Coding sample - R

Alexia Witthaus Viñé

2022-12-23

```
#Link to Github repo:
#https://github.com/alexiawitthaus/codingsampleR/blob/main/Coding%20sample%20R.Rmd

setwd("~/Library/CloudStorage/OneDrive-EmoryUniversity/Fall 22-Alexia's MacBook Air/Qtm research")

#import data
experts<- as_tibble(read.csv( "academicexpert_articles.csv"))
#exclude article without doi
experts_DOI<- filter(experts, !is.na(experts$DOI))
#exclude repetitive doi
experts_DOI_unique <- distinct(experts_DOI, experts_DOI$DOI, .keep_all = TRUE)
```

```
#Write functions to make data cleaning easier

### Function that counts the number of references in for each article found
count_num_ref<- function(request){
      references<- request$referenced_works
      num_references<- sum(!is.na(as.data.frame(references)))
      return(num_references)
}


### Function that keeps IDs of NAs preserved over iterations
vec_no_oa<- function(num_iter, list){
  dois_with_no_OpenAlex <- unlist(list)
  dois_with_no_OpenAlex <-
unique(
  dois_with_no_OpenAlex[dois_with_no_OpenAlex %in%
  unique(dois_with_no_OpenAlex)[table(dois_with_no_OpenAlex) == num_iter]])
  return(dois_with_no_OpenAlex)
}
```

```
#Create empty list for DOIs that don't have OpenAlexID
dois_with_no_OpenAlex <- list()


# Add columns to the expert_DOI database to add number of referenced work in each article
experts_DOI_unique <- experts_DOI_unique %>% mutate(openalex_id = NA,
                                                    num_references = NA)

#Account for random error in OpenAlexR package
```

```
num_iterations <- 3

for(iteration in 1:num_iterations){

  for (i in 1:30){
    doi <- experts_DOI_unique$DOI[i]
    #Runs for every article in iteration 1, or each article that is within the list of NAs
    if(iteration == 1 | (iteration > 1 & doi %in% dois_with_no_OpenAlex )){
        #Call the API
        oa_request <- oa_fetch(doi = doi)
        #If Null, fill columns created earlier with NA, and append article to list
        if(is.null(oa_request)){
        experts_DOI_unique$openalex_id[i]<- NA
        experts_DOI_unique$num_references[i] <- NA
        dois_with_no_OpenAlex <- append(dois_with_no_OpenAlex, doi)
        # Sum articles found, and append the number to the columns coded earlier
        }else{
        num_ref <- count_num_ref(request = oa_request)
        experts_DOI_unique$num_references[i] <- num_ref
        experts_DOI_unique$openalex_id[i] <- oa_request$id
        }
      }
    }
  }
```

```
#DOIs that don't have OpenAlexID
dois_with_no_OpenAlex <- vec_no_oa(num_iterations, dois_with_no_OpenAlex)
print(dois_with_no_OpenAlex)
```

```
## [1] "10.32674/jis.v11i2.459"
```

```
set.seed(12345)
random_number <- sample.int(30, 5)
#Number of references in each article, random sample of 5 rows
experts_DOI_unique[random_number , c('DOI', 'openalex_id', 'num_references')]
```

```
## # A tibble: 5 x 3
##   DOI                          openalex_id                       num_references
##   <chr>                        <chr>                                      <int>
## 1 10.1016/j.worlddev.2005.05.004 https://openalex.org/W1972591045            15
## 2 10.1111/saje.12162           https://openalex.org/W2618106908            41
## 3 10.1016/j.worlddev.2013.04.009 https://openalex.org/W2030286149           22
## 4 10.1016/j.jpolmod.2021.03.008  https://openalex.org/W3158110247           25
## 5 10.1016/S0305-750X(02)00118-3  https://openalex.org/W2125572538           19
```