Package 'specieshindex'

September 17, 2021

Type Package

Title How (scientifically) popular is a given species?

Version 0.2.1

Date 2021-01-19

Author Jessica Tam

Maintainer Jessica Tam <tamtinying@gmail.com>

Description Finds the h-index of a species.

Depends R (>= 3.5.0)

LazyData true

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

Suggests testthat,
   knitr,
   roxygen2,
   devtools,
   rmarkdown,
   RefManageR

License MIT + file LICENSE

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

RoxygenNote 7.1.2

VignetteBuilder knitr

Encoding UTF-8

R **topics documented:**

Allindices   *Index summary*

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 FetchSpT or FetchSpTAK.
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)
```

CountGenusT   *CountSpT wrapper*

Description

This is a wrapper function for CountGenusT_scopus, CountGenusT_wos, and CountGenusT_base.

Usage

```
CountGenusT(db, genus, synonyms, additionalkeywords)
```

Arguments

db	Literature database. Scopus ("scopus"), Web of Science ("wos"), or Base ("base").
genus	Genus classification from the binomial name.
synonyms	Alternate genus names.
additionalkeywords	
	Optional search terms.

Value

Search count of the genus with the given genus.

Examples

```
## Not run:
CountGenusT("scopus", genus = "Osphranter")

## End(Not run)
## Not run:
CountGenusT("scopus", genus = "Osphranter", synonyms = "Macropus", additionalkeywords = "conserv*")

## End(Not run)
```

CountGenusTAK   *CountSpTAK wrapper*

Description

This is a wrapper function for CountGenusTAK_scopus, CountGenusTAK_wos, and CountGenusTAK_base.

Usage

```
CountGenusTAK(db, genus, synonyms, additionalkeywords)
```

Arguments

db	Literature database. Scopus ("scopus"), Web of Science ("wos"), or Base ("base").
genus	Genus classification from the binomial name.
synonyms	Alternate genus names.
additionalkeywords	
	Optional search terms.

Value

Search count of the genus with the given genus.

Examples

```
## Not run:
CountGenusTAK("scopus", genus = "Osphranter")

## End(Not run)
## Not run:
CountGenusTAK("scopus", genus = "Osphranter", synonyms = "Macropus", additionalkeywords = "conserv*")

## End(Not run)
```

CountGenusTAK_base *Search count from BASE - title, abstract and author keywords*

Description

This function counts the total number of search results. It counts the publications with the genus name in the title, abstract and author keywords. A check will be conducted via gnr_resolve to validate the genus name.

Usage

```
CountGenusTAK_base(genus, synonyms, additionalkeywords)
```

Arguments

genus   Genus classification from the binomial name.

synonyms   Alternate genus names.

additionalkeywords
                  Optional search terms.

Value

Search count of the genus with the given genus.

References

Chamberlain, S. & Szocs, E. (2013). taxize - taxonomic search and retrieval in R. *F1000Research, 2*, 191.

Examples

```
## Not run:
CountGenusTAK_base("Bettongia")

#lower case letter in genus is also accepted and will return identical results

CountGenusTAK_base("bettongia")

## End(Not run)
## Not run:
CountGenusTAK_base("Bettongia", "conserv*")

#lower case letter in genus is also accepted and will return identical results

CountGenusTAK_base("bettongia", "conserv*")

## End(Not run)
```

CountGenusTAK_scopus   *Search count from Scopus - title, abstract, and keywords*

Description

This function counts the total number of search results. It counts the publications with the genus name in the title, abstract, and keywords. A check will be conducted via `gnr_resolve` to validate the genus name.

Usage

```
CountGenusTAK_scopus(
  genus,
  synonyms,
  additionalkeywords,
  datatype = "application/xml"
)
```

Arguments

genus   Genus classification from the binomial name.

synonyms   Alternate genus names.

additionalkeywords

                    Optional search terms.

datatype   Formats the URL to be sent to the API. The default is "application/xml".

Value

Search count of the genus with the given genus.

References

Chamberlain, S. & Szocs, E. (2013). taxize - taxonomic search and retrieval in R. *F1000Research, 2*, 191.

Examples

