R Script for Scraping PlumX Data

Abraham Cheung

2023-10-11

Contents

This script scrapes the PlumX metric counts and policy citations of research articles from the Social Science Research Network (SSRN) and Science Direct (SciDir). I use the `RSelenium` package to access the websites. Then I save the website as HTML to extract information using `rvest` package. I created six functions to scrape the PlumX data. The initial blocks of code define the function. The last block executes the functions.

The primary input is a CSV export of the Zotero citations. The script depends on using the CSV-generated Zotero list. The final output is 4 data frames of data as CSVs:

1. CSV of PlumX metric counts for SSRN articles
2. CSV of PlumX metric counts for SciDir articles
3. CSV of PlumX policy citations for SSRN articles
4. CSV of PlumX policy citations for SciDir articles

Libraries

```
pacman::p_load(tidyverse,rvest,lubridate,RSelenium,wdman,netstat,install = FALSE,update = FALSE)
```

Set Up Selenium

```
driver <- rsDriver(browser = "firefox",port=free_port(),verbose = F)
remDr <- driver$client
```

Load Zotero CSV

```
zotero_csv = read.csv("test_zotero_list.csv")
```

Set Up Functions

These functions assist in future functions.

```
# function to accept cookies for SSRN
accept_cks_ssrn <- function() {
  suppressMessages(
    try(
        remDr$findElement("css selector","button#onetrust-accept-btn-handler")$clickElement(),
        silent = TRUE))
}

# function for switching windows
switch_windows <- function(window_num) {
  all_windows = remDr$getWindowHandles()
  remDr$switchToWindow(all_windows[[window_num]])
}
```

ssrn_plum_html()

This function opens the SSRN article link, navigates to the PlumX page for the SSRN article, and then saves and returns the website as HTML.

- input: one SSRN link
- output: PlumX page for the SSRN link as HTML

```
ssrn_plumx_html <- function(ssrn_lnk) {

  ## 1. Navigate to the link
  remDr$navigate(ssrn_lnk)
  writeLines(paste("\nSuccessfully opened SSRN link:",ssrn_lnk, "\nWaiting 5 seconds for the page to co
  Sys.sleep(5)
  accept_cks_ssrn()
  Sys.sleep(5)

  ## 2. Open PlumX page
  remDr$findElement("css","a.plx-wrapping-print-link")$clickElement()
  writeLines("Successfully opened PlumX page \nWaiting 5 seconds for the page to completely load...")
  Sys.sleep(5)
```

```
  ## 3. Switch windows
  # I did not use the `switch_windows()` function since it returns this error. "can only open URLs for
  all_windows = remDr$getWindowHandles()
  remDr$switchToWindow(all_windows[[2]])
  writeLines("Successfully switched windows")
  accept_cks_ssrn()

  ## 4. Read HTML of PlumX page
  plumx_pg =
    remDr$getPageSource()[[1]] %>%
    read_html()
  writeLines("Successfully extracted HTML of SSRN PlumX page")

  return(plumx_pg)
}
```

scidir_plumx_html()

This function opens the SciDir article link, navigates to the PlumX page for the SciDir article, and then saves and returns the website as HTML.

- input: one Science Direct link
- output: PlumX page for the Science Direct link as HTML

```
scidir_plumx_html <- function(scidir_lnk) {

  ## 1. Navigate to the link
  remDr$navigate(scidir_lnk)
  writeLines(paste("\nSuccessfully opened Science Direct link: ",scidir_lnk, "\nWaiting 5 seconds for t
  Sys.sleep(5)

  ## 2. Scroll down then open PlumX page
  remDr$findElement("css", "body")$sendKeysToElement(list(key = "page_down"))
  Sys.sleep(3)
  remDr$findElement("css", "svg.svg-arrow")$clickElement()
  writeLines("Successfully opened PlumX page \nWaiting 5 seconds for the page to completely load...")
  Sys.sleep(5)

  ## 3. Switch windows
  all_windows = remDr$getWindowHandles()
  remDr$switchToWindow(all_windows[[2]])
  writeLines("Successfully switched windows")
  accept_cks_ssrn()

  ## 4. Read HTML of PlumX page
  plumx_pg =
    remDr$getPageSource()[[1]] %>%
    read_html()
  writeLines("Successfully extracted HTML of Science Direct PlumX page")

  return(plumx_pg)
}
```

count_plumx()

This function gathers article information and the PlumX metrics for any PlumX page as HTML.

