Web Scraping in R

You can download this .qmd file from here. Just hit the Download Raw File button.

Credit to Brianna Heggeseth and Leslie Myint from Macalester College for a few of these descriptions and examples.

Using rvest for web scraping

Please see `08_table_scraping.qmd` for a preview of web scraping techniques when no API exists, along with ethical considerations when scraping data. In this file, we will turn to scenarios when the webpage contains data of interest, but it is not already in table form.

Recall the four steps to scraping data with functions in the `rvest` library:

0. `robotstxt::paths_allowed()` Check if the website allows scraping, and then make sure we scrape "politely"
1. `read_html()`. Input the URL containing the data and turn the html code into an XML file (another markup format that's easier to work with).
2. `html_nodes()`. Extract specific nodes from the XML file by using the CSS path that leads to the content of interest. (use css="table" for tables.)
3. `html_text()`. Extract content of interest from nodes. Might also use `html_table()` etc.

More scraping ethics

`robots.txt`

`robots.txt` is a file that some websites will publish to clarify what can and cannot be scraped and other constraints about scraping. When a website publishes this file, this we need to comply with the information in it for moral and legal reasons.

We will look through the information in this tutorial and apply this to the NIH robots.txt file.

From our investigation of the NIH `robots.txt`, we learn:

- `User-agent: *`: Anyone is allowed to scrape
- `Crawl-delay: 2`: Need to wait 2 seconds between each page scraped
- No `Visit-time` entry: no restrictions on time of day that scraping is allowed
- No `Request-rate` entry: no restrictions on simultaneous requests
- No mention of `?page=`, `news-events`, `news-releases`, or `https://science.education.nih.gov/` in the `Disallow` sections. (This is what we want to scrape today.)

robotstxt package

We can also use functions from the `robotstxt` package, which was built to download and parse robots.txt files (more info). Specifically, the `paths_allowed()` function can check if a bot has permission to access certain pages.

A timeout to preview some technical ideas

HTML structure

HTML (hypertext markup language) is the formatting language used to create webpages. We can see the core parts of HTML from the rvest vignette.

Finding CSS Selectors

In order to gather information from a webpage, we must learn the language used to identify patterns of specific information. For example, on the NIH News Releases page, we can see that the data is represented in a consistent pattern of image + title + abstract.

We will identify data in a web page using a pattern matching language called CSS Selectors that can refer to specific patterns in HTML, the language used to write web pages.

For example:

- Selecting by tag:
  - `"a"` selects all hyperlinks in a webpage ("a" represents "anchor" links in HTML)
  - `"p"` selects all paragraph elements
- Selecting by ID and class:
  - `".description"` selects all elements with `class` equal to "description"

        - The `.` at the beginning is what signifies `class` selection.
        - This is one of the most common CSS selectors for scraping because in HTML, the `class` attribute is extremely commonly used to format webpage elements. (Any number of HTML elements can have the same `class`, which is not true for the `id` attribute.)
    - `"#mainTitle"` selects the SINGLE element with **id** equal to "mainTitle"
        - The `#` at the beginning is what signifies `id` selection.

```
<p class="title">Title of resource 1</p>
<p class="description">Description of resource 1</p>

<p class="title">Title of resource 2</p>
<p class="description">Description of resource 2</p>
```

Warning: Websites change often! So if you are going to scrape a lot of data, it is probably worthwhile to save and date a copy of the website. Otherwise, you may return after some time and your scraping code will include all of the wrong CSS selectors.

SelectorGadget

Although you can learn how to use CSS Selectors by hand, we will use a shortcut by installing the Selector Gadget tool.

- There is a version available for Chrome–add it to Chrome via the Chome Web Store.
    - Make sure to pin the extension to the menu bar. (Click the 3 dots > Extensions > Manage extensions. Click the "Details" button under SelectorGadget and toggle the "Pin to toolbar" option.)
- There is also a version that can be saved as a bookmark in the browser–see here.

You might watch the Selector Gadget tutorial video.

Case Study: NIH News Releases

Our goal is to build a data frame with the article title, publication date, and abstract text for the 50 most recent NIH news releases.