```
## Not run:
CountGenusTAK_scopus("Bettongia")

#lower case letter in genus is also accepted and will return identical results

CountGenusTAK_scopus("bettongia")

## End(Not run)
## Not run:
CountGenusTAK_scopus("Bettongia")

#lower case letter in genus is also accepted and will return identical results

CountGenusTAK_scopus("bettongia")

## End(Not run)
```

CountGenusTAK_wos   *Search count from Web of Science - title, abstract and author keywords*

Description

This function counts the total number of search results. It counts the publications with the genus name in the title, abstract and author keywords. A check will be conducted via `gnr_resolve` to validate the genus name.

Usage

```
CountGenusTAK_wos(genus, synonyms, additionalkeywords)
```

Arguments

genus   Genus classification from the binomial name.

synonyms   Alternate genus names.

additionalkeywords
                     Optional search terms.

Value

Search count of the genus with the given genus.

References

Chamberlain, S. & Szocs, E. (2013). taxize - taxonomic search and retrieval in R. *F1000Research, 2*, 191.

Examples

```
## Not run:
CountGenusTAK_wos("Bettongia")

#lower case letter in genus is also accepted and will return identical results

CountGenusTAK_wos("bettongia")

## End(Not run)
## Not run:
CountGenusTAK_wos("Bettongia", "conserv*")

#lower case letter in genus is also accepted and will return identical results

CountGenusTAK_wos("bettongia", "conserv*")

## End(Not run)
```

CountGenusT_base *Search count from BASE - title only*

Description

This function counts the total number of search results. It counts the publications with the genus name in the title only. A check will be conducted via `gnr_resolve` to validate the genus name.

Usage

```
CountGenusT_base(genus, synonyms, additionalkeywords)
```

Arguments

genus   Genus classification from the binomial name.

synonyms   Alternate genus names.

additionalkeywords
                    Optional search terms.

Value

Search count of the genus with the given genus.

References

Chamberlain, S. & Szocs, E. (2013). taxize - taxonomic search and retrieval in R. *F1000Research, 2*, 191.

Examples

```
## Not run:
CountGenusT_base("Bettongia")

#lower case letter in genus is also accepted and will return identical results

CountGenusT_base("bettongia")

## End(Not run)
## Not run:
CountGenusT_base("Bettongia", "conserv*")

#lower case letter in genus is also accepted and will return identical results

CountGenusT_base("bettongia", "conserv*")

## End(Not run)
```

CountGenusT_scopus   *Search count from Scopus - title only*

Description

This function counts the total number of search results. It counts the publications with the genus name in the title only. A check will be conducted via gnr_resolve to validate the genus name.

Usage

```
CountGenusT_scopus(
  genus,
  synonyms,
  additionalkeywords,
  datatype = "application/xml"
)
```

Arguments

genus	Genus classification from the binomial name.
synonyms	Alternate genus names.
additionalkeywords	
	Optional search terms.
datatype	Formats the URL to be sent to the API. The default is "application/xml".

Value

Search count of the genus with the given genus.

References

Chamberlain, S. & Szocs, E. (2013). taxize - taxonomic search and retrieval in R. *F1000Research, 2*, 191.

Examples

```
## Not run:
CountGenusT_scopus("Bettongia")

#lower case letter in genus is also accepted and will return identical results

CountGenusT_scopus("bettongia")

## End(Not run)
## Not run:
CountGenusT_scopus("Bettongia", "conserv*")

#lower case letter in genus is also accepted and will return identical results

CountGenusT_scopus("bettongia", "conserv*")