- input: PlumX page as HTML
- output: dataframe of PlumX metrics including article title, DOI, PlumX URL, timestamp, and PlumX counts for “Citations”,“Usage”, “Captures”, and “Social Media”.

```
count_plumx <- function(plumx_pg = plumx_pg) {

  ## 1. Gather article information
  article_title =
    plumx_pg %>%
    html_element("h1.artifact-title") %>%
    html_text()
  writeLines(paste("\nArticle title:",article_title))

  doi_txt =
    plumx_pg %>%
    html_element("span.anchor-text[data-reactid='2']") %>%
    html_text()
  writeLines(paste("Article DOI:",doi_txt))

  url_plumx = remDr$getCurrentUrl() %>% unlist()
  writeLines(paste("Article URL:",url_plumx))

  # create df with this info
  info_df = data.frame("article_title" = article_title, "doi_scrape" = doi_txt, "url_plumx" = url_plumx


  ## 2. Obtain CSS selectors for counts
  # This step gathers the four header names ("Citations","Usage", "Captures", and "Social Media"). Thes

  # find attribute name of the 4 names
  css_select =
    plumx_pg %>%
    html_element("div.card-content") %>%
    html_children() %>%
    html_children() %>%
    html_attrs() %>%
    unlist()

  # ignore the 'Ratings' heading
  # This heading is the fifth one, but we are ignoring it.
  no_rtgs = str_detect(css_select,"rating", negate = TRUE)
  css_select = css_select[no_rtgs]

  # create CSS selector from class names. make the CSS selectors usable in following steps in rvest.
  css_select = str_c("li.",css_select)
  css_select = str_replace_all(css_select," ",".")
  writeLines(paste("CSS selectors:",css_select))


  ## 3. Scrape the metric counts
```

```
  counts_df = data.frame()
  for (i in css_select) {

    # select element in HTML using the given CSS selector
    metric_card =
      plumx_pg %>%
      html_element(i) %>%
      html_text()

    # remove commas from text and numbers as characters
    metric_card = str_replace_all(metric_card,",","")

    # identify category name (one of the four) which is the last phrase in CSS selector
    class_catg = str_split_1(i, "\\-") %>% last()

    # extract category names and values in a dataframe
    category = str_extract_all(metric_card, "[A-Za-z \\-&]+") %>% unlist()
    category = str_c(category,class_catg, sep = "_") %>% str_replace_all(" ", "_")
    value = str_extract_all(metric_card, "[\\d]+") %>% unlist()

    temp_df = data.frame("Category" = category,"Values" = value)

    # remove categories that are labelled "SSRN". The duplicates will prevent the pivot from working co
    temp_df = filter(temp_df,!grepl("SSRN", temp_df$Category))

    # bind output from each iteration of the loop
    counts_df = bind_rows(counts_df,temp_df)

    writeLines(paste("Successfully gathered counts for", i))
  }


  ## 4. Combine metric counts and info
  counts_wide = pivot_wider(counts_df, names_from = Category, values_from = Values)
  output_df = cbind(info_df, counts_wide)


  ## 5. Close tab and switch
  remDr$closeWindow()
  switch_windows(1)
  writeLines("Closed PlumX tab. Returned to first tab.")

  return(output_df) # this is the final output

}
```

ssrn_plumx_counts()

This function loops through multiple SSRN links and combines the PlumX counts and info from each link into one dataframe.