Head over to the NIH News Releases page. Click the Selector Gadget extension icon or bookmark button. As you mouse over the webpage, different parts will be highlighted in orange. Click on the title (but not the live link portion!) of the first news release. You'll notice that the Selector Gadget information in the lower right describes what you clicked on.

(If SelectorGadget ever highlights too much in green, you can click on portions that you do not want to turn them red.)

Scroll through the page to verify that only the information you intend (the description paragraph) is selected. The selector panel shows the CSS selector (`.teaser-title`) and the number of matches for that CSS selector (10). (You may have to be careful with your clicking–there are two overlapping boxes, and clicking on the link of the title can lead to the CSS selector of "a".)

[**Pause to Ponder:**] Repeat the process above to find the correct selectors for the following fields. Make sure that each matches 10 results:

- The publication date

  .date-display-single

- The article abstract paragraph (which will also include the publication date)

  .teaser-description

Retrieving Data Using `rvest` and CSS Selectors

Now that we have identified CSS selectors for the information we need, let's fetch the data using the `rvest` package similarly to our approach in `08_table_scraping.qmd`.

```
# check that scraping is allowed (Step 0)
robotstxt::paths_allowed("https://www.nih.gov/news-events/news-releases")
```

```
 www.nih.gov
```

```
[1] TRUE
```

```
# Step 1: Download the HTML and turn it into an XML file with read_html()
nih <- read_html("https://www.nih.gov/news-events/news-releases")
```

Finding the exact node (e.g. ".teaser-title") is the tricky part. Among all the html code used to produce a webpage, where do you go to grab the content of interest? This is where SelectorGadget comes to the rescue!

```
# Step 2: Extract specific nodes with html_nodes()
title_temp <- html_nodes(nih, ".teaser-title")
title_temp
```

```
{xml_nodeset (10)}
 [1] <h4 class="teaser-title"><a href="/news-events/news-releases/nih-researc ...
 [2] <h4 class="teaser-title"><a href="/news-events/news-releases/study-illum ...
 [3] <h4 class="teaser-title"><a href="/news-events/news-releases/nih-funded- ...
 [4] <h4 class="teaser-title"><a href="/news-events/news-releases/surgery-kid ...
 [5] <h4 class="teaser-title"><a href="/news-events/news-releases/nih-sponsor ...
 [6] <h4 class="teaser-title"><a href="/news-events/news-releases/topical-ste ...
 [7] <h4 class="teaser-title"><a href="/news-events/news-releases/tecovirimat ...
 [8] <h4 class="teaser-title"><a href="/news-events/news-releases/nih-central ...
 [9] <h4 class="teaser-title"><a href="/news-events/news-releases/nih-funded- ...
[10] <h4 class="teaser-title"><a href="/news-events/news-releases/longer-brea ...
```

```
# Step 3: Extract content from nodes with html_text(), html_name(),
#    html_attrs(), html_children(), html_table(), etc.
# Usually will still need to do some stringr adjustments
title_vec <- html_text(title_temp)
title_vec
```

```
 [1] "NIH researchers develop eye drops that slow vision loss in animals"
 [2] "Study illuminates the structural features of memory formation at cellular and subcellul
 [3] "NIH-funded study identifies potential new stroke treatment"
 [4] "Surgery in kids with mild sleep-disordered breathing tied to fewer doctor visits, meds
 [5] "NIH-sponsored trial of Lassa vaccine opens"
 [6] "Topical steroid withdrawal diagnostic criteria defined by NIH researchers"
 [7] "Tecovirimat is safe but ineffective as treatment for clade II mpox"
 [8] "NIH centralizes peer review to improve efficiency and strengthen integrity "
 [9] "NIH-funded research team engineers new drug targeting pain sensation pathway"
[10] "Longer breastfeeding linked to blood-pressure lowering effects of certain infant gut ba
```

