

# Package ‘heapsofpapers’

July 22, 2025

**Title** Easily Download Heaps of PDF and CSV Files

**Version** 0.1.0

**Description** Makes it easy to download a large number of files such as PDF files and CSV files, while automatically slowing down requests, letting you know where it is up to, and adjusting for files that have already been downloaded.

**License** MIT + file LICENSE

**URL** <https://github.com/RohanAlexander/heapsofpapers>

**BugReports** <https://github.com/RohanAlexander/heapsofpapers/issues>

**Imports** aws.s3, curl, dplyr, fs, magrittr, rlang, scales, utils

**Suggests** knitr, rmarkdown, spelling, testthat (>= 3.0.0), tibble

**VignetteBuilder** knitr

**Config/testthat/edition** 3

**Encoding** UTF-8

**Language** en-US

**LazyData** false

**RoxygenNote** 7.1.1

**NeedsCompilation** no

**Author** Rohan Alexander [aut, cre, cph] (ORCID:  
<<https://orcid.org/0000-0003-1279-0700>>),  
A Mahfouz [aut]

**Maintainer** Rohan Alexander <rohan.alexander@utoronto.ca>

**Repository** CRAN

**Date/Publication** 2021-08-23 18:40:01 UTC

## Contents

check_for_existence . . . . .	2
get_and_save . . . . .	3
<b>Index</b>	<b>5</b>

---

check\_for\_existence      *check\_for\_existence*

---

## Description

The `check_for_existence` function looks at the folder that you are going to save your PDFs to and checks whether you have already got any of them. It then suggests that you filter to remove them.

## Usage

```
check_for_existence(data, save_names = "save_names", dir = "heaps_of")
```

## Arguments

<code>data</code>	A dataframe that contains URLs that you want to download and the names that you want to save them as.
<code>save_names</code>	The name of the column whose values should be the saved file names where the downloaded file will be saved, <code>save_names</code> by default.
<code>dir</code>	The directory to download files to, current working directory by default.

## Value

The data dataframe with a column specifying whether the file has been downloaded.

## Examples

```
## Not run: two_pdfs <-
  tibble::tibble(
    locations_are = c("https://osf.io/preprints/socarxiv/z4qg9/download",
                      "https://osf.io/preprints/socarxiv/a29h8/download"),
    save_here = c("competing_effects_on_the_average_age_of_infant_death.pdf",
                  "cesr_an_r_package_for_the_canadian_election_study.pdf")
  )

heapsofpapers::get_and_save(
  data = two_pdfs,
  links = "locations_are",
  save_names = "save_here"
)

heapsofpapers::check_for_existence(data = two_pdfs, save_names = "save_here")

## End(Not run)
```

get\_and\_save

*get\_these\_and\_save\_them***Description**

The `get_and_save` function works with a tibble of locations (usually URLs) and file names, and then downloads the PDF from the location to the file name, saving as it goes, and letting you know where it is up to. It politely waits around 5 seconds between calls to the location, and skips locations that give an error.

**Usage**

```
get_and_save(
  data,
  links = "links",
  save_names = "save_names",
  dir = "heaps_of",
  bucket = NULL,
  delay = 5,
  print_every = 1,
  dupe_strategy = "overwrite"
)
```

**Arguments**

<code>data</code>	A dataframe that contains URLs that you want to download and the names that you want to save them as.
<code>links</code>	The name of the column whose values should be the URLs that you want to download, <code>links</code> by default.
<code>save_names</code>	The name of the column whose values should be the saved file names where the downloaded file will be saved, <code>save_names</code> by default.
<code>dir</code>	The directory to download files to, current working directory by default.
<code>bucket</code>	name of AWS S3 bucket to save files to.
<code>delay</code>	The number of seconds to wait between downloads, default (and minimum) is five seconds. We automatically add a bit of noise to lessen the effect on systematic processes that might be otherwise working.
<code>print_every</code>	The default is that you get a print message for every file, but you can change this. If you want to print an update for every second file then set this equal to 2, for a printed update every tenth file, set it to 10, etc.
<code>dupe_strategy</code>	There are a variety of ways of dealing with the situation where you already have some of the files downloaded. By default the function will just get them again and overwrite. However you can also specify 'ignore' in which case those files will be ignored. You can also investigate duplicates yourself using <code>heapsopa-pers::check_for_existence()</code> .

**Value**

A print statement in the console about whether each of the links was saved (if not turned off by the user), and notification that the function has finished.

**Examples**

```
## Not run: two_pdfs <-
tibble::tibble(
  locations_are = c("https://osf.io/preprints/socarxiv/z4qg9/download",
                    "https://osf.io/preprints/socarxiv/a29h8/download"),
  save_here = c("competing_effects_on_the_average_age_of_infant_death.pdf",
                 "cesr_an_r_package_for_the_canadian_election_study.pdf")
)

heapsofpapers::get_and_save(
  data = two_pdfs,
  links = "locations_are",
  save_names = "save_here"
)

## End(Not run)
```

# Index

`check_for_existence`, [2](#)

`get_and_save`, [3](#)