- input: all SSRN links from the Zotero CSV

- output: dataframe of the PlumX metrics for each link

```
ssrn_plumx_counts <- function(zotero_csv) {
  output_df <- data.frame()

  ## 1. Filter for SSRN links
  ssrn_links =
    zotero_csv %>%
    filter(grepl("ssrn",Url)) %>%
    select(Url) %>%
    unlist
  writeLines(paste("Starting scrape for", length(ssrn_links),"SSRN links"))

  for (link in ssrn_links) {

    ## 2. Obtain PlumX counts from HTML
    writeLines(paste("Begin scraping data for",link))
    temp_df <- ssrn_plumx_html(ssrn_lnk = link) %>% count_plumx()

    ## 3. Combine the result with previous counts
    output_df <- bind_rows(output_df,temp_df)
    writeLines(paste("Number of entries in final output:",nrow(output_df)))
  }
  return(output_df)
}
```

scidir_plumx_counts()

This function loops through multiple SciDir links and combines the PlumX counts and info from each link into one dataframe.

- input: all SciDir links from the Zotero CSV
- output: dataframe of the PlumX metrics for each link

```
scidir_plumx_counts <- function(zotero_csv) {
  output_df <- data.frame()

  ## 1. Filter for Science Direct links
  scidir_links =
    zotero_csv %>%
    filter(grepl("sciencedirect",Url)) %>%
    select(Url) %>%
    unlist
  writeLines(paste("Beginning scrape for", length(scidir_links),"Science Direct links"))

  for (link in scidir_links) {

    ## 2. Obtain PlumX counts from HTML
    writeLines(paste("Begin scraping data for ",link))
    temp_df <- scidir_plumx_html(scidir_lnk = link) %>% count_plumx()

    ## 3. Combine the result with previous counts
```

```
    output_df <- bind_rows(output_df,temp_df)
    writeLines(paste("Number of entries in final output:",nrow(output_df)))

  }
  return(output_df)
}
```

count_plcy_cit()

This function scrapes the policy citation information for articles with policy citations.

- input: dataframe of the PlumX metric counts
- output: dataframe of the policy citations for each article

```
count_plcy_cit <- function(plumx_counts_df) {

  ## 1. Filter for the links with Policy Citations
  links =
    plumx_counts_df %>%
    drop_na(Policy_Citations_citation) %>%
    select(url_plumx) %>%
    unlist
  writeLines(paste("There are",length(links),"articles with Policy Citations"))

  output_df = data.frame()
  for (link in links) {


    ## 2. Open URL and obtain basic article info
    remDr$navigate(link)
    Sys.sleep(5)

    # article title
    webElem = remDr$findElements("css", "h1.artifact-title")
    article_title = unlist(lapply(webElem, function(x) x$getElementText()))
    writeLines(paste("\nArticle title:",article_title))

    # doi
    webElem = remDr$findElements("css", "span.anchor-text[data-reactid='2']")
    doi_txt = unlist(lapply(webElem, function(x) x$getElementText()))[1]
    writeLines(paste("Article DOI:",doi_txt))


    # 3. Navigate to the Policy Citations page
    remDr$findElement("css","button.button-link.button-link-secondary")$clickElement()

    plc_lnk = remDr$getCurrentUrl() %>% unlist()
    plc_pg = remDr$getPageSource()[[1]] %>% read_html()
    writeLines(paste("Policy Citation page link:",plc_lnk))


    ## 4. Obtain citation info
```

```
  # scrape citation titles
  citation_title =
    plc_pg %>%
    html_elements("h3.card-title") %>%
    html_text()

  # scrape citation date
  cit_metadata =
    plc_pg %>%
    html_elements("ul.card-metadata") %>%
    html_text()

  citation_date = str_extract(cit_metadata, ".*(\\d)")

  # authors of policy citation
  authors = str_extract(cit_metadata, "(?<=by ).*")

  # publisher name
  publisher = str_extract(cit_metadata,"(?<=\\d{4}).*(?= by)")

  # publisher url
  publisher_url =
    plc_pg %>%
    html_elements("ul.card-metadata") %>%
    html_elements("a.anchor.anchor-external-link") %>%
    html_attr("href")

  # policy citation url
  citation_url =
    plc_pg %>%
    html_elements("a.anchor.anchor-button.text-s.anchor-external-link") %>%
    html_attr("href")

  ## 5. Combine information into one dataframe. Bind to other articles.
  plc_citation_df =
    data.frame(
      article_title = article_title,
      DOI = doi_txt,
      plumx_plc_url = plc_lnk,
      citation_title = citation_title,
      citation_date = citation_date,
      publisher = publisher,
      publisher_url = publisher_url,
      authors = authors,
      citation_url = citation_url,
      timestamp = Sys.time()
    )

  writeLines(paste("There are",nrow(plc_citation_df),"policy citations for",article_title))

  output_df <- bind_rows(output_df,plc_citation_df)

}
```

```
  return(output_df)
}
```

