

# Using R for article screening - Systematic Review Management

Stephanie Towch

2024-03-19

## Introduction:

This is a guide on using R for article screening and systematic review management. In this document, we'll explore how the **revtools** package can streamline the systematic review process, making it easier for researchers to manage and analyse bibliographic data efficiently.

## Description:

The **revtools** package is designed for managing and analysing systematic review data. It offers a range of functions for importing, organising, screening, and summarising bibliographic data, facilitating efficient review processes. With **revtools**, researchers can easily handle large volumes of literature, streamline collaboration among reviewers, and generate comprehensive reports.

## Key Features:

- Import and manage bibliographic data from various formats (including .ris .bib and .csv files).
- Screen abstracts and full-text articles efficiently.
- Customise screening workflows and criteria.
- Generate summary statistics and visualisations.
- Facilitate collaboration and communication among review team members.
- Export data and results for further analysis or reporting.

## Who's it for?

The **revtools** package is useful for researchers, academics, and professionals conducting systematic reviews or meta-analyses in fields such as medicine, social sciences, and environmental studies.

## How to Install

Install the revtools package from CRAN using the following command: `install.packages("revtools")`

## More Information

Comprehensive documentation and tutorials are available on the package website and in the package vignettes.

- CRAN Documentation
- GitHub Repository
- Journal Article

### Example: Screening Titles and Abstracts:

To begin screening titles and abstracts using the `revtools` package, follow these steps:

1. Set the working directory to the location of your literature search files (add your file path).
2. Load the `revtools` package.

```
setwd("C:/Users/Steph/Desktop/PhD/Systematic Review/articles")  
library(revtools)
```

3. Read in the data file using the `read_bibliography` function. It can read different types of formats including `.ris`, `.bib` and `.csv` files.
4. Check the number of rows of data with the `nrow` function. This dataset should have 294 papers (rows).

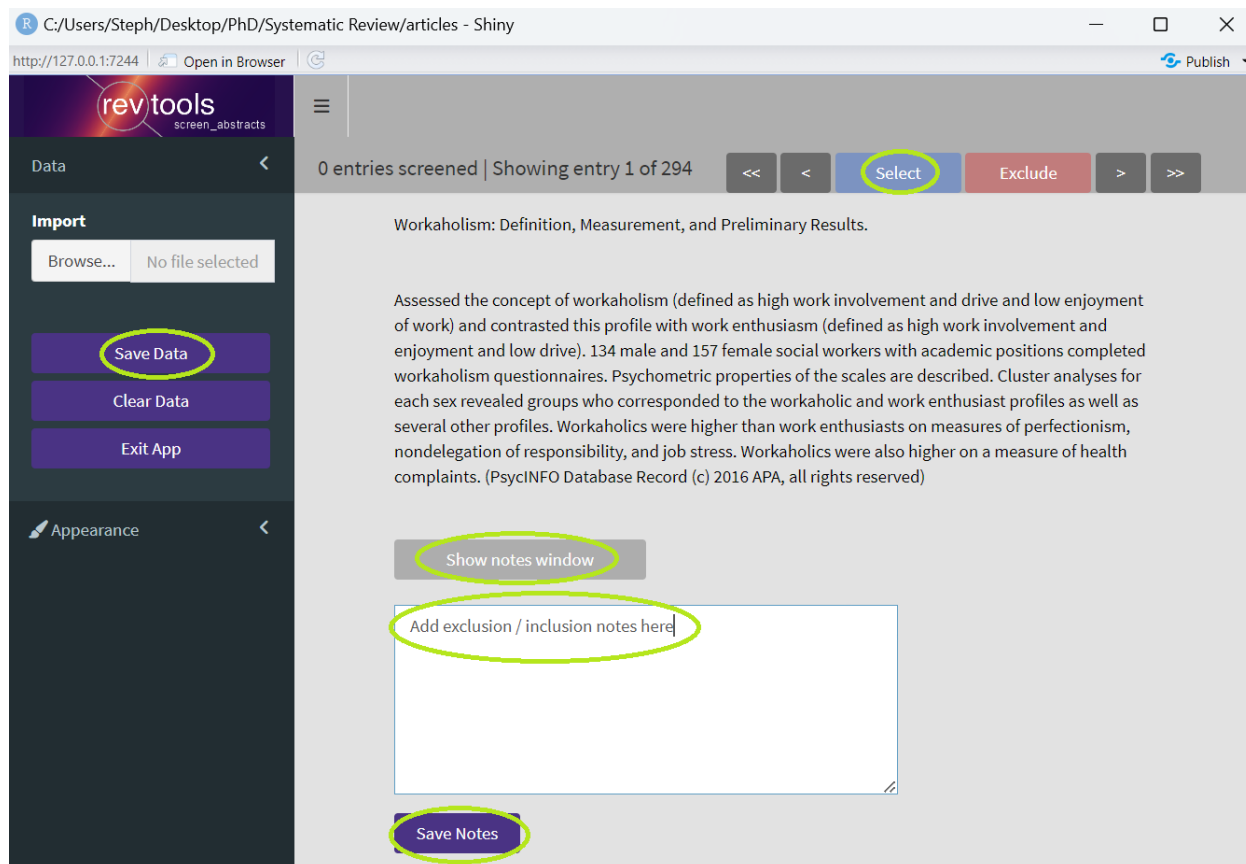
```
data <- read_bibliography("systematic_review_RISfile.ris")  
nrow(data)
```

5. Start screening titles and abstracts using the `screen_abstracts` function:

```
result <- screen_abstracts(data)
```

The Shiny app interface will appear where you can screen the titles and abstracts.

