

Package ‘specieshindex’

December 10, 2021

Type Package

Title How (scientifically) popular is a given species?

Version 0.4.1

Date 2021-01-19

Author Jessica Tam

Maintainer Jessica Tam <tamtinying@gmail.com>

Description Finds the h-index of a species.

Depends R (\geq 3.5.0)

LazyData true

Imports rscopus,

wosr,
rbace,
taxize,
dplyr,
tidyr,
data.table,
httr,
XML,
ggplot2,
ggpubr

Suggests devtools,

httpptest,
knitr,
RefManageR,
rmarkdown,
roxygen2,
testthat (\geq 3.0.0)

License MIT + file LICENSE

URL <https://github.com/jessicatytam/specieshindex>

RoxygenNote 7.1.2

VignetteBuilder knitr

Encoding UTF-8
Config/testthat/edition 3

R topics documented:

Allindices	2
Count	3
Fetch	4
getYear	5
languages	6
plotAllindices	6
plotPub	7
SourceType	8
SpH5	8
SpHAfterdate	9
SpHindex	9
Spi10	10
SpMindex	11
TotalCite	11
TotalJournals	12
TotalPub	12
YearsPublishing	13
Index	14

Allindices	<i>Index summary</i>
------------	----------------------

Description

This function returns a dataframe of the summary of all of the indices.

Usage

```
Allindices(data, genus, species, sourcetype = 0)
```

Arguments

- data The dataframe generated from [Fetch](#).
- genus Genus classification from the binomial name.
- species Species classification from the binomial name.
- sourcetype Source type; default is 0, enter 1 to add SourceType variables.

Value

A datarame of all of the indices in the package.

Examples

```
data(Woylie)
Allindices(Woylie,
           genus = "genus_name", species = "species_name",
           sourcetype = 0)
```

Count	<i>Search count of literature</i>
-------	-----------------------------------

Description

This function counts the total number of search results from Scopus, Web of Science, or BASE. A check will be conducted via [gnr_resolve](#) to validate the genus and species names.

Usage

```
Count(db, search, level, genus, species, synonyms, additionalkeywords)
```

Arguments

db	Literature database. Scopus ("scopus"), Web of Science ("wos"), or Base ("base").
search	Search fields. Title only ("t") or title, abstract, and keywords ("tak").
level	Taxonomic level. Genus ("genus") or species ("species").
genus	Genus classification from the binomial name.
species	Species classification from the binomial name.
synonyms	Alternate species names.
additionalkeywords	Optional search terms.

Value

Search count of the genus or species with the given `genus` and/or `species`.

Examples

```
## Not run:
Count(db = "scopus",
      search = "t",
      level = "species",
      genus = "Osphranter", species = "rufus")

## End(Not run)
## Not run:
Count(db = "scopus",
      search = "t",
```

```

    level = "species",
    genus = "Osphranter", species = "rufus",
    synonyms = "Macropus rufus",
    additionalkeywords = "conserv*")

## End(Not run)

```

Fetch

Fetch citation records

Description

This function fetches citation information from Scopus, Web of Science, or BASE. Duplicates are to be removed by the user after fetching the data.

Usage

```

Fetch(
  db,
  search,
  level,
  genus,
  species,
  synonyms,
  additionalkeywords,
  language = 0
)

```

Arguments

<code>db</code>	Literature database. Scopus ("scopus"), Web of Science ("wos"), or Base ("base").
<code>search</code>	Search fields. Title only ("t") or title, abstract, and keywords ("tak").
<code>level</code>	Taxonomic level. Genus ("genus") or species ("species").
<code>genus</code>	Genus classification from the binomial name.
<code>species</code>	Species classification from the binomial name.
<code>synonyms</code>	Alternate species names.
<code>additionalkeywords</code>	Optional search terms.
<code>language</code>	Language of the paper; default is 0, enter 1 to retrieve the variable. Scopus only.

Value

A dataframe of the genus' or species' citation records with the given `genus` and/or `species`.

Examples

```
## Not run:
Fetch(db = "scopus",
      search = "t",
      level = "species",
      genus = "Osphranter", species = "rufus")

## End(Not run)
## Not run:
Fetch(db = "scopus",
      search = "t",
      level = "species",
      genus = "Osphranter", species = "rufus",
      synonyms = "Macropus rufus",
      additionalkeywords = "conserv*")

## End(Not run)
```

getYear

Extract year

Description

Extracts the year of each publication of the output from any of the Fetch functions and counts the number of publications each year.

Usage

```
getYear(data, genus, species)
```

Arguments

data	Output from any of the fetch function.
genus	Genus classification from the binomial name.
species	Species classification from the binomial name.

Value

A dataframe with the year and frequency of the publications

Examples

```
getYear(data = Woylie,
        genus = "Bettongia", species = "penicillata")
```

languages

*Languages***Description**

List of languages of documents found on Scopus. Data was retrieved on 9 March 2021.

Usage

```
languages
```

Format

A csv file with the complete list of languages of documents found on Scopus.

Source

<https://www.elsevier.com/solutions/scopus/how-scopus-works/content>

plotAllindices

*Index plot***Description**

Plots the indices of a single species or combined.

Usage

```
plotAllindices(data)
```

Arguments

data	The dataframe generated from Allindices .
------	---

Value

```
ggplot
```

Examples

```
W <- Allindices(Woylie,
                genus = "Bettongia", species = "penicillata")
Q <- Allindices(Quokka,
                genus = "Setonix", species = "brachyurus")
P <- Allindices(Platypus,
                genus = "Ornithorhynchus", species = "anatinus")
K <- Allindices(Koala,
                genus = "Phascolarctos", species = "cinereus")
CombineSp <- dplyr::bind_rows(W, Q, P, K)
plotAllindices(CombineSp)
```

plotPub

Publication plot

Description

Plots the publication by year of a single species or combined.