Execute Functions

```
# SSRN links
ssrn_plumx_out = ssrn_plumx_counts(zotero_csv = zotero_csv)
```

```
## Starting scrape for 3 SSRN links
## Begin scraping data for https://papers.ssrn.com/abstract=3566298
##
## Successfully opened SSRN link: https://papers.ssrn.com/abstract=3566298
## Waiting 5 seconds for the page to completely load...
## Successfully opened PlumX page
## Waiting 5 seconds for the page to completely load...
## Successfully switched windows
## Successfully extracted HTML of SSRN PlumX page
##
## Article title: World Health Organization Declared a Pandemic Public Health Menace: A Systematic Revi
## Article DOI: 10.2139/ssrn.3566298
## Article URL: https://plu.mx/ssrn/a/?ssrn_id=3566298
## CSS selectors: li.row.metric-details-item.metric-details-citation
## CSS selectors: li.row.metric-details-item.metric-details-usage
## CSS selectors: li.row.metric-details-item.metric-details-capture
## CSS selectors: li.row.metric-details-item.metric-details-social_media
## Successfully gathered counts for li.row.metric-details-item.metric-details-citation
## Successfully gathered counts for li.row.metric-details-item.metric-details-usage
## Successfully gathered counts for li.row.metric-details-item.metric-details-capture
## Successfully gathered counts for li.row.metric-details-item.metric-details-social_media
## Closed PlumX tab. Returned to first tab.
## Number of entries in final output: 1
## Begin scraping data for https://papers.ssrn.com/abstract=3815670
##
## Successfully opened SSRN link: https://papers.ssrn.com/abstract=3815670
## Waiting 5 seconds for the page to completely load...
## Successfully opened PlumX page
## Waiting 5 seconds for the page to completely load...
## Successfully switched windows
## Successfully extracted HTML of SSRN PlumX page
##
## Article title: Casting Light on an Underserved Population: Evidence Review of HIV Among Migrants in
## Article DOI: 10.2139/ssrn.3815670
## Article URL: https://plu.mx/ssrn/a/?ssrn_id=3815670
## CSS selectors: li.row.metric-details-item.metric-details-citation
## CSS selectors: li.row.metric-details-item.metric-details-usage
## Successfully gathered counts for li.row.metric-details-item.metric-details-citation
## Successfully gathered counts for li.row.metric-details-item.metric-details-usage
## Closed PlumX tab. Returned to first tab.
## Number of entries in final output: 2
## Begin scraping data for https://papers.ssrn.com/abstract=4014499
##
```

```
## Successfully opened SSRN link: https://papers.ssrn.com/abstract=4014499
## Waiting 5 seconds for the page to completely load...
## Successfully opened PlumX page
## Waiting 5 seconds for the page to completely load...
## Successfully switched windows
## Successfully extracted HTML of SSRN PlumX page
##
## Article title: Persistent SARS-CoV-2 Infection with Accumulation of Mutations in a Patient with Poorl
## Article DOI: 10.2139/ssrn.4014499
## Article URL: https://plu.mx/ssrn/a/?ssrn_id=4014499
## CSS selectors: li.row.metric-details-item.metric-details-citation
## CSS selectors: li.row.metric-details-item.metric-details-usage
## CSS selectors: li.row.metric-details-item.metric-details-capture
## CSS selectors: li.row.metric-details-item.metric-details-mention
## CSS selectors: li.row.metric-details-item.metric-details-social_media
## Successfully gathered counts for li.row.metric-details-item.metric-details-citation
## Successfully gathered counts for li.row.metric-details-item.metric-details-usage
## Successfully gathered counts for li.row.metric-details-item.metric-details-capture
## Successfully gathered counts for li.row.metric-details-item.metric-details-mention
## Successfully gathered counts for li.row.metric-details-item.metric-details-social_media
## Closed PlumX tab. Returned to first tab.
## Number of entries in final output: 3
```

