Split reference list helper for pilot and collaborative screening rounds

Coralie Williams

2022-12-02

Contents

When screening for a systematic review or meta-analysis, we conduct several pilot screening rounds. Pilot screenings help us refine our search string, decision tree, and increase the overall accuracy of our screening[1].

During a pilot screening, we want to select a random subset of references that would be a representative sample of the full set to determine whether our hit rate is sufficient. When possible, screening rounds are conducted in collaboration with another reviewer (and sometimes more). To speed up the screening process and avoid biases we want to randomly allocate a subset of papers to each collaborator by splitting a reference list in several subsets.

There are two reasons we'd want to automate the selection and splitting of a reference list:

1. It is time consuming to randomly select papers (>100 papers is tedious to select by hand!)
2. We are not really good at selecting things at random (actually computers aren't really good at being truly random either*)

Below is the R (www.r-project.org) code to run two functions that may come in useful when conducting your pilot and collaborative screenings with Rayyan (https://rayyan.ai/), or any other software where you can upload your pilot reference list.

These functions will help you quickly obtain a subset of references for pilot screenings, and to randomly split a references list for collaboration between two reviewers. I am working on some improvements and additional functionalities (i.e. split between multiple collaborators), so stay tuned…

1. Select random pilot set:

Load the function

First, load the `getpilotref` function below in your environment:

```
# ------------------------------------
# getpilotref function
# ------------------------------------
## Description:
#     Function to obtain a random subset of references for pilot screening.
#
# Arguments
# - x: data frame with reference list
# - n: number of papers for pilot subset (default is 10)
# - write: logical argument whether to save the pilot list as a csv file in current working directory (
# - fileName: name of file (default is "pilot")

getpilotref <- function(x, n=10, write=FALSE, fileName="pilot"){

  if (length(n) == 1L && n%%1==0 && n>0 && n<=nrow(x)) {

    # sample randomly the vector n of row indexes and remove id column in the final dataset
    x$ids <- 1:nrow(x)
    pdat <- x[which(x$ids %in% sample(x$ids, n)),]
    pilot <- pdat[,-which(colnames(pdat)=="ids")]

    } else {
      # error message n value provided is not valid
      stop("Incompatible value n supplied, please check. n must be a positive integer no higher than th
    }

  if (write==T){

    # save generated pilot list in working directory using the name provided
    write_csv(pilot, paste(fileName, ".csv", sep=""), na="")

    # print out summary of saved file name
    cat(paste("Pilot random sample set of ", n, " articles is saved as: ", fileName, ".csv", sep=""))

  }

  return(pilot)
}
```

Let's try it out

For the purpose of demonstration, here is an example reference list, an exported csv file from Rayyan:

```
# Read example butterfly reference list
articles<-read.csv("https://raw.githubusercontent.com/coraliewilliams/2022/main/data/articles_butterfly
```

First, let's obtain a random reference set of 10 papers without saving it as a csv file:

```
p10 <- getpilotref(articles)
```

Now, let's obtain a subset of 100 papers for a pilot screening and save the subset as a csv file called "pilot100.csv". Make sure you have the `readr` package installed and loaded in your environment.

```
library(readr)
p100 <- getpilotref(articles, n=100, write=T, fileName="pilot100")
```

```
## Pilot random sample set of 100 articles is saved as: pilot100.csv
```

This will save a csv file *pilot100.csv* in your working directory. If you are unsure where is your working directory run this command `getwd()` in your console.

2. Split reference list with another collaborator

Load the `splitref_prop` function in your environment:

```
# ------------------------------------
# splitref_prop function
# ------------------------------------
## Description:
#     Function to split in two a reference list based on input proportions.
#
## Arguments:
# - x: data frame with reference list
# - p: vector of two numerical proportions for each split, it must have two positive numerical values t
# - write: logical argument whether to save the pilot list as csv in current working directory.
# - fileName: name to give to the suffix of the two split csv files.

splitref_prop <- function(x, p=c(0.5, 0.5), write=F, sname="split") {

    if (length(p) == 2L && is.numeric(p) && sum(p) == 1 && all(p > 0)) {

      # randomly allocated a numerical id to each reference
      rids <- sample(1:nrow(x))

      # get index of row to split on using the proportion values provided
      spl <- floor(p[-length(p)] * nrow(x))

      # get indices of two data frames based on split ids
      indx1 <- rids[1:spl]
      indx2 <- rids[(spl + 1):nrow(x)]

      # save split subsets in two separate datasets
      split1 <<- x[indx1,]
      split2 <<- x[indx2,]

      # print out summary message
      cat(paste(c("Reference list was randomly split into",length(p), "proportions of", p[1]*100, "% an

      if (write == T) {
        # save files
        write_csv(split1, paste(sname, "_set1", ".csv", sep = ""), na ="")
        write_csv(split2, paste(sname, "_set2", ".csv", sep = ""), na ="")

        }
      } else {
      # error message if provided n value is not valid
      stop("Incompatible values for p (proportions) supplied, please check.
           Proportion values must be positive integers less than 1, and the total sum of all proportions

    }
}
```

Let's try it out

Let's try it out on the example butterfly reference list. First let's split the reference list in two equal splits (50% each):

```
splitref_prop(articles)
```

```
## Reference list was randomly split into 2 proportions of 50 % and 50 %
```

This will give you two separate data frames to share between two reviewers: `split1` and `split2`

Now let's get 30% of references in the first subset (`split1`) and 70% in the second subset (`split2`), for example if one reviewer has more time to spend on the screening:

```
splitref_prop(articles, p=c(0.3,0.7))
```

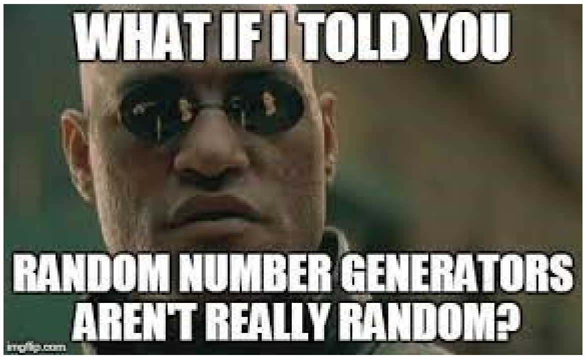
```
## Reference list was randomly split into 2 proportions of 30 % and 70 %
```

Let's save the 30% and 70% split list of references as csv files with the suffix "testsplit":

```
splitref_prop(articles, p=c(0.3,0.7), write=T, sname="testsplit")
```

```
## Reference list was randomly split into 2 proportions of 30 % and 70 %
```

This will save two csv files *testsplit_set1.csv* and *testsplit_set2.csv* in your working directory.



*random number generators from most computer programs are actually "pseudo-random", meaning they are produced from a deterministic mathematical model or algorithm. Pseudo-random number generators are usually good enough for their intended purpose (basically better than what any human could do). A good pseudo-random number generator will reproduce statistics that are consistent with true randomness, but they are not truly random. A truly random number can be generated from a constantly changing physical process that can't be modeled as an algorithm. If you're curious about true randomness check out these websites: https://www.random.org/; https://qrng.anu.edu.au/random-colours/

Any comments, questions or feedback, please contact me at coralie.williams@unsw.edu.au

[1] Foo,Y. Z., O'Dea, R. E., Koricheva, J., Nakagawa, S., & Lagisz, M. (2021). A practical guide to question formation, systematic searching and study screening for literature reviews in ecology and evolution. *Methods in Ecology and Evolution*, *12*(9), 1705–1720. https://doi.org/10.1111/2041-210X.13654