You can also write this altogether with a pipe:

```
robotstxt::paths_allowed("https://www.nih.gov/news-events/news-releases")
```

```
 www.nih.gov
```

```
[1] TRUE
```

```
read_html("https://www.nih.gov/news-events/news-releases") |>
  html_nodes(".teaser-title") |>
  html_text()
```

```
 [1] "NIH researchers develop eye drops that slow vision loss in animals"
 [2] "Study illuminates the structural features of memory formation at cellular and subcellul
 [3] "NIH-funded study identifies potential new stroke treatment"
 [4] "Surgery in kids with mild sleep-disordered breathing tied to fewer doctor visits, meds'
 [5] "NIH-sponsored trial of Lassa vaccine opens"
 [6] "Topical steroid withdrawal diagnostic criteria defined by NIH researchers"
 [7] "Tecovirimat is safe but ineffective as treatment for clade II mpox"
 [8] "NIH centralizes peer review to improve efficiency and strengthen integrity "
 [9] "NIH-funded research team engineers new drug targeting pain sensation pathway"
[10] "Longer breastfeeding linked to blood-pressure lowering effects of certain infant gut ba
```

And finally we wrap the 4 steps above into the `bow` and `scrape` functions from the `polite` package:

```r
session <- bow("https://www.nih.gov/news-events/news-releases", force = TRUE)

nih_title <- scrape(session) |>
  html_nodes(".teaser-title") |>
  html_text()
nih_title
```

```
 [1] "NIH researchers develop eye drops that slow vision loss in animals"
 [2] "Study illuminates the structural features of memory formation at cellular and subcellul
 [3] "NIH-funded study identifies potential new stroke treatment"
 [4] "Surgery in kids with mild sleep-disordered breathing tied to fewer doctor visits, meds'
 [5] "NIH-sponsored trial of Lassa vaccine opens"
 [6] "Topical steroid withdrawal diagnostic criteria defined by NIH researchers"
 [7] "Tecovirimat is safe but ineffective as treatment for clade II mpox"
 [8] "NIH centralizes peer review to improve efficiency and strengthen integrity "
 [9] "NIH-funded research team engineers new drug targeting pain sensation pathway"
[10] "Longer breastfeeding linked to blood-pressure lowering effects of certain infant gut ba
```

Putting multiple columns of data together.

Now repeat the process above to extract the publication date and the abstract.

```r
nih_pubdate <- scrape(session) |>
  html_nodes(".date-display-single") |>
  html_text()
nih_pubdate
```

```
 [1] "March 21, 2025" "March 20, 2025" "March 17, 2025" "March 17, 2025"
 [5] "March 17, 2025" "March 14, 2025" "March 12, 2025" "March 6, 2025"
 [9] "March 5, 2025"  "March 4, 2025"
```

```r
nih_description <- scrape(session) |>
  html_nodes(".teaser-description") |>
  html_text()
nih_description
```

```
 [1] "March 21, 2025 -     \n          Treatment shows potential to slow the progression of 
 [2] "March 20, 2025 -     \n          NIH-funded study uses cutting-edge imaging techniques
 [3] "March 17, 2025 -     \n          Preclinical study in rodents suggests that uric acid 
 [4] "March 17, 2025 -     \n          NIH-funded study supports use of adenotonsillectomy i
 [5] "March 17, 2025 -     \n          Lassa fever is a viral hemorrhagic disease that can b
 [6] "March 14, 2025 -     \n          Criteria may help guide treatment of dermatitis. "
 [7] "March 12, 2025 -     \n          NIH-sponsored trial data offer further evidence to he
 [8] "March 6, 2025 -     \n          The proposed approach is expected to save more than $6
 [9] "March 5, 2025 -     \n          Study of CB1 receptor has implications for chronic pai
[10] "March 4, 2025 -     \n          Nursing for at least six months may spur beneficial gu
```

