Package 'MDPIexploreR'

May 7, 2026

Title Web Scraping and Bibliometric Analysis of MDPI Journals

Version 0.3.0

URL https://github.com/pgomba/MDPI_exploreR

Description Provides comprehensive tools to scrape and analyze data from the MDPI journals. It allows users to extract metrics such as submission-to-acceptance times, article types, and whether articles are part of special issues. The package can also visualize this information through plots. Additionally, 'MDPIexploreR' offers tools to explore patterns of self-citations within articles and provides insights into guest-edited special issues.

VignetteBuilder knitr

License CC BY 4.0

Encoding UTF-8

RoxygenNote 7.3.2

LazyData true

Imports dplyr, ggplot2, lubridate, magrittr, rvest, scales, stringr, tidyr

Suggests knitr, rmarkdown

NeedsCompilation no

Author Pablo Gómez Barreiro [aut, cre] (ORCID: <https://orcid.org/0000-0002-3140-3326>)

Maintainer Pablo Gómez Barreiro <pablogomezbr@hotmail.es>

Depends R (>= 3.5.0)

Repository CRAN

Date/Publication 2025-03-19 21:50:02 UTC

Contents

`agriculture` *Article data extracted from MDPI journal Agriculture*

Description

Article data extracted from MDPI journal Agriculture

Usage

```
agriculture
```

Format

`agriculture`**:**
A data frame with 7,160 rows and 7 columns:

i Article URL
article_type Article tyope classifier
Received Date article was submitted to journal
Accepted Date article was accepted for publication
tat Article turnaround time, or Accepted-Received
year Year the article was accepted
issue_type Type of issue where article is published ...

article_find *This function retrieves the URLs for all published articles from a specified journal. Users can provide the journal's code 'see MDPI_journals.rda', and the function will return the URLs of all articles available within the journal.*

Description

This function retrieves the URLs for all published articles from a specified journal. Users can provide the journal's code 'see MDPI_journals.rda', and the function will return the URLs of all articles available within the journal.

Usage

```
article_find(journal)
```

Arguments

journal	A string containing the name of a MDPI journal

Value

A vector (class: `character`) containing a list of articles URLs from target journal

Examples

```
## Not run:
agr_articles<-article_find("agriculture")

## End(Not run)
```

article_info *This function extracts key editorial information from one or more paper URLs. Specifically, it retrieves the submission, revision, and acceptance dates, as well as the article type. The function also calculates the turnaround time (the duration from submission to acceptance) and identifies whether the paper is part of a special issue.*

Description

This function extracts key editorial information from one or more paper URLs. Specifically, it retrieves the submission, revision, and acceptance dates, as well as the article type. The function also calculates the turnaround time (the duration from submission to acceptance) and identifies whether the paper is part of a special issue.

Usage

```
article_info(vector, sleep = 2, sample_size, show_progress = TRUE)
```

Arguments

vector	A vector with urls.
sleep	Number of seconds between scraping iterations. 2 sec. by default
sample_size	A number. How many papers do you want to explore from the main vector. Leave blank for all
show_progress	Logical. If TRUE, a progress bar is displayed during the function execution. Defaults to TRUE.

Value

A data frame (class: data.frame) with the following columns:

i The URL of the article from which the information is retrieved.

article_type The classification of the article (e.g., editorial, review).

Received The date the article was received by the publisher.

Revised The date the article was confirmed as revised by the publisher.

Accepted The date the article was accepted for publication.

tat The turnaround time, calculated as the number of days between the received and accepted dates.

year The year in which the article was accepted for publication.

issue_type Indicates whether the article is part of a special issue.

open_peer_review Indictes if article peer review is publicly available or not

Examples

```
url<-c("https://www.mdpi.com/2073-4336/8/4/45","https://www.mdpi.com/2073-4336/11/3/39")
## Not run:
info<-article_info(url, 1.5)

## End(Not run)
```

---

clean_names *This function will standardize the editors and authors names to facilitate matching them to one another.*

Description

Takes a vector of names to return the names without abbreviated middle names, academic titles and hyphens.

