Package: popler (via r-universe)

October 28, 2024

Title Popler R Package

Version 0.2.0

Description Browse and query the popler database.

URL https://github.com/ropensci/popler, https://docs.ropensci.org/popler/

BugReports https://github.com/ropensci/popler/issues

Depends R (>= 3.5.0)

Imports crul (>= 0.7.4), dbplyr (>= 1.1.0), dplyr (>= 0.5.0), ggplot2 (>= 3.0.0), grid (>= 3.3.0), jsonlite (>= 1.5), lazyeval (>= 0.2.0), magrittr (>= 1.5.0), purrr (>= 0.2.5), rlang (>= 0.2.1), rmarkdown (>= 1.2), stats (>= 3.3.0), stringr (>= 1.2.0), tibble (>= 1.4.2), tidyr (>= 1.0.0), utils (>= 3.3.0)

License MIT + file LICENSE

Encoding UTF-8

Suggests fs (>= 1.2.2), knitr (>= 1.15.1), maps (>= 3.3.0), mapproj (>= 1.2.6), testthat(>= 2.0.0), covr (>= 3.1.0), stubthat (>= 1.2.1), mockr (>= 0.1)

LazyData true

RoxygenNote 6.1.1

VignetteBuilder knitr

Repository https://ropensci.r-universe.dev

RemoteUrl https://github.com/ropensci/popler

RemoteRef master

RemoteSha 65fd5a87f24322b64a7c1f9edeb805bfa8935a89

Contents

`filter.browse` *Methods for dplyr verbs*

Description

Subsets a `browse` or `get_data` object based on logical statements.

Usage

```
## S3 method for class 'browse'
filter(.data, ...)

## S3 method for class 'get_data'
filter(.data, ...)
```

Arguments

`.data`	A `browse` or `get_data` object
`...`	logical conditions

mutate.browse *Methods for dplyr verbs*

Description

Add new columns to a browse or get_data object.

Usage

```
## S3 method for class 'browse'
mutate(.data, ...)

## S3 method for class 'get_data'
mutate(.data, ...)
```

Arguments

.data	A browse or get_data object
...	Name-value pairs of expressions. Use NULL to drop a variable.

popler *popler package*

Description

popler is a package for interacting with the PostgreSQL data base with the same name. popler contains data on long-term population dynamics from the LTER network. Every exported function is prefixed with pplr_ and then a verb (e.g. pplr_get_data() or noun (e.g. pplr_citation. Accessing popler does not require an API key, all you need is an internet connection and you are ready to go!

Authors

Aldo Compagnoni <aldo.compagnoni@aggiemail.usu.edu>

Andrew Bibian <ajbibian@rice.edu>

Brad Ochocki <brad.ochocki@rice.edu>

Sam Levin <sam.levin@idiv.de>

Tom Miller <tom.miller@rice.edu>

pplr_browse *Browse the metadata of projects contained in the popler database*

Description

pplr_browse() reports the metadata of LTER studies contained in the popler database. The user can subset which datasets, and which metadata variables to visualize.

Usage

```
pplr_browse(..., full_tbl = FALSE, vars = NULL, view = FALSE,
  keyword = NULL, report = FALSE)
```

Arguments

...	A logical expression to subset the table containing the metadata of datasets contained in popler
full_tbl	logical; Should the function returns the standard columns, or the full main table? Default is FALSE.
vars	A vector of characters in case the user want to select which variables of popler's main table should be selected?
view	If TRUE, opens up a spreadsheet-style data viewer.
keyword	A string used to select individual datasets based on pattern matching. The string is matched to every string element in the variables of the metadata table in popler.
report	logical; If TRUE, function produces a markdown report about each study's metadata, and opens it as a html page. Default is FALSE.

Value

A data frame combining the metadata of each project and the taxonomic units associated with each project.

This data frame is of class popler, data.frame, tbl_df, and tbl.