## End(Not run)
```

CountGenusT_wos	*Search count from Web of Science - title only*

Description

This function counts the total number of search results. It counts the publications with the genus name in the title only. A check will be conducted via `gnr_resolve` to validate the genus name.

Usage

```
CountGenusT_wos(genus, synonyms, additionalkeywords)
```

Arguments

genus   Genus classification from the binomial name.

synonyms   Alternate genus names.

additionalkeywords
                   Optional search terms.

Value

Search count of the genus with the given genus.

References

Chamberlain, S. & Szocs, E. (2013). taxize - taxonomic search and retrieval in R. *F1000Research, 2*, 191.

Examples

```
## Not run:
CountGenusT_wos("Bettongia")

#lower case letter in genus is also accepted and will return identical results

CountGenusT_wos("bettongia")

## End(Not run)
## Not run:
CountGenusT_wos("Bettongia", "conserv*")

#lower case letter in genus is also accepted and will return identical results

CountGenusT_wos("bettongia", "conserv*")

## End(Not run)
```

CountSpT   *CountSpT wrapper*

Description

This is a wrapper function for `CountSpT_scopus`, `CountSpT_wos`, and `CountSpT_base`.

Usage

```
CountSpT(db, genus, species, synonyms, additionalkeywords)
```

Arguments

db	Literature database. Scopus ("scopus"), Web of Science ("wos"), or Base ("base").
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 species with the given `genus` and `species`.

Examples

```
## Not run:
CountSpT("scopus", genus = "Osphranter", species = "rufus")

## End(Not run)
## Not run:
CountSpT("scopus", genus = "Osphranter", species = "rufus", synonyms = "Macropus rufus", additionalkeywords = "
## End(Not run)
```

CountSpTAK   *CountSpTAK wrapper*

Description

This is a wrapper function for `CountSpTAK_scopus`, `CountSpTAK_wos`, and `CountSpTAK_base`.

Usage

```
CountSpTAK(db, genus, species, synonyms, additionalkeywords)
```

Arguments

db	Literature database. Scopus ("scopus"), Web of Science ("wos"), or Base ("base").
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 species with the given genus and species.

Examples

```
## Not run:
CountSpTAK("scopus", genus = "Osphranter", species = "rufus")

## End(Not run)
## Not run:
CountSpTAK("scopus", genus = "Osphranter", species = "rufus", synonyms = "Macropus rufus", additionalkeywords =

## End(Not run)
```

CountSpTAK_base   *Search count from BASE - title, abstract and author keywords*

Description

This function counts the total number of search results. It counts the publications with the binomial name in the title, abstract and author keywords. A check will be conducted via gnr_resolve to validate the genus and species names.

Usage

```
CountSpTAK_base(genus, species, synonyms, additionalkeywords)
```

Arguments

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 species with the given genus and species.

References

Chamberlain, S. & Szocs, E. (2013). taxize - taxonomic search and retrieval in R. *F1000Research, 2*, 191.

Examples

```
## Not run:
CountSpTAK_base("Bettongia", "penicillata")

#lower case letter in genus is also accepted and will return identical results

CountSpTAK_base("bettongia", "penicillata")

## End(Not run)
## Not run:
CountSpTAK_base("Bettongia", "penicillata", "conserv*")

#lower case letter in genus is also accepted and will return identical results

CountSpTAK_base("bettongia", "penicillata", "conserv*")

## End(Not run)
```

CountSpTAK_scopus   *Search count from Scopus - title, abstract, and keywords*

Description

This function counts the total number of search results. It counts the publications with the binomial name in the title, abstract, and keywords. A check will be conducted via `gnr_resolve` to validate the genus and species names.

Usage

```
CountSpTAK_scopus(
  genus,
  species,
  synonyms,
  additionalkeywords,
  datatype = "application/xml"
)
```

Arguments

genus   Genus classification from the binomial name.

species   Species classification from the binomial name.

synonyms   Alternate species names.

additionalkeywords
                    Optional search terms.

datatype   Formats the URL to be sent to the API. The default is "application/xml".

Value

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

References

Chamberlain, S. & Szocs, E. (2013). taxize - taxonomic search and retrieval in R. *F1000Research, 2*, 191.

Examples