```
ssrn_plcy_cit = count_plcy_cit(ssrn_plumx_out)
```

```
## There are 3 articles with Policy Citations
##
## Article title: World Health Organization Declared a Pandemic Public Health Menace: A Systematic Revie
## Article DOI: 10.2139/ssrn.3566298
## Policy Citation page link: https://plu.mx/ssrn/a/policy_citation?ssrn_id=3566298
## There are 1 policy citations for World Health Organization Declared a Pandemic Public Health Menace:
##
## Article title: Casting Light on an Underserved Population: Evidence Review of HIV Among Migrants in t
## Article DOI: 10.2139/ssrn.3815670
## Policy Citation page link: https://plu.mx/ssrn/a/policy_citation?ssrn_id=3815670
## There are 1 policy citations for Casting Light on an Underserved Population: Evidence Review of HIV A
##
## Article title: Persistent SARS-CoV-2 Infection with Accumulation of Mutations in a Patient with Poorl
## Article DOI: 10.2139/ssrn.4014499
## Policy Citation page link: https://plu.mx/ssrn/a/policy_citation?ssrn_id=4014499
## There are 2 policy citations for Persistent SARS-CoV-2 Infection with Accumulation of Mutations in a
```

```
# Science Direct links
scidir_plumx_out = scidir_plumx_counts(zotero_csv)
```

```
## Beginning scrape for 3 Science Direct links
## Begin scraping data for  https://www.sciencedirect.com/science/article/pii/S1098301516317764
##
## Successfully opened Science Direct link:  https://www.sciencedirect.com/science/article/pii/S10983015
## Waiting 5 seconds for the page to completely load...
## Successfully opened PlumX page
## Waiting 5 seconds for the page to completely load...
## Successfully switched windows
```

```
## Successfully extracted HTML of Science Direct PlumX page
##
## Article title: Systematic Review Of Studies Estimating The Cost-Effectiveness Of Hiv Pre-Exposure Pr
## Article DOI: 10.1016/j.jval.2016.09.409
## Article URL: https://plu.mx/plum/a/?doi=10.1016/j.jval.2016.09.409&theme=plum-sciencedirect-theme&hi
## CSS selectors: li.row.metric-details-item.metric-details-citation
## CSS selectors: li.row.metric-details-item.metric-details-capture
## Successfully gathered counts for li.row.metric-details-item.metric-details-citation
## Successfully gathered counts for li.row.metric-details-item.metric-details-capture
## Closed PlumX tab. Returned to first tab.
## Number of entries in final output: 1
## Begin scraping data for  https://www.sciencedirect.com/science/article/pii/S2352301822000066
##
## Successfully opened Science Direct link:  https://www.sciencedirect.com/science/article/pii/S2352301
## Waiting 5 seconds for the page to completely load...
## Successfully opened PlumX page
## Waiting 5 seconds for the page to completely load...
## Successfully switched windows
## Successfully extracted HTML of Science Direct PlumX page
##
## Article title: Scaling up access to HIV pre-exposure prophylaxis (PrEP): should nurses do the job?