- Review the title and abstract against your inclusion/exclusion criteria.
- Click on 'Show Notes'.
- Add any notes relating to selecting or excluding the paper (it will append an extra column to the saved `.csv` file) and click the 'Save Notes' button (it doesn't automatically save).
- Either click on 'Select' or 'Exclude' the paper (it will append an extra column to the saved `.csv` file).



6. If you have a large number of papers to screen, click on 'Save Data', add your file name and save .csv file.

## Save As

### File Name

### File Type




7. You can continue screening a data file where you left off by loading the file you saved:

```
y <- read_bibliography("first_screen.csv")
result <- screen_abstracts(y)
```

- To review whether the papers were either 'Selected' or 'Excluded' and if any notes were added, check the last two columns of the saved .csv file.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Y	Z	
1	label	type	accession	author	title	journal	year	volume	issue	pages	publisher	issn	address	abstract	keywords	doi	screened_abstracts	notes	
37	Hoffman_Z	JOUR	2023-5241	Hoffman a	Navigating J	Divers Hi	2023	NA	NA	NA	Education	1938-8926	Pryor, Jone	Gender an	transgend	10.1037/dl	excluded	NWA	
38	Lyngfelt_2	JOUR	2023-4367	Lyngfelt an	Bridging At	Cambridge	2023		53	1	63-77	Taylor & Fr	0305-764X	Lyngfelt, At	The overar	Children's	10.1080/0i	excluded	Qualitative
39	Serrano-Fr	JOUR	2023-1462	Serrano-Fr	Work addik	Work	2021		68	3	779-788	IOS Press	1051-9815	Serrano-Fr	Backgroun	Workaholi	10.3233/W	selected	

### Example: Checking for Duplicate Papers

To screen for duplicate DOIs (Digital Object Identifiers) from multiple data files using the `find_duplicates` and `extract_unique_references` functions, follow these steps:

- Load in the data files using the `read_bibliography` function.
- Combine the data using the `merge_columns` function.
- Find duplicate DOIs within the dataset using the `find_duplicates` function.
- Extract the unique references from the dataset using the `extract_unique_references` function.

```
data <- read_bibliography("systematic_review_RISfile.ris")
nrow(data) # 294 papers

data2 <- read_bibliography("Google_Scholar_Data_14.ris") #add in google scholar data
nrow(data2) # 14 papers

combined_data <- merge_columns(data,data2)
nrow(combined_data) # 308 papers

# Find duplicated DOIs within the dataset
doi_match <- find_duplicates(combined_data,
                             match_variable = "doi",
                             group_variables = NULL,
                             match_function = "exact"
)

# Automatically extract one row per duplicate
data_unique <- extract_unique_references(combined_data, doi_match)
nrow(data_unique) #should be 308 papers i.e. no duplicates found
```

### Example: Divide screening papers among team members

To allocate screening papers among review team members using the `allocate_effort` and `distribute_tasks` functions, follow these steps to save a .csv file per reviewer depending on the amount of allocated effort you assigned.

- Load in the dataset using the `read_bibliography` function.
- Allocate effort among reviewers using the `allocate_effort` function. **Note:** The proportions should equate to 1.
- Distribute the search data among reviewers using the `distribute_tasks` function. This will generate a separate .csv file per reviewer depending on how much allocated effort has been assigned.

```
# Load in search data
data <- read_bibliography("systematic_review_RISfile.ris")
nrow(data)

# Allocate effort among reviewers
result <- allocate_effort(
  reviewers = c("Halley", "Steph"),
  effort = c(0.1, 0.9), #proportion allocated to each reviewer. should equal 1
  max_reviewers = 2,
  quiet = TRUE
)

# distribute search data among reviewers
distribute_tasks(data, result, write_csv = TRUE,
  file_name = "reviewer.csv", return_data = FALSE)
```

4. Each reviewer can then screen the titles and abstracts by loading their assigned .csv file and utilising the `screen_abstracts` function, which initiates the Shiny app interface.

```
# Now screen data allocated to each reviewer
Steph_data <- read_bibliography("reviewer_Steph.csv")
Steph_result <- screen_abstracts(Steph_data)

Halley_data <- read_bibliography("reviewer_Halley.csv")
Steph_result <- screen_abstracts(Halley_data)
```

## More Information

For further details and functionality, see Martin Westgate's (2019) paper who created the `revtools` package.

- Westgate, M. J. (2019). Managing systematic literature reviews with `revtools` for R. *Journal of the Royal Statistical Society: Series Methods in Ecology and Evolution*, 10(3), 134-148.

## Conclusion

The `revtools` package offers a powerful set of tools for managing systematic review data effectively. By leveraging its functionalities, researchers can streamline their review processes, collaborate more efficiently, and produce high-quality systematic reviews.