2019

Type Package

Title A User-Friendly Biodiversity Data Cleaning App for the Inexperienced R User

Description Provides features to manage the complete workflow for biodiversity data cleaning. Uploading data, gathering input from users (in order to adjust cleaning procedures), cleaning data and finally, generating various reports and several versions of the data. Facilitates user-level data cleaning, designed for the inexperienced R user. T Gueta et al (2018) <doi:10.3897/biss.2.25564>. T Gueta et al (2017) <doi:10.3897/tdwgproceedings.1.2031

Version 0.1.15

Date 2019-04-10

License GPL-3

URL https://github.com/bd-R/bdclean, https://bd-r.github.io/The-bdverse/index.html

BugReports https://github.com/bd-R/bdclean/issues

Maintainer Thiloshon Nagarajah <thiloshon@gmail.com>

Imports rmarkdown, knitr, shiny, shinydashboard, shinyjs, leaflet, DT, data.table, rgbif, spocc, finch, bdDwC, bdchecks, methods, tools

Depends R (>= 2.10)

RoxygenNote 6.1.1

Suggests testthat, roxygen2, covr

LazyData true

NeedsCompilation no

Author Thiloshon Nagarajah [aut, cre], Tomer Gueta [aut] (<https://orcid.org/0000-0003-1557-8596>), Vijay Barve [aut] (<https://orcid.org/0000-0002-4852-2567>), Ashwin Agrawal [aut], Povilas Gibas [aut] (<https://orcid.org/0000-0001-5311-6021>), Yohay Carmel [aut] (<https://orcid.org/0000-0002-5883-0184>)

Repository CRAN

Date/Publication 2019-04-11 14:45:17 UTC

R topics documented:

bdclean . . . 2
BdQuestion-class . . . 3
BdQuestionContainer-class . . . 3
cleaning_function . . . 3
clean_data . . . 4
create_default_questionnaire . . . 5
create_report_data . . . 6
earliest_date . . . 7
get_checks_list . . . 8
get_user_response . . . 8
perform_Cleaning . . . 9
run_bdclean . . . 9
run_questionnaire . . . 10
spatial_resolution . . . 11
taxo_level . . . 12
temporal_resolution . . . 13

Index **14**

`bdclean` *bdclean: Biodiversity Data Cleaning Workflows.*

Description

Biodiversity Data Cleaning Workflows using R would be helpful to clean biodiversity occurrence data typically downloaded from Global Biodiversity Information Facility (http://www.gbif.org/) or similar biodiversity data portals. There are several data cleaning operations needed to be performed on most of the data downloaded, in order to achieve minimum quality to use the data further for any analysis or modelling.

Data cleaning

- run_bdclean
- clean_data

Citation

- Gueta, T., Barve, V., Nagarajah, T., Agrawal, A. & Carmel, Y. (2019). bdclean: Biodiversity data cleaning workflows (R package V 0.1.13). Retrieved from https://github.com/bd-R/bdclean/

BdQuestion-class *The Question Reference Class*

Description

The Question Reference Class

BdQuestionContainer-class *The Question Container Reference Class*

Description

The Question Container Reference Class

Methods

`initialize(bdquestions = NA)` Construct an instance of BdQuestionContainer after validating the type.

cleaning_function *Data decision function (binary decision) required in bdclean internal usage.*

Description

NOTE: This is an package internal function. Do not use for external uses. Exported to make it available for shiny app.

Usage

```
cleaning_function(bddata)
```

Arguments

bddata	The dataframe to clean

Examples

```
if(interactive()){

library(rgbif)
occdat <- occ_data(
  country = 'AU', # Country code for australia
  classKey = 359, # Class code for mammalia
  limit = 50 # Get only 50 records
)
myData <- occdat$data
cleaned_data <- cleaning_function(myData)

}
```

`clean_data` *Data cleaning according to Questionnaire Responses.*

Description

Use `run_questionnaire` to add Questionnaire Responses and pass it to this function to process the data faster.