## Article DOI: 10.1016/s2352-3018(22)00006-6
## Article URL: https://plu.mx/plum/a/?doi=10.1016/S2352-3018(22)00006-6&theme=plum-sciencedirect-theme
## CSS selectors: li.row.metric-details-item.metric-details-citation
## CSS selectors: li.row.metric-details-item.metric-details-capture
## CSS selectors: li.row.metric-details-item.metric-details-social_media
## Successfully gathered counts for li.row.metric-details-item.metric-details-citation
## Successfully gathered counts for li.row.metric-details-item.metric-details-capture
## Successfully gathered counts for li.row.metric-details-item.metric-details-social_media
## Closed PlumX tab. Returned to first tab.
## Number of entries in final output: 2
## Begin scraping data for  https://www.sciencedirect.com/science/article/pii/S1413867020301641
##
## Successfully opened Science Direct link:  https://www.sciencedirect.com/science/article/pii/S141386
## Waiting 5 seconds for the page to completely load...
## Successfully opened PlumX page
## Waiting 5 seconds for the page to completely load...
## Successfully switched windows
## Successfully extracted HTML of Science Direct PlumX page
##
## Article title: High acceptability of PrEP teleconsultation and HIV self-testing among PrEP users dur
## Article DOI: 10.1016/j.bjid.2020.11.002
## Article URL: https://plu.mx/plum/a/?doi=10.1016/j.bjid.2020.11.002&theme=plum-sciencedirect-theme&hi
## CSS selectors: li.row.metric-details-item.metric-details-citation
## CSS selectors: li.row.metric-details-item.metric-details-capture
## Successfully gathered counts for li.row.metric-details-item.metric-details-citation
## Successfully gathered counts for li.row.metric-details-item.metric-details-capture
## Closed PlumX tab. Returned to first tab.
## Number of entries in final output: 3
```

```
scidir_plcy_cit = count_plcy_cit(scidir_plumx_out)
```

```
## There are 3 articles with Policy Citations
##
```

```
## Article title: Systematic Review Of Studies Estimating The Cost-Effectiveness Of Hiv Pre-Exposure Pr
## Article DOI: 10.1016/j.jval.2016.09.409
## Policy Citation page link: https://plu.mx/plum/a/policy_citation?doi=10.1016/j.jval.2016.09.409&theme
## There are 1 policy citations for Systematic Review Of Studies Estimating The Cost-Effectiveness Of Hi
##
## Article title: Scaling up access to HIV pre-exposure prophylaxis (PrEP): should nurses do the job?
## Article DOI: 10.1016/s2352-3018(22)00006-6
## Policy Citation page link: https://plu.mx/plum/a/policy_citation?doi=10.1016/S2352-3018(22)00006-6&tl
## There are 1 policy citations for Scaling up access to HIV pre-exposure prophylaxis (PrEP): should nu
##
## Article title: High acceptability of PrEP teleconsultation and HIV self-testing among PrEP users duri
## Article DOI: 10.1016/j.bjid.2020.11.002
## Policy Citation page link: https://plu.mx/plum/a/policy_citation?doi=10.1016/j.bjid.2020.11.002&theme
## There are 1 policy citations for High acceptability of PrEP teleconsultation and HIV self-testing am
```

```
# Save as CSVs
write.csv(ssrn_plumx_out, "SSRN_PlumX_counts.csv")
write.csv(ssrn_plcy_cit, "SSRN_PlumX_plcy_citations.csv")
write.csv(scidir_plumx_out, "SciDir_PlumX_counts.csv")
write.csv(scidir_plcy_cit, "SciDir_PlumX_plcy_citations.csv")
```

End Selenium Session

```
driver$server$stop()
```

```
## [1] TRUE
```