Programs

true

2021-11-30

Programs

Setup

Most parameters are set in the `config.R`:

```
source(file.path(rprojroot::find_rstudio_root_file(),"pathconfig.R"),echo=TRUE)
```

```
##
## > basepath <- rprojroot::find_rstudio_root_file()
##
## > dataloc <- file.path(basepath, "data", "replication_data")
##
## > interwrk <- file.path(basepath, "data", "interwrk")
##
## > hindexloc <- file.path(basepath, "data", "h_index_data")
##
## > crossrefloc <- file.path(basepath, "data", "crossref")
##
## > TexBase <- file.path(basepath, "text")
##
## > TexIncludes <- file.path(basepath, "text", "includes")
##
## > Outputs <- file.path(basepath, "analysis")
##
## > notes <- file.path(basepath, "text", "hautahi_notes")
##
## > programs <- file.path(basepath, "programs")
##
## > for (dir in list(dataloc, interwrk, hindexloc, crossrefloc,
## +     TexIncludes, Outputs)) {
## +     if (file.exists(dir)) {
## +     }
## +     else {
## +     .... [TRUNCATED]
##
## > MRAN.snapshot <- "2019-01-01"
##
## > options(repos = c(CRAN = paste0("https://mran.revolutionanalytics.com/snapshot/",
## +     MRAN.snapshot)))
```

```
source(file.path(programs,"config.R"), echo=TRUE)
```

```
##
## > HindexRaw <- "h-index-assignment1.csv"
##
## > HindexClean <- "hindex.csv"
```

```
##
## > zenodo.id <- "2639919"
##
## > zenodo.id <- "2639920"
##
## > zenodo.api = "https://zenodo.org/api/records/"
##
## > citations.key <- "18vujiGq3FPgvpwop7ND-EA8kGBscStUQO79sOR16Y0k"
```

Note that the path `interwrk` is transitory, and is only kept during processing. It will be empty in the replication archive.

Any libraries needed are called and if necessary installed through `libraries.R`:

```
source(file.path(basepath,"global-libraries.R"),echo=TRUE)
```

```
##
## > mran.date <- "2021-10-01"
##
## > get_os <- function() {
## +     sysinf <- Sys.info()
## +     if (!is.null(sysinf)) {
## +         os <- sysinf["sysname"]
## +         if (os == "Darwin")
## +   .... [TRUNCATED]
##
## > ifelse(get_os() == "linux", options(repos = c(REPO_NAME = paste0("https://packagemanager.rstudio.c
## +     mran.date, "+Y3J ..." ... [TRUNCATED]
## [[1]]
##                                                          CRAN
## "https://mran.revolutionanalytics.com/snapshot/2019-01-01"
##
##
## > pkgTest <- function(x) {
## +     if (!require(x, character.only = TRUE)) {
## +         install.packages(x, dep = TRUE)
## +         if (!require(x, charact .... [TRUNCATED]
##
## > global.libraries <- c("dplyr", "devtools", "rprojroot",
## +     "tictoc", "ggplot2")
##
## > results <- sapply(as.list(global.libraries), pkgTest)

## Loading required package: dplyr

## Warning: As of rlang 0.4.0, dplyr must be at least version 0.8.0.
## * dplyr 0.7.8 is too old for rlang 0.4.11.
## * Please update dplyr with `install.packages("dplyr")` and restart R.

##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
```

```
##     intersect, setdiff, setequal, union

## Loading required package: devtools

## Warning in register(): Can't find generic `is_informative_error` in package
## testthat to register S3 method.

## Warning in register(): Can't find generic `is_informative_error` in package
## testthat to register S3 method.

## Loading required package: rprojroot

## Loading required package: tictoc

## Loading required package: ggplot2
```

```
source(file.path(programs,"libraries.R"), echo=TRUE)
```

```
##
## > libraries <- c("dplyr", "devtools", "rcrossref", "readr",
## +     "tidyr", "data.table", "rjson", "ggplot2", "Rcpp")
##
## > results <- sapply(as.list(libraries), pkgTest)

## Loading required package: rcrossref

## Loading required package: readr

## Loading required package: tidyr

## Loading required package: data.table

##
## Attaching package: 'data.table'

## The following objects are masked from 'package:dplyr':
##
##     between, first, last

## Loading required package: rjson

## Loading required package: Rcpp

##
## > cbind(libraries, results)
##       libraries    results
##  [1,] "dplyr"      "OK"
##  [2,] "devtools"   "OK"
##  [3,] "rcrossref"  "OK"
##  [4,] "readr"      "OK"
##  [5,] "tidyr"      "OK"
##  [6,] "data.table" "OK"
##  [7,] "rjson"      "OK"
##  [8,] "ggplot2"    "OK"
##  [9,] "Rcpp"       "OK"
##
## > libraries3 <- c("magick", "summarytools")
##
## > if (get_os() == "linux") {
## +     libraries2 <- c("Rcpp")
## +     results2 <- sapply(as.list(libraries2), pkgTest)
## +     pkgTestSrc <- function(x) {
```

```
## +  .... [TRUNCATED]
```

```
## Loading required package: magick
```

```
## Linking to ImageMagick 6.9.12.3
## Enabled features: cairo, fontconfig, freetype, heic, lcms, pango, raw, rsvg, webp
## Disabled features: fftw, ghostscript, x11
```

```
## Loading required package: summarytools
```

```
## [1] "OK" "OK"
```