Examples

```
## Not run:
# No arguments return the standard 16 columns of popler's main table
default_vars = pplr_browse()

# full_tbl = TRUE returns the full table
all_vars = pplr_browse(full_tbl = TRUE)

# subset only data from the sevilleta LTER, and open the relative report in a html page
sev_data = pplr_browse(lterid == "SEV", report = TRUE)
```

```
# consider only plant data sets
plant_data = pplr_browse(kingdom == "Plantae")

# Select only the data you need
three_columns = pplr_browse(vars = c("title","proj_metadata_key","genus","species"))

# Select only the data you need
study_21 = pplr_browse( proj_metadata_key == 25)

# Select studies that contain word "parasite"
parasite_studies = pplr_browse( keyword = "parasite")

## End(Not run)
```

`pplr_citation` *Provide citations for a popler object returned by* `pplr_browse` *or* `pplr_get_data` *object.*

Description

Returns a bibliography, Bibtex citations, and acknowledgment template.

Usage

```
pplr_citation(input, bibtex_path = NULL)
```

Arguments

`input`	An object of class `browse` or `get_data`.
`bibtex_path`	Specify the filename and location for the generated markdown file (optional).

Value

A list of references from `input`.

Examples

```
## Not run:
# make a browse object
metadata <- pplr_browse(proj_metadata_key %in% c(17, 317, 494))

# cite the projects
cite <- pplr_citation(metadata)

# cite$bibliography          # the bibliography
# cite$Bibtex                # Bibtex entries for each dataset
# cite$acknowledgement       # acknowledgement template

## End(Not run)
```

pplr_cov_unpack *Unpack the covariates contained in the dataset downloaded via* pplr_get_data()

Description

Create a data frame by "extracting" the covariates column contained in an dataset downloaded with pplr_get_data().

Usage

```
pplr_cov_unpack(input)
```

Arguments

input	An object of class get_data.

Value

A data frame whose columns represent the covariates of the dataset downloaded via pplr_get_data(). Note that these covariates are contained in the covariates column datasets downloaded using pplr_get_data().

Examples

```
## Not run:
library(dplyr)
demo_d <- pplr_get_data(proj_metadata_key == 8)
as.tbl( pplr_cov_unpack( demo_d ) )

## End(Not run)
```

pplr_dictionary *Dictionary of the popler metadata variables*

Description

Describes the metadata variables contained in the popler database, and shows their content.

Usage

```
pplr_dictionary(..., full_tbl = FALSE)
```

Arguments

...	A sequence of (unquoted) variables specifying one or more variables of popler's main table for which dictionary information is needed
full_tbl	logical; If TRUE, the function returns a table describing the variables of the full main table. If FALSE, the function returns a table describing the standard variables. Default is FALSE.

Examples

```
## Not run:
# Column names
column_names <- pplr_dictionary(full_tbl = FALSE)

# Dictionary information
dictionary_lter <- pplr_dictionary(lterid, full_tbl = FALSE)

# multiple columns
dictionary_lter_lat <- pplr_dictionary(lterid,lat_lter, full_tbl = FALSE)

## End(Not run)
```

---

pplr_get_data *Download data from the popler database*

Description

This function downloads datasets contained in the popler database. The user can download data directly, using a logical expression, or indirectly, using objects created by pplr_browse.

Usage

```
pplr_get_data(..., cov_unpack = FALSE)
```

Arguments

...	An object produced by pplr_browse or a logical expression.
cov_unpack	logical; if TRUE, function pplr_cov_unpack is applied to the variable covariates of the downloaded dataset in order to extract the variables contained in therein and combine the new columns with the default output. Default is FALSE.

Details

. By default, the following variables are included when a user calls pplr_get_data().

- authors
- authors_contact
- year

- day
- month
- sppcode
- genus
- species
- datatype
- spatial_replication_level_1_label
- spatial_replication_level_1
- spatial_replication_level_2_label
- spatial_replication_level_2
- spatial_replication_level_3_label
- spatial_replication_level_3
- spatial_replication_level_4_label
- spatial_replication_level_4
- spatial_replication_level_5_label
- spatial_replication_level_5
- proj_metadata_key
- structure_type_1
- structure_type_2
- structure_type_3
- structure_type_4
- treatment_type_1
- treatment_type_2
- treatment_type_3
- covariates

Value

This data fame is of class `get_data`, and `data.frame`.