Combine these extracted variables into a single tibble. Make sure the variables are formatted correctly - e.g. `pubdate` has `date` type, `description` does not contain the `pubdate`, etc.

```r
# use tibble() to put multiple columns together into a tibble
nih_top10 <- tibble(title = nih_title,
                    pubdate = nih_pubdate,
                    description = nih_description)
nih_top10
```

```
# A tibble: 10 x 3
   title                                                     pubdate description
   <chr>                                                     <chr>   <chr>      
 1 "NIH researchers develop eye drops that slow vision loss~ March ~ "March 21,~
 2 "Study illuminates the structural features of memory for~ March ~ "March 20,~
 3 "NIH-funded study identifies potential new stroke treatm~ March ~ "March 17,~
 4 "Surgery in kids with mild sleep-disordered breathing ti~ March ~ "March 17,~
 5 "NIH-sponsored trial of Lassa vaccine opens"              March ~ "March 17,~
 6 "Topical steroid withdrawal diagnostic criteria defined ~ March ~ "March 14,~
 7 "Tecovirimat is safe but ineffective as treatment for cl~ March ~ "March 12,~
 8 "NIH centralizes peer review to improve efficiency and s~ March ~ "March 6, ~
 9 "NIH-funded research team engineers new drug targeting p~ March ~ "March 5, ~
10 "Longer breastfeeding linked to blood-pressure lowering ~ March ~ "March 4, ~
```

```r
# now clean the data
nih_top10 <- nih_top10 |>
  mutate(pubdate = mdy(pubdate),
         description = str_trim(str_replace(description, ".*\\n", "")))
nih_top10
```

```
# A tibble: 10 x 3
   title                                                 pubdate    description
   <chr>                                                 <date>     <chr>
 1 "NIH researchers develop eye drops that slow vision l~ 2025-03-21 Treatment ~
 2 "Study illuminates the structural features of memory ~ 2025-03-20 NIH-funded~
 3 "NIH-funded study identifies potential new stroke tre~ 2025-03-17 Preclinica~
 4 "Surgery in kids with mild sleep-disordered breathing~ 2025-03-17 NIH-funded~
 5 "NIH-sponsored trial of Lassa vaccine opens"           2025-03-17 Lassa feve~
 6 "Topical steroid withdrawal diagnostic criteria defin~ 2025-03-14 Criteria m~
 7 "Tecovirimat is safe but ineffective as treatment for~ 2025-03-12 NIH-sponso~
 8 "NIH centralizes peer review to improve efficiency an~ 2025-03-06 The propos~
 9 "NIH-funded research team engineers new drug targetin~ 2025-03-05 Study of C~
10 "Longer breastfeeding linked to blood-pressure loweri~ 2025-03-04 Nursing fo~
```

NOW - continue this process to build a tibble with the most recent 50 NIH news releases, which will require that you iterate over 5 webpages! You should write at least one function, and you will need iteration–use both a `for` loop and appropriate `map_()` functions from `purrr`. Some additional hints:

- Mouse over the page buttons at the very bottom of the news home page to see what the URLs look like.
- Include `Sys.sleep(2)` in your function to respect the `Crawl-delay: 2` in the NIH `robots.txt` file.
- Recall that `bind_rows()` from `dplyr` takes a list of data frames and stacks them on top of each other.

[**Pause to Ponder:**] Create a function to scrape a single NIH press release page by filling missing pieces labeled `???`:

```r
# Helper function to reduce html_nodes() |> html_text() code duplication
get_text_from_page <- function(page, css_selector) {
  page |>
    html_nodes(css_selector) |>
    html_text()
}
```

```r
# Main function to scrape and tidy desired attributes
scrape_page <- function(url) {
    Sys.sleep(2)
    page <- read_html(url)
    article_titles <- get_text_from_page(page, ".teaser-title")
    article_dates <- get_text_from_page(page, ".date-display-single")
    article_dates <- mdy(article_dates)
    article_description <- get_text_from_page(page, ".teaser-description")
    article_description <- str_trim(str_replace(article_description,
                                                ".*\\n",
                                                "")
                                    )

    tibble(
      title = article_titles,
      dates = article_dates,
      description = article_description
    )
}

scrape_page("https://www.nih.gov/news-events/news-releases")
```

```
# A tibble: 10 x 3
   title                                                   dates      description
   <chr>                                                   <date>     <chr>
 1 "NIH researchers develop eye drops that slow vision l~ 2025-03-21 Treatment ~
 2 "Study illuminates the structural features of memory ~ 2025-03-20 NIH-funded~
 3 "NIH-funded study identifies potential new stroke tre~ 2025-03-17 Preclinica~
 4 "Surgery in kids with mild sleep-disordered breathing~ 2025-03-17 NIH-funded~
 5 "NIH-sponsored trial of Lassa vaccine opens"           2025-03-17 Lassa feve~
 6 "Topical steroid withdrawal diagnostic criteria defin~ 2025-03-14 Criteria m~
 7 "Tecovirimat is safe but ineffective as treatment for~ 2025-03-12 NIH-sponso~
 8 "NIH centralizes peer review to improve efficiency an~ 2025-03-06 The propos~
 9 "NIH-funded research team engineers new drug targetin~ 2025-03-05 Study of C~
10 "Longer breastfeeding linked to blood-pressure loweri~ 2025-03-04 Nursing fo~
```

