

standardizeSnapshot

December 9, 2022

<code>create_logger</code>	<i>Create logger</i>
----------------------------	----------------------

Description

Initializes a logger with a given logfile.

Usage

```
create_logger(my_logfile)
```

Arguments

`my_logfile` path to the log file (character).

Value

Returns a logger and creates a logfile at the given path. If the path given does not exist, also creates this path.

<code>fill_capture_info</code>	<i>Fill capture info</i>
--------------------------------	--------------------------

Description

Fills different columns related to capture info.

Usage

```
fill_capture_info(df, classifier = c("zooniverse", "traptagger", "digikam"))
```

Arguments

- df a dataframe for which columns should be filled.
- classifier The classifier

Value

A dataframe with values filled for locationID, cameraID, roll and capture. Also fills season if Zooniverse.

get_cameraID	<i>Get camera ID</i>
--------------	----------------------

Description

Returns the complete cameraID as locationID_camera.

Usage

```
get_cameraID(locationID, camera, classifier)
```

Arguments

- locationID the location code vector (usually 3 letters)
- camera the "old" camera code vector, which is expected not to contain the locationID already. In case it does, then it will not be added and give a message.
- classifier The classifier used. If it is traptagger, will not display message because it is expected that cameraID will already be locationID_camera. It is optional (if not specified, it will display the message by default)

Value

a vector of same lengths as locationID and camera with pasted locationID_camera. If camera is already in format locationID_camera, then it does not changes and displays a message.

get_csv_files_and_folders	
	<i>Get files and folder</i>

Description

Get all csv files from the input character vector. If 'input' is a folder, will list all files within input; if 'input' is a file, will only list this file.

Usage

```
get_csv_files_and_folders(input, except)
```

Arguments

input	a character vector of valid paths: can be files or folders, or a mix of both
except	files to ignore (optional)

Value

A dataframe with columns folder and files, where folder is the folder up to a given file and file is the file.

get_eventID	<i>Get eventID</i>
-------------	--------------------

Description

Get the eventID from the capture info. All vectors must be the same length or they will be recycled.

Usage

```
get_eventID(locationID, cameraID, roll, captureID)
```

Arguments

locationID	character vector of location
cameraID	character vector of cam_site
roll	character vector of roll
captureID	character vector of captureID

Value

A character vector of the same length of the inputs (or the longest input) with fields formatted as season#cam_site#roll#event_no

get_final_filename	<i>Get final filemane</i>
--------------------	---------------------------

Description

Return the filename from the file columns.

Usage

```
get_final_filename(df)
```

Arguments

df	The dataframe to be copied. Must have columns locationID, season, roll.
----	-------------------------------------------------------------------------

Value

The filename for this file in the format locationID_Sseason_Rroll.csv It there are several locationID, seasons or rolls, they are separated by a dash in the filename: locationID1-locationID2...

get_logfile_name	<i>Name logfile</i>
------------------	---------------------

Description

Create a name for the logfile from an input file/folder.

Usage

```
get_logfile_name()
```

Value

A string with format "log__YYYY-MM-DD_HH:MM:SS.log" where YYYY-MM-DD_HH:MM:SS is the current date/time.

guess_classifier	<i>Guess classifier</i>
------------------	-------------------------

Description

Guesses the classifier used to annotate the data based on the column names given in colnames_df.

Usage

```
guess_classifier(colnames_df)
```

Arguments

colnames_df A character vector of column names.

Value

The classifier: either zooniverse, digikam or traptagger.

PEcAn.standardizeSnapshot

standardizeSnapshot is a R package to standardize camera trap records files from the Snapshot Safari project.

Description

Snapshot Safari data comes in different standards, following the method that was used to classify pictures (Zooniverse, TrapTagger or Digikam). This package allows to standardize all data sources to a unique file format and then cleans the files to homogenize records. See Snapshot Safari: a large-scale collaborative to monitor Africa's remarkable biodiversity (Pardo et al, 2021) (<https://www.sajs.co.za/article/view/8134>)

prepare_digikam	<i>Prepare Digikam data</i>
-----------------	-----------------------------

Description

Standardizes Digikam data to be ready to be standardized: removes 'X' index column, (if it exists), merges 'Directory' and 'Filename' to create a path, splits 'metadata_Behaviour' column into the relevant behaviors and recodes 'metadata_young_present' into 0 and 1.

Usage

```
prepare_digikam(df)
```

Arguments

df Dataframe to standardize (must have columns 'metadata_Behaviour' and 'metadata_young_present').

Value

Returns the dataframe without 'X' column, 'Directory' and 'Filename' merged into 'file_path_1' column, 'metadata_Behaviour' splitted into up to 6 columns 'Eating', 'Drinking', 'Resting', 'Standing', 'Moving', 'Interacting' (depending on the behaviours that were present in the data) and the 'metadata_young_present' column recoded so that 'Yes'/'No' becomes 0/1.

prepare_traptagger	<i>Prepare TrapTagger data</i>
--------------------	--------------------------------

Description

Prepares data processed with TrapTagger to be standardized by creating columns date and time from timestamp and formatting capture_label_xx into a unique column.

Usage

```
prepare_traptagger(df)
```

Arguments

df The dataframe to standardize (must have columns starting with capture_label and column timestamp)

Value

Returns the dataframe with 2 additional columns date and time, minus the timestamp column, and with merged capture_label

read_snapshot_file	<i>Read a file</i>
--------------------	--------------------

Description

Read a file

Usage

```
read_snapshot_file(filename, base_folder, verbose = FALSE)
```

Arguments

filename	The name of the file which should exist in base_folder. starting at the base of the R project.
base_folder	The folder to look in
verbose	logical. Display the file name?

