

Package ‘africabirdmap’

April 1, 2019

Type Package

Title Functions to clean, analyze and plot Africa Bird Map data.

Version 0.1.0

Author Les Underhill, Michael Brooks, Marc Burman, Colin Jackson

Maintainer David Clarence <dclarance@gmail.com>

Description Enables citizen science by providing easy to use tools so researches can focus on questions rather than struggling with the data.

License To be populated

Encoding UTF-8

LazyData true

Depends dplyr, scales, RCurl, digest, RMySQL, RJSONIO, digest

RoxygenNote 6.1.1

R topics documented:

days_since_month	1
extract_data	2
hello	3
reporting_rate	3
underhill_curves	4
underhill_smoother	5

Index	7
--------------	----------

days_since_month	<i>Calculate days from a given calendar month</i>
------------------	---

Description

An a-historical method to calculate pentades by calculating distance from a given calendar month.

Usage

```
days_since_month(date, start_month = 7)
```

Arguments

date	A 'date' object. This can be a single date or a vector of dates.
start_month	An integer between 1 and 12 (inclusive) that indicates the month you want to measure distance in days from. Default is 7.

Details

This functions assumes that all years are 365 days. Necessary corrections for leap years are made.

Value

This function returns an integer between 1 and 365 indicating the number of days elapsed since the first of the month indicated in 'start_month'

Examples

```
# single date conversion example
days_since_month(date = as.Date('2018-09-14'), start_month = 7)

# multiple dates conversion examples
sequence_of_dates <- c(as.Date("2019-01-01"), as.Date("2019-03-19"))
days_since_month(date = sequence_of_dates, start_month = 6)
```

extract_data	<i>Export data from the Africa Bird Map</i>
--------------	---

Description

Pull data from the African Bird Map fit for reporting rate and other analyses. This function is temporary and will be made redundant. It is unsafe and is only for demonstrative purposes.

Usage

```
extract_data(username, user_id, password, species_id)
```

Arguments

username	The citizen scientist's bird map email
user_id	The citizen scientist's bird map user number (usually 5 digits)
password	The citizen scientist's password.
species_id	The species_id for which data is extracted. A complete list of species name and ids are available on the Kenya Bird Map website.

Details

The function requires a username and password in plain text. This is not how the final function will look. The final function will call the RStudio API for a safe transaction. This is just for demonstrative purposes.

Value

A dataframe with every list submitted to Africa Bird Map (includes all countries).

Examples

```
## Not run:

# Extract data for the African Black-shouldered kite.
extract_data(username = "dclarance@gmail.com" ,user_id = 40664, password = "abcd", species_id = 103s)

## End(Not run)
```

hello	<i>Hello, World!</i>
-------	----------------------

Description

Prints 'Hello, world!'.

Usage

```
hello()
```

Examples

```
hello()
```

reporting_rate	<i>Calculate the reporting rate of a species</i>
----------------	--

Description

Obtain a reporting rate of a species over a specified time period and spatial configuration.

Usage

```
reporting_rate(df, species_id, start_date = "1970-01-01",
  end_date = Sys.Date(), pentad_id)
```

Arguments

df	A dataframe obtained from the species API call. Use 'extract_data' if required.
species_id	An integer id. The KBM website has a list of all species with their ids.
start_date	A character or date object denoting the starting point from which reporting rate is to be measured.
end_date	A character or date object denoting the end point from which reporting rate is to be measured.
pentad_id	A character vector denoting the pentad id.

Details

This functions assumes that all years are 365 days. Necessary corrections for leap years are made.

Value

A numeric object between 0 and 1 (inclusive) denoting the reporting rate of a species. The reporting rate is equal to the number of times a species has been recorded divided by the number of full protocol cards in the specified time and region.

Examples

```
## Not run:

# the reporting rate of African Black-shouldered Kite in Nairobi National Park
reporting_rate(df, species_id = 130, start_date = '1970-01-01', end_date = Sys.Date(), pentad_id = '0120-3650')

## End(Not run)
```

underhill_curves	<i>Plot the underhill smoother and associated confidence intervals for a species.</i>
------------------	---

Description

Fit a line through the reporting rates and its associated 1 and 2 sigma confidence intervals. 1 and 2 sigma roughly correspond to 67 percent and 95 percent confidence intervals. The line is over Pentades defined in the 'underhill_smoother()' function.

Usage

```
underhill_curves(underhill_smoother, species_id, species_name)
```

Arguments

underhill_smoother	The dataframe returned by the 'underhill_smoother function'.
species_id	The species_id for which data is extracted. A complete list of species name and ids are available on the Kenya Bird Map website.
species_name	The name of the species you are plotting. This is to use in the title of the plots.

Details

The dark, thinner interval is the 1 sigma interval and corresponds to 67

Value

A ggplot plot

Examples

```
## Not run:
```

```
underhill_curves(underhill_smoother, species_id = 103, species_name = "African Black-shouldered Kite")
```

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

underhill_smoother	<i>Fit the underhill smoother to a reporting rate distribution</i>
--------------------	--

Description

Fit the underhill smoother (put reference here) to reporting rate distribution over pentades. The underhill smoother is a locally weighted binomial general linear model using a logit link, where the weights are generated using an exponential distribution.

Usage

```
underhill_smoother(raw_data, species_id, start_month = 7, pentade_window,
  first_pentade = 1, last_pentade = 73)
```

Arguments

raw_data	Data for a species extracted using 'extract_data()'. The StartDate column should be of type 'date'.
species_id	The species_id for which data is extracted. A complete list of species name and ids are available on the Kenya Bird Map website.
start_month	The month to start the pentades from. Default is 7.
pentade_window	The number of pentades on either side of the target day to give the weights to.
first_pentade	The starting value of the pentades. Default is 1.
last_pentade	The last value of the pentades. Default is 73.

Details

Pentades are blocks of 5 consecutive days. The implementation measures pentades from a defined month 'start_month'. This implementation needs review and should not be considered final.

Value

A dataframe where each row is a pentade with the 'fit' values, the standard errors ('se'), the 'reporting_rates' and its associated inputs. Ignore the various date columns as those are only meant to be used in graphs.

Examples

```
## Not run:
```

```
underhill_smoother(raw_data, species_id, start_month = 7, pentade_window, first_pentade = 1, last_pentade = 73)
```

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

Index

days_since_month, [1](#)

extract_data, [2](#)

hello, [3](#)

reporting_rate, [3](#)

underhill_curves, [4](#)

underhill_smoother, [5](#)