[**Pause to Ponder:**] Use a for loop over the first 5 pages:

```r
pages <- vector("list", length = 6)
pos <- 0
```

```
for (i in 2025:2024) {
  for (j in 0:2) {
    pos <- pos + 1
    url <- str_c("https://www.nih.gov/news-events/news-releases?", i,
                 "&page=", j, "&1=")
    pages[[pos]] <- scrape_page(url)
  }
}

df_articles <- bind_rows(pages)
head(df_articles)
```

```
# A tibble: 6 x 3
  title                                                     dates      description
  <chr>                                                     <date>     <chr>
1 NIH researchers develop eye drops that slow vision los~ 2025-03-21 Treatment ~
2 Study illuminates the structural features of memory fo~ 2025-03-20 NIH-funded~
3 NIH-funded study identifies potential new stroke treat~ 2025-03-17 Preclinica~
4 Surgery in kids with mild sleep-disordered breathing t~ 2025-03-17 NIH-funded~
5 NIH-sponsored trial of Lassa vaccine opens              2025-03-17 Lassa feve~
6 Topical steroid withdrawal diagnostic criteria defined~ 2025-03-14 Criteria m~
```

[**Pause to Ponder:**] Use map functions in the purrr package:

```
library(purrr)

base_url <- "https://www.nih.gov/news-events/news-releases?page="
urls_all_pages <- str_c(base_url, seq(0,5))

pages2 <- purrr::map(urls_all_pages, scrape_page)
df_articles2 <- bind_rows(pages2)
head(df_articles2)
```

```
# A tibble: 6 x 3
  title                                                     dates      description
  <chr>                                                     <date>     <chr>
1 NIH researchers develop eye drops that slow vision los~ 2025-03-21 Treatment ~
2 Study illuminates the structural features of memory fo~ 2025-03-20 NIH-funded~
3 NIH-funded study identifies potential new stroke treat~ 2025-03-17 Preclinica~
4 Surgery in kids with mild sleep-disordered breathing t~ 2025-03-17 NIH-funded~
5 NIH-sponsored trial of Lassa vaccine opens              2025-03-17 Lassa feve~
6 Topical steroid withdrawal diagnostic criteria defined~ 2025-03-14 Criteria m~
```

On Your Own

1. Go to https://www.bestplaces.net and search for Minneapolis, Minnesota. This is a site some people use when comparing cities they might consider working in and/or moving to. Using SelectorGadget, extract the following pieces of information from the Minneapolis page:

- property crime (on a scale from 0 to 100)
- minimum income required for a single person to live comfortably
- average monthly rent for a 2-bedroom apartment
- the "about" paragraph (the very first paragraph above "Location Details")

2. Write a function called `scrape_bestplaces()` with arguments for `state` and `city`. When you run, for example, `scrape_bestplaces("minnesota", "minneapolis")`, the output should be a 1 x 6 tibble with columns for `state`, `city`, `crime`, `min_income_single`, `rent_2br`, and `about`.

3. Create a 5 x 6 tibble by running `scrape_bestplaces()` 5 times with 5 cities you are interested in. You might have to combine tibbles using `bind_rows()`. Be sure you look at the URL at bestplaces.net for the various cities to make sure it works as you expect. For bonus points, create the same 5 x 6 tibble for the same 5 cities using `purrr:map2`!