```
## Not run:
CountSpTAK_scopus("Bettongia", "penicillata")

#lower case letter in genus is also accepted and will return identical results

CountSpTAK_scopus("bettongia", "penicillata")

## End(Not run)
## Not run:
CountSpTAK_scopus("Bettongia", "penicillata")

#lower case letter in genus is also accepted and will return identical results

CountSpTAK_scopus("bettongia", "penicillata")

## End(Not run)
```

CountSpTAK_wos   *Search count from Web of Science - title, abstract and author keywords*

Description

This function counts the total number of search results. It counts the publications with the binomial name in the title, abstract and author keywords. A check will be conducted via gnr_resolve to validate the genus and species names.

Usage

```
CountSpTAK_wos(genus, species, synonyms, additionalkeywords)
```

Arguments

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 species with the given genus and `species`.

References

Chamberlain, S. & Szocs, E. (2013). taxize - taxonomic search and retrieval in R. *F1000Research, 2*, 191.

Examples

```
## Not run:
CountSpTAK_wos("Bettongia", "penicillata")

#lower case letter in genus is also accepted and will return identical results

CountSpTAK_wos("bettongia", "penicillata")

## End(Not run)
## Not run:
CountSpTAK_wos("Bettongia", "penicillata", "conserv*")

#lower case letter in genus is also accepted and will return identical results

CountSpTAK_wos("bettongia", "penicillata", "conserv*")

## End(Not run)
```

CountSpT_base *Search count from BASE - title only*

Description

This function counts the total number of search results. It counts the publications with the binomial name in the title only. A check will be conducted via `gnr_resolve` to validate the genus and species names.

Usage

```
CountSpT_base(genus, species, synonyms, additionalkeywords)
```

Arguments

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 species with the given `genus` and `species`.

References

Chamberlain, S. & Szocs, E. (2013). taxize - taxonomic search and retrieval in R. *F1000Research, 2*, 191.

Examples

```
## Not run:
CountSpT_base("Bettongia", "penicillata")

#lower case letter in genus is also accepted and will return identical results

CountSpT_base("bettongia", "penicillata")

## End(Not run)
## Not run:
CountSpT_base("Bettongia", "penicillata", "conserv*")

#lower case letter in genus is also accepted and will return identical results

CountSpT_base("bettongia", "penicillata", "conserv*")

## End(Not run)
```

CountSpT_scopus   *Search count from Scopus - title only*

Description

This function counts the total number of search results. It counts the publications with the binomial
name in the title only. A check will be conducted via `gnr_resolve` to validate the genus and species
names.

Usage

```
CountSpT_scopus(
  genus,
  species,
  synonyms,
  additionalkeywords,
  datatype = "application/xml"
)
```

Arguments

genus   Genus classification from the binomial name.

species   Species classification from the binomial name.

synonyms   Alternate species names.

additionalkeywords
                  Optional search terms.

datatype   Formats the URL to be sent to the API. The default is "application/xml".

Value

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

References

Chamberlain, S. & Szocs, E. (2013). taxize - taxonomic search and retrieval in R. *F1000Research, 2*, 191.

Examples

```
## Not run:
CountSpT_scopus("Bettongia", "penicillata")

#lower case letter in genus is also accepted and will return identical results

CountSpT_scopus("bettongia", "penicillata")

## End(Not run)
## Not run:
CountSpT_scopus("Bettongia", "penicillata", "conserv*")

#lower case letter in genus is also accepted and will return identical results

CountSpT_scopus("bettongia", "penicillata", "conserv*")