Value

A dataframe with the contents of the csv file

```
recode_behavior_digikam
```

Recode a behavior column

Description

Recode a behavior column

Usage

```
recode_behavior_digikam(behaviors)
```

Arguments

behaviors	a character vector of behaviors, expected to be separated by a ' _&_ '
-----------	------------------------------------------------------------------------

Value

A dataframe with one column per behavior initially present, with 1 if the behavior was present and 0 otherwise.

```
rename_standard
```

Rename columns according to standard

Description

This function renames the existing columns in a dataframe to match the standard names.

Usage

```
rename_standard(
  df,
  classifier = c("zooniverse", "traptagger", "digikam"),
  standard_colnames
)
```

Arguments

<code>df</code>	The dataframe with the columns to rename
<code>classifier</code>	The classifier used to create the dataframe <code>df</code> . Can be 'zooniverse', 'traptagger', 'digikam'.
<code>standard_colnames</code>	A dataframe with 2 columns (at least) named like the classifier and 'new'. The column named like the classifier contains column names that are expected in the initial file. These names will be matched in the column names of <code>df</code> using partial matching (case insensitive and removing blanks). The column 'new' contains the column names that are expected in the final file. Columns in the classifier column will be renamed following the name of the corresponding value in 'new'. If no old column corresponds to 'new' (indicated with a NA) then the column will be created and filled with NAs.

Value

Returns a dataframe for which the columns have been renamed

<code>standardize_columns</code>	<i>Standardize data</i>
----------------------------------	-------------------------

Description

Given a dataframe, standardizes column names to match a standard. It removes all original columns not in the standard and adds missing columns (which are NA)

Usage

```
standardize_columns(
  df,
  classifier = c("zooniverse", "traptagger", "digikam"),
  standard_colnames,
  verbose = FALSE
)
```

Arguments

<code>df</code>	The dataframe to standardize
<code>classifier</code>	The classifier used to generate the data: possible values are 'zooniverse', 'traptagger', 'digikam'
<code>standard_colnames</code>	A dataframe with 2 columns (at least) named like the classifier and 'new'. The column named like the classifier contains column names that are expected in the initial file. These names will be matched in the column names of <code>df</code> using partial matching (case insensitive and removing blanks). The column 'new' contains the column names that are expected in the final file. Columns in the classifier

column will be renamed following the name of the corresponding value in 'new'. If no old column corresponds to 'new' (indicated with a NA) then the column will be created and filled with NAs.

verbose Should the info output messages be displayed?

Value

A dataframe with 25 columns with the standardized names

standardize_date	<i>Standardize dates</i>
------------------	--------------------------

Description

Standardizes a vector of dates to the format "AAAA-MM-DD".

Usage

```
standardize_date(dates)
```

Arguments

dates A dates vector (character). Accepts all formats coercible to date with lubridate::as_date, else will look for "%d-%m-%Y" or "%m-%d-%Y".

Value

A vector of dates in the format "AAAA-MM-DD".

standardize_species	<i>Standardize species</i>
---------------------	----------------------------

Description

Eliminate species duplicate names (things like 'birdofprey' and 'birdsofprey')

Usage

```
standardize_species(species)
```

Arguments

species vector of species names

Value

the vector of species names with names standardized

standardize_time	<i>Standardize times</i>
------------------	--------------------------

Description

Standardizes a times vector to match the format "HH:MM:SS"

Usage

```
standardize_time(times)
```

Arguments

times	A times vector (character). Will work if the times are already "HH:MM:SS" or "HH:MM:SS AM/PM"
-------	-----------------------------------------------------------------------------------------------

Value

The vector standardized in the format "HH:MM:SS"

write_log_message	<i>Write log message</i>
-------------------	--------------------------

Description

Writes a log message. If a logger is provided, writes to that logger; if it is NA, displays a message.

Usage

```
write_log_message(message, logger = NA, level = "info")
```

Arguments

message	The message to display/write to the logger
logger	Logger to write to (log4r object of class "logger") (defaults to NA)
level	Logging level: either 'info', 'warn', 'debug' or 'error'.

Value

Either a message or writes a log (with the logger parameters)

`write_standardized_file`*Write the standardized file*

Description

Write the standardized file

Usage

```
write_standardized_file(df, in_filename, to)
```

Arguments

<code>df</code>	The standardized file
<code>in_filename</code>	The path to the original file. If it has subfolders, the subfolder structure is copied in to.
<code>to</code>	The target folder to copy data in. It must exist.

Value

Writes the file to the folder `to/xxx` where `xxx` is the subdirectory in which the original file was in. Also returns the path to the file.

Index

`create_logger`, [1](#)

`fill_capture_info`, [1](#)

`get_cameraID`, [2](#)

`get_csv_files_and_folders`, [3](#)

`get_eventID`, [3](#)

`get_final_filename`, [4](#)

`get_logfile_name`, [4](#)

`guess_classifier`, [5](#)

`PEcAn.standardizeSnapshot`, [5](#)

`prepare_digikam`, [5](#)

`prepare_traptagger`, [6](#)

`read_snapshot_file`, [6](#)

`recode_behavior_digikam`, [7](#)

`rename_standard`, [7](#)

`standardize_columns`, [8](#)

`standardize_date`, [9](#)

`standardize_species`, [9](#)

`standardize_time`, [10](#)

`write_log_message`, [10](#)

`write_standardized_file`, [11](#)