Usage

```
clean_names(name_vector)
```

Arguments

name_vector	A string with names separated by commas

Value

A vector (class: character) containing names

Examples

```
clean_names(c("Matthias M. Bauer","Thomas Garca Morrison","Wolfgang Nitsche", "Elias Biobaca L." ))
```

---

guest_editor_info *Obtain information from guest edited special issues*

Description

Deprecated: This function is deprecated and will be removed in a future version of the package. Use special_issue_info() instead. It extracts data from special issues, including guest editors' paper counts (excluding editorials), time between last submission and issue closure, and whether guest editors served as academic editors for any published papers.

Usage

```
guest_editor_info(journal_urls, sample_size, sleep = 2, show_progress = TRUE)
```

Arguments

journal_urls	A list of MDPI special issues URLs
sample_size	A number. How many special issues do you want to explore from the main vector. Leave blank for all
sleep	Number of seconds between scraping iterations. 2 sec. by default
show_progress	Logical. If TRUE, a progress bar is displayed during the function execution. Defaults to TRUE.

Value

A data frame (class: data.frame) with the following columns:

special_issue The URL of the special issue from which the information is retrieved.

num_papers Number of special issues contained in the special issue, not considering editorial type articles

flags Number of articles in the special issue with guest editorial pressence

prop_flag Proportion of articles in the special issue in which a guest editor is present

deadline Time at which the special issue was or will be closed

latest_sub Time at which last article present in the special issue was submitted

rt_sum_vector2 Numeric vector showing number of articles in which each individual guest editor is present

aca_flag Number of articles in the special issue where the academic editor is a guest editor too

d_over_deadline Day differential between special issue closure and latest article submission

Examples

```
## Not run:
ge_issue<-"https://www.mdpi.com/journal/plants/special_issues/5F5L5569XN"
ge_info<-guest_editor_info(ge_issue)

## End(Not run)
```

---

horticulturae *Article data extracted from MDPI journal Horticulturae*

Description

Article data extracted from MDPI journal Horticulturae

Usage

```
horticulturae
```

Format

`horticulturae`**:**

A data frame with 7,160 rows and 7 columns:

i Article URL

article_type Article tyope classifier

Received Date article was submitted to journal

Accepted Date article was accepted for publication

tat Article turnaround time, or Accepted-Received

year Year the article was accepted

issue_type Type of issue where article is published ...

MDPI_journals *MDPI journal names and code*

Description

Extracts names and codes of current MDPI journals.

Usage

```
MDPI_journals()
```

Value

A data frame (class: `data.frame`) with the following columns:

journal Full name of the MDPI journal

num_papers Journal code used for ID and web scraping purposes

Examples

```
## Not run:
journal_table<-MDPI_journals()

## End(Not run)
```

plot_articles *Plots information obtained from article_info(). For analysis purposes, Editorial and Correction type articles are ignored.*

Description

Plots information obtained from article_info(). For analysis purposes, Editorial and Correction type articles are ignored.

Usage

```
plot_articles(articles_info, journal, type)
```

Arguments

`articles_info`	Output dataframe from function articles_info.
`journal`	A string with the name of the journal for graph title purposes
`type`	select "summary","issues", "tat", "review" or "type" depending on desired graph

Value

A plot (class: ggplot) depicting the desired information obtained from article_info

Examples

```
plot_articles(agriculture,"Agriculture",type="summary")
```

---

selfcite_check *Calculates number of authors selfcitations against all references*

Description

Calculates number of authors selfcitations against all references

Usage

```
selfcite_check(article_url, verbose = TRUE)
```

Arguments

article_url	A valid MDPI article url
verbose	Logical. If TRUE, informative messages will be printed during the function execution. Defaults to TRUE.

Value

A string (class: data.frame)with the following columns:

selfcite The number of articles in references authored by any of the main article authors

total_ref Total number of references in the article

Examples

```
## Not run:
paper_url<-"https://www.mdpi.com/2223-7747/13/19/2785"
sc<-selfcite_check(paper_url)

## End(Not run)
```

special_issue_find *Retrieves all special issues of a specified journal with URLs. Filters results by issue status (open, closed, or all) and optional year range.*

Description

Retrieves all special issues of a specified journal with URLs. Filters results by issue status (open, closed, or all) and optional year range.