## End(Not run)
```

CountSpT_wos *Search count from Web of Science - title only*

Description

This function counts the total number of search results. It counts the publications with the binomial name in the title only. A check will be conducted via `gnr_resolve` to validate the genus and species names.

Usage

```
CountSpT_wos(genus, species, synonyms, additionalkeywords)
```

Arguments

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 species with the given `genus` and `species`.

References

Chamberlain, S. & Szocs, E. (2013). taxize - taxonomic search and retrieval in R. *F1000Research, 2*, 191.

Examples

```
## Not run:
CountSpT_wos("Bettongia", "penicillata")

#lower case letter in genus is also accepted and will return identical results

CountSpT_wos("bettongia", "penicillata")

## End(Not run)
## Not run:
CountSpT_wos("Bettongia", "penicillata", "conserv*")

#lower case letter in genus is also accepted and will return identical results

CountSpT_wos("bettongia", "penicillata", "conserv*")

## End(Not run)
```

FetchGenusT　　　　　　　　　　　　　*FetchGenusT wrapper*

Description

This is a wrapper function for `FetchSpT_scopus`, and `FetchSpT_wos`.

Usage

```
FetchGenusT(db, genus, synonyms, additionalkeywords, language = 0)
```

Arguments

db	Literature database. Scopus ("scopus") or Web of Science ("wos").
genus	Genus classification from the binomial name.
synonyms	Alternate genus names.
additionalkeywords	
	Optional search terms.
language	Language of the paper; default is 0, enter 1 to retrieve the variable. Scopus only.

Value

A dataframe of the genus' citation records with the given genus.

Examples

```
## Not run:
FetchGenusT("scopus", genus = "Osphranter")

## End(Not run)
## Not run:
FetchGenusT("scopus", genus = "Osphranter", synonyms = "Macropus", additionalkeywords = "conserv*")

## End(Not run)
```

FetchGenusTAK   *FetchGenusTAK wrapper*

Description

This is a wrapper function for `FetchGenusTAK_scopus`, and `FetchGenusTAK_wos`.

Usage

```
FetchGenusTAK(db, genus, synonyms, additionalkeywords, language = 0)
```

Arguments

db	Literature database. Scopus ("scopus") or Web of Science ("wos").
genus	Genus classification from the binomial name.
synonyms	Alternate genus names.
additionalkeywords	
	Optional search terms.
language	Language of the paper; default is 0, enter 1 to retrieve the variable. Scopus only.

Value

A dataframe of the genus' citation records with the given genus.

Examples

```
## Not run:
FetchGenusTAK("scopus", genus = "Osphranter")

## End(Not run)
## Not run:
FetchGenusTAK("scopus", genus = "Osphranter", synonyms = "Macropus", additionalkeywords = "conserv*")

## End(Not run)
```

FetchGenusTAK_scopus *Fetch data from Scopus - title, abstract and keywords*

Description

This function fetches citation information from Scopus using genus name found in the title, abstract and keywords of the publications. Duplicates are to be removed by the user after fetching the data.

Usage

```
FetchGenusTAK_scopus(genus, synonyms, additionalkeywords, language = 0)
```

Arguments

genus   Genus classification from the binomial name.

synonyms   Alternate genus names.

additionalkeywords
                     Optional search terms.

language   Language of the paper; default is 0, enter 1 to retrieve the variable.

Value

A dataframe of the genus' citation records with the given genus.

Examples

```
## Not run:
FetchGenusTAK_scopus("Bettongia")

#lower case letter in genus is also accepted and will return identical results

FetchGenusTAK_scopus("bettongia")

## End(Not run)
## Not run:
FetchGenusTAK_scopus("Bettongia", "conserv*")

#lower case letter in genus is also accepted and will return identical results

FetchGenusTAK_scopus("bettongia", "conserv*")

## End(Not run)
```

FetchGenusTAK_wos   *Fetch data from Web of Science - title, abstract and author keywords.*

Description

This function fetches citation information from Web of Science using genus name found in the title, abstract and author keywords of the publications. Duplicates are to be removed by the user after fetching the data.

Usage