Data cleaning and merging

We combine our collected data with bibliometric data, both manually extracted from "Web of Science" and collected from CrossRef. We also download the cleaned data for both the replication and the Web of Science data here. These programs should be runnable by anybody.

Download the replication data from Zenodo

The responses to the replication attempts are stored on Google Sheets, and considered private. We have separately cleaned the data, anonymized it, and uploaded to Zenodo (see https://www.github.com/labordynamicsinstitute/ldi-replication-dataprep). Here, we simply download the data, with a bit of additional data cleaning.

- Input data: On Zenodo
- Output data: path 'interwrk','repllist2.Rds"

```
source(file.path(programs,"01_download_replication_data.R"),echo=TRUE)
```

At the end of this step, the `interwrk` directory should have the following data files:

- entryQ_pub.Rds - the main data from the "Entry" questionnaire (assessment)
- exitQ_pub.Rds - the main data from the post-replication "Exit" questionnaire (assessment)
- replication_list_pub.Rds - the assignment spreadsheet.

Get CrossRef information

The master replication list has all the DOIs. We look up the DOI at CrossRef.

Note: downloading references from CrossRef can take a while. It is set to `eval=FALSE` and needs to be run manually.

- inputs: entryQ, exitQ, replication list (in `dataloc`)
- outputs: crossref_info.Rds (in `crossrefloc`) and intermediate raw data in `interwrk`

```
source(file.path(programs,"02_get_crossref.R"),echo=TRUE)
```

Some diagnostics

When finding DOIs, some articles might not be found. When that is the case, they are reported here.

```
if ( file.exists(file.path(interwrk,"crossref.diagnostics.Rds"))) {
  crossref.diagnostics <- readRDS(file=file.path(interwrk,"crossref.diagnostics.Rds"))
} else {
```

```
  crossref.diagnostics <- data.frame()
}
```

- DOIs to download (unique DOIs in all replication files): r NROW(dois.df)

- DOIs successfully looked up on CrossRef: r nrow(bibinfo.df)

- DOIs not found: r nrow(crossref.diagnostics) (should be ZERO)

DOI
10.1257
10.1257/mac.4.2..218
aej-policy-2
10.1257/mic.6.4.237
10.1257/mic.6.4.362
10.1257/mic.6.1.182
AEJPOLICY-10
10.1257/app.20150057
NA

```
source(file.path(programs,"04_clean_replicationlist.R"),echo=TRUE)
```

Download the h-index information

```
source(file.path(programs,"06_gen_hindex_list.R"),echo=TRUE)
```

```
source(file.path(programs,"07_readclean_hindex_list.R"),echo=TRUE)
save.image("./data/interwrk/my_work_space.RData")
```

Paper analysis

The actual paper analysis is done as part of the paper itself (in `knitr` format), see ../text/README.md for more details.

Note that the path `interwrk` is transitory, and is only kept during processing. It will be empty in the replication archive.