Examples

```
## Not run:
# browse a study, then get the data associated with it
parasite = pplr_browse(proj_metadata_key == 25)
gh_data = pplr_get_data(parasite)

# insect data sets from the SEV lter site
insect_sev = pplr_browse(class == "Insecta" & lterid == "SEV")
insect_25_yrs96_99 = pplr_get_data(insect_sev)

insect_21_25 = pplr_get_data( (proj_metadata_key == 43 |
                               proj_metadata_key == 25) )

## End(Not run)
```

pplr_maps *Generate maps of LTER sites*

Description

Generates maps of LTER sites in a given input object. Sizes of site markers correspond to the number of studies at a given site.

Usage

```
pplr_maps(input, return_plot = FALSE)
```

Arguments

input	An object created by either pplr_browse() or pplr_get_data()
return_plot	logical; if TRUE function returns the ggplot object for subsequent modification. If FALSE, function returns an invisible copy of the input object (useful for piping). Default is FALSE.

Value

The input object (invisibly) or a ggplot2 object.

Examples

```
## Not run:

library(dplyr) # make %>% available

browse_object <- pplr_browse(proj_metadata_key == 11)

browse_object %>%
  pplr_maps()

## End(Not run)
```

pplr_metadata_url *Get metadata information from a data object*

Description

Load the webpage containing the metadata of the data sets contained in objects produced by pplr_browse or downloaded through pplr_get_data(). If you downloaded data from multiple projects, this function will open multiple webpages. This is a wrapper of function browseURL in base.

Usage

```
pplr_metadata_url(input)
```

Arguments

input	An object produced by the function pplr_get_data().

Examples

```
## Not run:
# Load the metadata webpages of the projects that contain data from the Poa genus.
fes_d <- pplr_browse(genus == "Festuca")
pplr_metadata_url( fes_d )

## End(Not run)
```

---

pplr_report_dictionary

A user-friendly dictionary of the popler metadata

Description

Provides information on the variables of metadata contained in the popler database, and the kind of data contained in those variables.

Usage

```
pplr_report_dictionary(full_tbl = FALSE, md_file = NULL,
  html_file = NULL)
```

Arguments

full_tbl	logical; if TRUE function returns the variables contained in the full main table. If FALSE, functions returns only the standard variables. Default is FALSE.
md_file	Specify the filename and location for the generated markdown file (optional)
html_file	Specify the filename and location for the generated html file (optional)

Value

This function is called for its side effects and does not return an R object.

Examples

```
## Not run:
# Full dictionary
pplr_report_dictionary(full_tbl = TRUE)

# "Abridged" version
pplr_report_dictionary()

## End(Not run)
```

`pplr_report_metadata` *Open a report of the metadata of project(s) as an html page*

Description

Generates a readable report of the metadata describing data sets contained in popler. The report contains citations, the links to the original URL of each data set, and example code to obtain the metadata and data of the projects represented in the html page.

Usage

```
pplr_report_metadata(input, md_file = "./browse.Rmd",
  html_file = "./browse.html")
```

Arguments

`input`	A popler object returned by `pplr_browse()` or `pplr_get_data()`
`md_file`	Specify the filename and location for the generated markdown file (optional)
`html_file`	Specify the filename and location for the generated html file (optional)

Value

An invisible copy of `input`.

Examples

```
## Not run:
# Full dictionary
one_spp <- pplr_browse(community == "no" & duration_years > 15)
pplr_report_metadata(one_spp)

data <- pplr_get_data(one_spp)
pplr_report_metadata(data) # same as above

## End(Not run)
```

pplr_search *search*

Description

search

Usage

```
pplr_search(proj_metadata_key, limit = 10, offset = 0, ...)
```

Arguments

proj_metadata_key	project metadata key
limit	number of records to return, default: 10
offset	record number to start at, default: first record
...	curl options passed on to [crul::HttpClient]

Examples

```
# basic example
pplr_search(proj_metadata_key = 13)
# pass in curl options for debugging, seeing http request details
pplr_search(proj_metadata_key = 13, verbose = TRUE)
```

pplr_site_rep *Spatial-temporal replication of data sets*

Description

Functions to examine the number of temporal replicates contained within each spatial replication level of a dataset. `pplr_site_rep_plot` plots the temporal replicates available for each site. `pplr_site_rep` produces logical vectors that identify the spatial replicates with enough temporal replication, or summary tables.