Usage

```
special_issue_find(journal, type = "closed", years = NULL, verbose = TRUE)
```

Arguments

journal	MDPI journal code
type	"closed", "open" or "all" special issues. "closed" by default.
years	A vector containing special issues closure dates to limit the search to certain years
verbose	Logical. If TRUE, informative messages will be printed during the function execution. Defaults to TRUE.

Value

A vector.

Examples

```
## Not run:
special_issue_find("covid")

## End(Not run)
```

special_issue_info *Obtain information from special issues*

Description

#' Extracts data from special issues, including guest editors' paper counts excluding editorials, time between last submission and issue closure, and whether guest editors served as academic editors for any published papers.

Usage

```
special_issue_info(journal_urls, sample_size, sleep = 2, show_progress = TRUE)
```

Arguments

journal_urls	A list of MDPI special issues URLs
sample_size	A number. How many special issues do you want to explore from the main vector. Leave blank for all
sleep	Number of seconds between scraping iterations. 2 sec. by default
show_progress	Logical. If TRUE, a progress bar is displayed during the function execution. Defaults to TRUE.

Value

A data frame (class: data.frame) with the following columns:

special_issue The URL of the special issue from which the information is retrieved.

num_papers Number of special issues contained in the special issue, not considering editorial type articles

flags Number of articles in the special issue with guest editorial pressence

prop_flag Proportion of articles in the special issue in which a guest editor is present

deadline Time at which the special issue was or will be closed

latest_sub Time at which last article present in the special issue was submitted

rt_sum_vector2 Numeric vector showing number of articles in which each individual guest editor is present

aca_flag Number of articles in the special issue where the academic editor is a guest editor too

d_over_deadline Day differential between special issue closure and latest article submission

Examples

```
## Not run:
ge_issue<-"https://www.mdpi.com/journal/plants/special_issues/plant-root"
speciali_info<-special_issue_info(ge_issue)

## End(Not run)
```

---

topic_find *Retrieves all topics of a specified journal with URLs. Filters results by issue status (open, closed, or all) and optional year range.*

Description

Retrieves all topics of a specified journal with URLs. Filters results by issue status (open, closed, or all) and optional year range.

Usage

```
topic_find(journal, type = "closed", years = NULL, verbose = TRUE)
```

Arguments

journal	MDPI journal code
type	"closed", "open" or "all" topics. "closed" by default.
years	A vector containing topics closure dates to limit the search to certain years
verbose	Logical. If TRUE, informative messages will be printed during the function execution. Defaults to TRUE.

Value

A vector.

Examples

```
## Not run:
topic_find("covid")

## End(Not run)
```

---

topic_info *Obtain information from guest edited topics*

Description

#' Extracts data from topics, including guest editors' paper counts excluding editorials, time between last submission and issue closure, and whether guest editors served as academic editors for any published papers. Includes names of journals participating in topic

Usage

```
topic_info(journal_urls, sample_size, sleep = 2, show_progress = TRUE)
```

Arguments

journal_urls	A list of MDPI topics URLs
sample_size	A number. How many topics do you want to explore from the main vector. Leave blank for all
sleep	Number of seconds between scraping iterations. 2 sec. by default
show_progress	Logical. If TRUE, a progress bar is displayed during the function execution. Defaults to TRUE.

Value

A data frame (class: `data.frame`) with the following columns:

topic The URL of the topics contained in the topic, not considering editorial type articles

flags Number of articles in the topic with guest editorial pressence

prop_flag Proportion of articles in the topic in which a guest editor is present

deadline Time at which the topic was or will be closed

latest_sub Time at which last article present in the topic was submitted

rt_sum_vector2 Numeric vector showing number of articles in which each individual guest editor is present

aca_flag Number of articles in the topic where the academic editor is a guest editor too

d_over_deadline Day differential between topic closure and latest article submission

journals List of journals participating in the topic

Examples

```
## Not run:
ge_issue<-"https://www.mdpi.com/topics/mechanisms_resistance_plant_diseases_volume"
ge_info<-topic_info(ge_issue)

## End(Not run)
```

Index