```
FetchGenusTAK_wos(genus, synonyms, additionalkeywords)
```

Arguments

genus   Genus classification from the binomial name.

synonyms   Alternate genus names.

additionalkeywords
                     Optional search terms.

Value

A dataframe of the genus' citation records with the given genus.

Examples

```
## Not run:
FetchGenusTAK_wos("Bettongia")

#lower case letter in genus is also accepted and will return identical results

FetchGenusTAK_wos("bettongia")

## End(Not run)
## Not run:
FetchGenusTAK_wos("Bettongia", "conserv*")

#lower case letter in genus is also accepted and will return identical results

FetchGenusTAK_wos("bettongia", "conserv*")

## End(Not run)
```

FetchGenusT_scopus *Fetch data from Scopus - title only*

Description

This function fetches citation information from Scopus using genus name found in the title of the publications. Duplicates are to be removed by the user after fetching the data.

Usage

```
FetchGenusT_scopus(genus, synonyms, additionalkeywords, language = 0)
```

Arguments

genus   Genus classification from the binomial name.

synonyms   Alternate genus names.

additionalkeywords
               Optional search terms.

language   Language of the paper; default is 0, enter 1 to retrieve the variable.

Value

A dataframe of the genus' citation records with the given genus.

Examples

```
## Not run:
FetchGenusT_scopus("Bettongia")

#lower case letter in genus is also accepted and will return identical results

FetchGenusT_scopus("bettongia")

## End(Not run)
## Not run:
FetchGenusT_scopus("Bettongia", "conserv*")

#lower case letter in genus is also accepted and will return identical results

FetchGenusT_scopus("bettongia", "conserv*")

## End(Not run)
```

FetchGenusT_wos   *Fetch data from Web of Science - title only*

Description

This function fetches citation information from Web of Science using genus name found in the title of the publications. Duplicates are to be removed by the user after fetching the data.

Usage

```
FetchGenusT_wos(genus, synonyms, additionalkeywords)
```

Arguments

genus   Genus classification from the binomial name.

synonyms   Alternate genus names.

additionalkeywords
                     Optional search terms.

Value

A dataframe of the genus' citation records with the given genus.

Examples

```
## Not run:
FetchGenusT_wos("Bettongia")

#lower case letter in genus is also accepted and will return identical results

FetchGenusT_wos("bettongia")

## End(Not run)
## Not run:
FetchGenusT_wos("Bettongia", "conserv*")
```

```
#lower case letter in genus is also accepted and will return identical results

FetchGenusT_wos("bettongia", "conserv*")

## End(Not run)
```

FetchSpT   *FetchSpT wrapper*

Description

This is a wrapper function for `FetchSpT_scopus`, and `FetchSpT_wos`.

Usage

```
FetchSpT(db, genus, species, synonyms, additionalkeywords, language = 0)
```

Arguments

db	Literature database. Scopus ("scopus") or Web of Science ("wos").
genus	Genus classification from the binomial name.
species	Species classification from the binomial name.
synonyms	Alternate species names.
additionalkeywords	
	Optional search terms.
language	Language of the paper; default is 0, enter 1 to retrieve the variable. Scopus only.

Value

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

Examples

```
## Not run:
FetchSpT("scopus", genus = "Osphranter", species = "rufus")

## End(Not run)
## Not run:
FetchSpT("scopus", genus = "Osphranter", species = "rufus", synonyms = "Macropus rufus", additionalkeywords = "

## End(Not run)
```

FetchSpTAK   *FetchSpTAK wrapper*

Description

This is a wrapper function for `FetchSpTAK_scopus`, and `FetchSpTAK_wos`.

Usage