Usage

```
pplr_site_rep(input, freq = 1, duration = 10, rep_level = 1,
  return_logical = TRUE)

pplr_site_rep_plot(input, return_plot = FALSE)
```

Arguments

input	An object of produced by pplr_get_data. Note that this is not an output from pplr_browse, as the raw data is required to calculate the amount of replication.
freq	A number corresponding to the desired annual frequency of replicates. Studies that are replicated more frequently will be included in the counts and those that replicated less frequently will be excluded. If return_logical = TRUE, rows that contain information from sites that are replicated at the desired frequency will have a TRUE value, and rows that are not will have a FALSE value. Values greater than 1 will select sampling done multiple times per year. For example, freq = 2 indicates a desired sampling frequency of 6 months. Conversely, freq = 0.5 indicates a desired sampling done once every 2 years.
duration	An integer corresponding to the desired number of yearly replicates. Rows containing site information from sites with more replication will be included, while those with less will be excluded.
rep_level	An integer corresponding to the level of spatial replication over which verify yearly temporal replication. Values between 1 and 5 are possible (though higher levels may not be present for some datasets). Higher values correspond to higher levels of spatial nestedness. The default value of rep_level = 1 corresponds to sites.
return_logical	logical; if TRUE, the function returns a logical vector. This vector can be used to subset the dataset. If FALSE, the function returns a summary table of class tbl. This table shows, in variable number_of_samples, how many temporal replicates per year are contained by each spatial replicate. Default is TRUE.
return_plot	A logical indicating whether to return a copy of the input data or the ggplot object created by the function. Use TRUE to return the ggplot object for subsequent modification. Use FALSE to return an invisible copy of the input object (useful for piping). Default is FALSE.

Details

pplr_site_rep_plot produces a scatterplot showing the sites (spatial_replication_level_1) and years for which data is available.

pplr_site_rep works with any level of spatial replication and produces either a summary table of temporal replication or a logical vector that can be used to subset a data set based on the desired frequency and length of time.

Value

pplr_site_rep_plot: input object (invisibly) or a ggplot2 object. Use return_plot to control.

pplr_site_rep: A tbl or a logical vector of length dim(input)[1]. Use return_logical to control.

Examples

```
## Not run:

library(ggplot2)
```

```
library(dplyr)

# produce logical vector and subset using it. This can also be piped into a
# the plotting function for visiualization

good_studies <- pplr_get_data(lterid == 'SEV') %>%
                  .[pplr_site_rep(input = .,
                                  duration = 12,
                                  rep_level = 3), ] %>%
                  pplr_site_rep_plot()


# Or, make a neat summary table and decide where to go from there
SEV <- pplr_get_data(lterid == 'SEV')

rep_table <- pplr_site_rep(input = SEV,
                           freq = 0.5,
                           duration = 12,
                           return_logical = FALSE)

# pplr_site_rep_plot ---------------

# create an unmodified figure
BNZ <- pplr_get_data(lterid == 'BNZ')

pplr_site_rep_plot(BNZ)

# Return the figure instead of the data for subsequent modification
Antarctica <- pplr_get_data(lterid == 'PAL')

pplr_site_rep_plot(Antarctica,
              return_plot = TRUE) +
   ggtitle("Penguins Rock!")

# Use within pipes. Cannot return and modify the figure this way.
pplr_get_data(lterid == 'SEV') %>%
  pplr_site_rep_plot(return_plot = FALSE) %>%
  pplr_report_metadata()

## End(Not run)
```

---

pplr_summary *search*

Description

search

Usage

```
pplr_summary(limit = 10, offset = 0, ...)
```

Arguments

limit	number of records to return, default: 10
offset	record number to start at, default: first record
...	curl options passed on to [crul::HttpClient]

Examples

```
# basic example
pplr_summary()
# pass in curl options for debugging, seeing http request details
pplr_summary(verbose = TRUE)
```

---

pplr_summary_table_update

Update popler*'s summary table*

Description

Automatically retrieve most up to date version of popler summary table

Usage

```
pplr_summary_table_update()
```

Value

This function is called for its side effect and does not return anything

Note

The summary_table is often called internally by popler functions, but can also be accessed directly by calling pplr_summary_table_import().

Index