Usage

```
plotPub(data)
```

Arguments

data The dataframe generated from [getYear](#).

Value

ggplot

Examples

```
extract_year_W <- getYear(data = Woylie,
                          genus = "Bettongia", species = "penicillata")
extract_year_Q <- getYear(data = Quokka,
                          genus = "Setonix", species = "brachyurus")
extract_year_P <- getYear(data = Platypus,
                          genus = "Ornithorhynchus", species = "anatinus")
extract_year_K <- getYear(data = Koala,
                          genus = "Phascolarctos", species = "cinereus")
Combine_pub <- rbind(extract_year_W, extract_year_Q, extract_year_P, extract_year_K)
plotPub(Combine_pub)
```

SourceType	<i>Source type</i>
------------	--------------------

Description

This function calculates the total number of items for each document type.

Usage

```
SourceType(data)
```

Arguments

data	The dataframe generated from Fetch .
------	--

Value

A dataframe with each document and their counts.

Examples

```
data(Woylie)
SourceType(Woylie)
```

SpH5	<i>Species h5 index</i>
------	-------------------------

Description

This function calculates the h-index of a species in the past 5 years.

Usage

```
SpH5(data)
```

Arguments

data	The dataframe generated from Fetch .
------	--

Value

H5 index.

References

Suzuki, H. (2012). *Google Scholar Metrics for Publications*. Retrieved from <https://scholar.googleblog.com/2012/04/google-scholar-metrics-for-publications.html>.

Examples

```
data(Woylie)
SpH5(Woylie)
```

SpHAfterdate	<i>Species h-index with a given time frame</i>
--------------	--

Description

This function calculates the h-index using a given date up till the newest record.

Usage

```
SpHAfterdate(data, date)
```

Arguments

data	The dataframe generated from Fetch .
date	The lower limit of the timeframe.

Value

H-index of the given time period.

Examples

```
data(Woylie)
SpHAfterdate(Woylie, "2000-01-01")
```

SpHIndex	<i>Species h-index</i>
----------	------------------------

Description

This function calculates the h-index of a species.

Usage

```
SpHIndex(data)
```

Arguments

data	The dataframe generated from Fetch .
------	--

Value

H-index.

References

Bertoli-Barsotti, L. & Lando, T. (2015). On a formula for the h-index. *Journal of Informetrics*, 9(4), 762-776.

Hirsch, J. (2005). An index to quantify an individual's scientific research output. *Proceedings of the National Academy of Sciences of the United States of America*, 102(46), 16569-16572.

Examples

```
data(Woylie)
SpHindex(Woylie)
```

Spi10

Species i10 index

Description

This function calculates the i10 index of a species. i10 index counts all of the publications with 10 or more citations.

Usage

```
Spi10(data)
```

Arguments

`data` The dataframe generated from [Fetch](#).

Value

i10 index.

References

Cornell University (2019). *i10-index*. Retrieved from <https://guides.library.cornell.edu/c.php?g=32272&p=203393>.

Examples

```
data(Woylie)
Spi10(Woylie)
```

SpMindex	<i>Species m-index</i>
----------	------------------------

Description

This function calculates the m-index of species. M-index uses the h-index and divides it by the number of years of activity.

Usage

```
SpMindex(data)
```

Arguments

`data` The dataframe generated from [Fetch](#).

Value

M-index.

References

University of Pittsburgh (2019). *Research Impact and Metrics: Author metrics*. Retrieved from <https://pitt.libguides.com/bibliometricIndicators/AuthorMetrics>.

Examples

```
data(Woylie)
SpMindex(Woylie)
```

TotalCite	<i>Total citations</i>
-----------	------------------------

Description

This function calculates the total number of citations.

Usage

```
TotalCite(data)
```

Arguments

`data` The dataframe generated from [Fetch](#).

Value

A numerical value of the total number of citations.

Examples

```
data(Woylie)
TotalCite(Woylie)
```

TotalJournals	<i>Total journals</i>
---------------	-----------------------

Description

This function calculates the total number of journals.

Usage

```
TotalJournals(data)
```

Arguments

`data` The dataframe generated from [Fetch](#).

Value

An integer of the total number of journals.

Examples

```
data(Woylie)
TotalJournals(Woylie)
```

TotalPub	<i>Total publications</i>
----------	---------------------------

Description

This function calculates the total number of publications.

Usage

```
TotalPub(data)
```

Arguments

`data` The dataframe generated from [Fetch](#).

Value

An integer of the total number of publications.

Examples

```
data(Woylie)
TotalPub(Woylie)
```

YearsPublishing	<i>Years since first publication</i>
-----------------	--------------------------------------

Description

The number of years since the first publication in relation to the species.

Usage

```
YearsPublishing(data)
```

Arguments

`data` The dataframe generated from [Fetch](#).

Value

Number of years.

Examples

```
data(Woylie)
YearsPublishing(Woylie)
```

Index

- * **datasets**
 - languages, [6](#)
- Allindices, [2](#), [6](#)
- Count, [3](#)
- Fetch, [2](#), [4](#), [8-13](#)
- getYear, [5](#), [7](#)
- gnr_resolve, [3](#)
- languages, [6](#)
- plotAllindices, [6](#)
- plotPub, [7](#)
- SourceType, [8](#)
- SpH5, [8](#)
- SpHAfterdate, [9](#)
- SpHindex, [9](#)
- SpI10, [10](#)
- SpMindex, [11](#)
- TotalCite, [11](#)
- TotalJournals, [12](#)
- TotalPub, [12](#)
- YearsPublishing, [13](#)