```
FetchSpTAK(db, genus, species, synonyms, additionalkeywords, language = 0)
```

Arguments

db	Literature database. Scopus ("scopus") or Web of Science ("wos").
genus	Genus classification from the binomial name.
species	Species classification from the binomial name.
synonyms	Alternate species names.
additionalkeywords	
	Optional search terms.
language	Language of the paper; default is 0, enter 1 to retrieve the variable. Scopus only.

Value

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

Examples

```
## Not run:
FetchSpTAK("scopus", genus = "Osphranter", species = "rufus")

## End(Not run)
## Not run:
FetchSpTAK("scopus", genus = "Osphranter", species = "rufus", synonyms = "Macropus rufus", additionalkeywords =

## End(Not run)
```

FetchSpTAK_scopus   *Fetch data from Scopus - title, abstract and keywords*

Description

This function fetches citation information from Scopus using genus and species name found in the title, abstract and keywords of the publications. Duplicates are to be removed by the user after fetching the data.

Usage

```
FetchSpTAK_scopus(genus, species, synonyms, additionalkeywords, language = 0)
```

Arguments

genus	Genus classification from the binomial name.
species	Species classification from the binomial name.
synonyms	Alternate species names.
additionalkeywords	
	Optional search terms.
language	Language of the paper; default is 0, enter 1 to retrieve the variable.

Value

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

Examples

```
## Not run:
FetchSpTAK_scopus("Bettongia", "penicillata")

#lower case letter in genus is also accepted and will return identical results

FetchSpTAK_scopus("bettongia", "penicillata")

## End(Not run)
## Not run:
FetchSpTAK_scopus("Bettongia", "penicillata", "conserv*")

#lower case letter in genus is also accepted and will return identical results

FetchSpTAK_scopus("bettongia", "penicillata", "conserv*")

## End(Not run)
```

FetchSpTAK_wos   *Fetch data from Web of Science - title, abstract and author keywords.*

Description

This function fetches citation information from Web of Science using genus and species name found in the title, abstract and author keywords of the publications. Duplicates are to be removed by the user after fetching the data.

Usage

```
FetchSpTAK_wos(genus, species, synonyms, additionalkeywords)
```

Arguments

genus	Genus classification from the binomial name.
species	Species classification from the binomial name.
synonyms	Alternate species names.
additionalkeywords	
	Optional search terms.

Value

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

Examples

```
## Not run:
FetchSpTAK_wos("Bettongia", "penicillata")

#lower case letter in genus is also accepted and will return identical results

FetchSpTAK_wos("bettongia", "penicillata")

## End(Not run)
## Not run:
FetchSpTAK_wos("Bettongia", "penicillata", "conserv*")

#lower case letter in genus is also accepted and will return identical results

FetchSpTAK_wos("bettongia", "penicillata", "conserv*")

## End(Not run)
```

FetchSpT_scopus   *Fetch data from Scopus - title only*

Description

This function fetches citation information from Scopus using genus and species name found in the title of the publications. Duplicates are to be removed by the user after fetching the data.

Usage

```
FetchSpT_scopus(genus, species, synonyms, additionalkeywords, language = 0)
```

Arguments

genus	Genus classification from the binomial name.
species	Species classification from the binomial name.
synonyms	Alternate species names.
additionalkeywords	
	Optional search terms.
language	Language of the paper; default is 0, enter 1 to retrieve the variable.

Value

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

Examples

```
## Not run:
FetchSpT_scopus("Bettongia", "penicillata")

#lower case letter in genus is also accepted and will return identical results

FetchSpT_scopus("bettongia", "penicillata")

## End(Not run)
## Not run:
FetchSpT_scopus("Bettongia", "penicillata", "conserv*")

#lower case letter in genus is also accepted and will return identical results

FetchSpT_scopus("bettongia", "penicillata", "conserv*")

## End(Not run)
```

FetchSpT_wos   *Fetch data from Web of Science - title only*

Description

This function fetches citation information from Web of Science using genus and species name found in the title of the publications. Duplicates are to be removed by the user after fetching the data.

Usage