Usage

```
clean_data(data, custom_questionnaire = NULL, clean = TRUE,
  missing = FALSE, report = TRUE, format = c("html_document",
  "pdf_document"))
```

Arguments

`data`	Biodiversity data in a data frame
`custom_questionnaire`	Custom user created questionnaire responses if to pypass answering questions each time.
`clean`	Whether to clean after flagging. If false only flagging will be done.
`missing`	How to treat data with missing values. Default: false - will be treated as bad data.
`report`	Whether to print report of cleaning done.
`format`	Formats of the cleaning report required. Options are: Markdown, HTML or / and PDF

Details

Use `create_default_questionnaire` to create default questionnaire object. You can add your custom questions to this questionnaire and then pass it to this function to process the data.

Value

data frame with clean data

Examples

```
custom_questionnaire <- create_default_questionnaire()

if(interactive()){

library(rgbif)
occdat <- occ_data(
  country = 'AU', # Country code for australia
  classKey = 359, # Class code for mammalia
  limit = 50 # Get only 50 records
)
myData <- occdat$data

responses <- run_questionnaire()
cleaned_data <- clean_data(myData, responses)

cleaned_data2 <- clean_data(myData)

}
```

---

create_default_questionnaire

Create the package default Questionnaire.

Description

Create the package default Questionnaire.

Usage

```
create_default_questionnaire()
```

Value

BdQuestionContainer object with default Questions

Examples

```
customQuestionnaire <- create_default_questionnaire()
```

`create_report_data` *Generate data required to create report, function required in bdclean internal usage.*

Description

NOTE: This is an package internal function. Do not use for external uses. Exported to make it available for shiny app.

Usage

```
create_report_data(input_data, flagged_data, cleaned_data, responses,
  cleaning_true, format)
```

Arguments

`input_data`	The input dataframe before cleaning
`flagged_data`	The flagged data for cleaning
`cleaned_data`	The data with flagged records removed
`responses`	The BDQuestions object with user responses
`cleaning_true`	Flag specifying if the cleaning should be done, or just flagging
`format`	The format of the report to be generated

Examples

```
if(interactive()){

library(rgbif)
occdat <- occ_data(
  country = 'AU', # Country code for australia
  classKey = 359, # Class code for mammalia
  limit = 50 # Get only 50 records
)
myData <- occdat$data

question <- BdQuestion()
responses <- get_user_response(question)

cleaned_data <- create_report_data(myData, myData, myData, responses, T, 'pdf')

}
```

`earliest_date` *Clean data based on earliest date.*

Description

Clean data based on earliest date.

Usage

```
earliest_date(bddata, res = "1700-01-01")
```

Arguments

`bddata`	Bio diversity data in a data frame
`res`	The earliest data required

samplePassData

When resolution is 20-Jan-2005, records recorded after the date will pass.

sampleFailData

When resolution is 20-Jan-2005, records recorded before the date will fail.

targetDWCField

eventDate

checkCategory

temporal

Examples

```
if(interactive()){

library(rgbif)
occdat <- occ_data(
  country = 'AU', # Country code for australia
  classKey = 359, # Class code for mammalia
  limit = 50 # Get only 50 records
)
myData <- occdat$data

responses <- earliest_date(myData, '2000-01-01')

}
```

get_checks_list *Returning checks list, function required in bdclean internal usage.*

Description

NOTE: This is an package internal function. Do not use for external uses.

Usage

```
get_checks_list()
```

Examples

```
if(interactive()){

all_checks <- get_checks_list()

}
```

get_user_response *Internal function for getting user response*

Description

Internal function for getting user response

Usage

```
get_user_response(bd_question)
```

Arguments

bd_question	The BDQuestion object to get users responses.

Examples

```
if(interactive()){

question <- BdQuestion()
responses <- get_user_response(question)

}
```

perform_Cleaning *Data decision function (threshold tuning) required in bdclean internal usage.*

Description

NOTE: This is an package internal function. Do not use for external uses.

Usage

```
perform_Cleaning(flagged_data, cleaning_threshold = 5)
```

Arguments

flagged_data	The dataset with flags to be cleaned.
cleaning_threshold	The Cleaning tolerance. Not used in current version.