```
FetchSpT_wos(genus, species, synonyms, additionalkeywords)
```

Arguments

genus   Genus classification from the binomial name.

species   Species classification from the binomial name.

synonyms   Alternate species names.

additionalkeywords

        Optional search terms.

Value

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

Examples

```
## Not run:
FetchSpT_wos("Bettongia", "penicillata")

#lower case letter in genus is also accepted and will return identical results

FetchSpT_wos("bettongia", "penicillata")

## End(Not run)
```

```
## Not run:
FetchSpT_wos("Bettongia", "penicillata", "conserv*")

#lower case letter in genus is also accepted and will return identical results

FetchSpT_wos("bettongia", "penicillata", "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")
```

Koala   *Koala dataset*

Description

Citation records of koala (Phascolarctos cinereus) from Scopus. Data was retrieved on 10 July 2020.

Usage

```
Koala
```

Format

A data frame with 773 rows and 20 variables

Source

http://api.elsevier.com/content/search/scopus

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

Platypus	*Platypus dataset*

Description

Citation records of platypus (Ornithorhynchus anatinus) from Scopus. Data was retrieved on 10 July 2020.

Usage

```
Platypus
```

Format

A data frame with 321 rows and 20 variables

Source

http://api.elsevier.com/content/search/scopus

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)
```

Quokka   *Quokka dataset*

Description

Citation records of quokka (Setonix brachyurus) from Scopus. Data was retrieved on 10 July 2020.

Usage

```
Quokka
```

Format

A data frame with 242 rows and 20 variables

Source

<http://api.elsevier.com/content/search/scopus>

SourceType   *Source type*

Description

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

Usage

```
SourceType(data)
```

Arguments

data   The dataframe generated from `FetchSpT` or `FetchSpTAK`.

Value

A dataframe with each document and their counts.

Examples

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

SpH5 *Species h5 index*

Description

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

Usage

```
SpH5(data)
```

Arguments

data   The dataframe generated from `FetchSpT` or `FetchSpTAK`.

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 *Species h-index with a given time frame*

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 `FetchSpT` or `FetchSpTAK`.
date   The lower limit of the timeframe.

Value

H-index of the given time period.

Examples

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

SpHindex *Species h-index*

Description

This function calculates the h-index of a species.

Usage

```
SpHindex(data)
```

Arguments

data   The dataframe generated from FetchSpT or FetchSpTAK.

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 FetchSpT or FetchSpTAK.

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   *Species m-index*

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 FetchSpT or FetchSpTAK.

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   *Total citations*

Description

This function calculates the total number of citations.

Usage

```
TotalCite(data)
```

Arguments

data   The dataframe generated from FetchSpT or FetchSpTAK.

Value

A numerical value of the total number of citations.

Examples

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

TotalJournals   *Total journals*

Description

This function calculates the total number of journals.

Usage

```
TotalJournals(data)
```

Arguments

data   The dataframe generated from FetchSpT or FetchSpTAK.

Value

An integer of the total number of journals.

Examples

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

TotalPub   *Total publications*

Description

This function calculates the total number of publications.

Usage

```
TotalPub(data)
```

Arguments

data   The dataframe generated from `FetchSpT` or `FetchSpTAK`.

Value

An integer of the total number of publications.

Examples

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

Woylie   *Woylie dataset*

Description

Citation records of woylie (Bettongia penicillata) from Scopus. Data was retrieved on 10 July 2020.

Usage

```
Woylie
```

Format

A data frame with 113 rows and 20 variables

Source

http://api.elsevier.com/content/search/scopus

YearsPublishing | *Years since first publication*

Description

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

Usage

```
YearsPublishing(data)
```

Arguments

data   The dataframe generated from `FetchSpT` or `FetchSpTAK`.

Value

Number of years.

Examples

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

Index