Examples

```
if(interactive()){

library(rgbif)
occdat <- occ_data(
  country = 'AU', # Country code for australia
  classKey = 359, # Class code for mammalia
  limit = 50 # Get only 50 records
)
myData <- occdat$data
cleaned_data <- perform_Cleaning(myData)

}
```

run_bdclean *Launch bdclean Shiny Application*

Description

Launch bdclean Shiny Application

Usage

```
run_bdclean()
```

Examples

```
if(interactive()){

run_bdclean()

}
```

`run_questionnaire` *Execute the Questionnaire and save user responses.*

Description

Execute the Questionnaire and save user responses.

Usage

```
run_questionnaire(custom_questionnaire = NULL)
```

Arguments

`custom_questionnaire`
: Custom User Created Questionnaire if already available.

Value

list with BdQuestionObjects containing user answers

Examples

```
if(interactive()){

responses <- run_questionnaire()

}
```

`spatial_resolution` *Clean data based on spatial resolution*

Description

Clean data based on spatial resolution

Usage

```
spatial_resolution(bddata, res = 100)
```

Arguments

`bddata`	Bio diversity data in a data frame
`res`	The highest coordinate uncertainty required

samplePassData

When resolution is 100 meters, Coordinate Uncertainties below 100 meters will pass.

sampleFailData

When resolution is 100 meters, Coordinate Uncertainties above 100 meters will fail.

targetDWCField

coordinateUncertaintyInMeters

checkCategory

spatial

Examples

```
if(interactive()){

library(rgbif)
occdat <- occ_data(
  country = 'AU', # Country code for australia
  classKey = 359, # Class code for mammalia
  limit = 50 # Get only 50 records
)
myData <- occdat$data

responses <- spatial_resolution(myData, 1500)

}
```

`taxo_level` *Clean data based on lower taxon level*

Description

Clean data based on lower taxon level

Usage

```
taxo_level(bddata, res = "SPECIES")
```

Arguments

`bddata`	Bio diversity data in a data frame
`res`	The low rank of species required

samplePassData

When resolution is Species, Subspecies and Species will pass.

sampleFailData

When resolution is Species, Family or Genus or any lower ranks will fail.

targetDWCField

taxonRank

checkCategory

taxonomic

Examples

```
if(interactive()){

library(rgbif)
occdat <- occ_data(
  country = 'AU', # Country code for australia
  classKey = 359, # Class code for mammalia
  limit = 50 # Get only 50 records
)
myData <- occdat$data

responses <- taxo_level(myData, 'SPECIES')

}
```

`temporal_resolution` *Clean data based on temporal resolution*

Description

Clean data based on temporal resolution

Usage

```
temporal_resolution(bddata, res = "Day")
```

Arguments

`bddata`	Bio diversity data in a data frame
`res`	restriction of records with/without data, month, year fields

samplePassData

When resolution is day, records with day specified will pass.

sampleFailData

When resolution is month, records with NA/empty month specified will fail.

targetDWCField

day, month, year

checkCategory

temporal

Examples

```
if(interactive()){

library(rgbif)
occdat <- occ_data(
  country = 'AU', # Country code for australia
  classKey = 359, # Class code for mammalia
  limit = 50 # Get only 50 records
)
myData <- occdat$data

responses <- taxo_level(temporal_resolution, 'Day')

}
```

Index

bdclean, 2
bdclean-package *(*bdclean*)*, 2
BdQuestion *(*BdQuestion-class*)*, 3
BdQuestion-class, 3
BdQuestionContainer
*(*BdQuestionContainer-class*)*, 3
BdQuestionContainer-class, 3

clean_data, *2*, 4
cleaning_function, 3
create_default_questionnaire, 5
create_report_data, 6

earliest_date, 7

get_checks_list, 8
get_user_response, 8

perform_Cleaning, 9

run_bdclean, *2*, 9
run_questionnaire, 10

spatial_resolution, 11

taxo_level, 12
temporal_resolution, 13