More Scraping Examples

Max Turgeon

SCI 2000–Introduction to Data Science

Example i

```
library(rvest)
library(tidyverse)

url <- "https://en.wikipedia.org/wiki/World_population"
world_pop_tables <- read_html(url) %>%
  html_nodes("table.wikitable")

length(world_pop_tables)
```

```
## [1] 13
```

Example ii

```
# Looking at each of them one at a time
# E.g. world_pop_tables[[1]], world_pop_tables[[2]]
# We recognize it's the 3rd table
library(knitr)

world_pop_tables[[3]] %>%
  html_table() %>%
  select(Rank, Country, Population) %>%
  kable()
```

Example iii

Rank	Country	Population
1	China	1,407,257,360
2	India	1,375,030,248
3	United States	331,402,425
4	Indonesia	269,603,400
5	Pakistan	220,892,331
6	Brazil	212,925,393
7	Nigeria	206,139,587
8	Bangladesh	170,397,280
9	Russia	146,748,590
10	Mexico	127,792,286

Example iv

```
# Alternatively, if you don't like double brackets
library(purrr)

world_pop_tables %>%
  pluck(3) %>%
  html_table() %>%
  select(Rank, Country, Population) %>%
  kable()
```

Example v

Rank	Country	Population
1	China	1,407,257,360
2	India	1,375,030,248
3	United States	331,402,425
4	Indonesia	269,603,400
5	Pakistan	220,892,331
6	Brazil	212,925,393
7	Nigeria	206,139,587
8	Bangladesh	170,397,280
9	Russia	146,748,590
10	Mexico	127,792,286

CSS Selectors and attributes

Syntax	Explanation
`div[target]`	Selects `div` elements with a `target` attribute
`div[target=foo]`	Selects `div` elements with `target="foo"`
`a[href^="https"]`	Selects `a` elements whose `href` attribute begins with `"https"`
`a[href$=".pdf"]`	Selects `a` elements whose `href` attribute ends with `".pdf"`
`a[href*="flower"]`	Selects every `a` elements whose href attribute contains `"flower"`

Exercise

On IMDb.com, search for any movie of your choice. Use web scraping to extract the average rating. What CSS selector did you use?

Solution

```
library(rvest)
# url for "I Care A Lot" (2020)
url <- "https://www.imdb.com/title/tt9893250/"

read_html(url) %>%
    html_elements("span[itemprop=ratingValue]") %>%
    html_text()
```

```
## [1] "6.2"
```

Example—PubMed i

- We search PubMed for articles on the adverse effects of hemodialysis at home.
    - The query is `"Hemodialysis, Home/Adverse Effects"[Mesh]`.
- This leads to the following URL: `https://pubmed.ncbi.nlm.nih.gov/?term=%22Hemodialysis%2C+Home%2FAdverse+Effects%22%5BMesh%5D`
- On this page, the article links are all the form `https://pubmed.ncbi.nlm.nih.gov/30041224/`
    - But they are written using a relative path!

Example—PubMed ii

```
url <- paste0("https://pubmed.ncbi.nlm.nih.gov/",
              "?term=%22Hemodialysis%2C+Home%2F",
              "Adverse+Effects%22%5BMesh%5D")

results <- read_html(url) %>%
  html_nodes("a") %>%
  html_attr("href") # Extract the attribute

length(results)
```

```
## [1] 87
```

Example—PubMed iii

```
library(stringr)
# This is empty...
str_subset(results,
           "https://pubmed.ncbi.nlm.nih.gov/\\d+/")
```

```
## character(0)
```

```
# But this gives us what we want!
urls <- str_subset(results, "/\\d+/")
urls
```

Example—PubMed iv

```
## [1] "/30041224/" "/21775973/" "/25441436/"
"/29778559/" "/25925825/"
## [6] "/29661768/" "/29656667/" "/18638237/"
"/30659057/" "/25920990/"
```

- If we click on one of these links, we can see an abstract.
- This abstract is inside a `div` of class `abstract-content`.
- If we loop through the URLs, read in the HTML file, and extract the abstract, we can do a downstream analysis with them (e.g. sentiment analysis).

Example—PubMed v

```
# Add domain to relative URLs
full_urls <- paste0("https://pubmed.ncbi.nlm.nih.gov",
                    urls)
full_urls
```

```
## [1]
"https://pubmed.ncbi.nlm.nih.gov/30041224/"
## [2]
"https://pubmed.ncbi.nlm.nih.gov/21775973/"
## [3]
"https://pubmed.ncbi.nlm.nih.gov/25441436/"
```

Example—PubMed vi

```
## [4]
"https://pubmed.ncbi.nlm.nih.gov/29778559/"
## [5]
"https://pubmed.ncbi.nlm.nih.gov/25925825/"
## [6]
"https://pubmed.ncbi.nlm.nih.gov/29661768/"
## [7]
"https://pubmed.ncbi.nlm.nih.gov/29656667/"
## [8]
"https://pubmed.ncbi.nlm.nih.gov/18638237/"
## [9]
"https://pubmed.ncbi.nlm.nih.gov/30659057/"
```

Example—PubMed vii

```
## [10]
"https://pubmed.ncbi.nlm.nih.gov/25920990/"

# Let's see how it could work
# with the first url
read_html(full_urls[1]) %>%
  html_nodes("div.abstract-content") %>%
  html_text()
```

Example—PubMed viii

[1] ”\n \n \n\n\n \n \n \n Home hemodialysis (HHD) has been available as a modality of renal replacement therapy since the 1960s. HHD allows intensive dialysis such as nocturnal hemodialysis or short daily hemodialysis. Previous studies have shown that patients receiving HHD have an increased survival and better quality of life compared with those receiving in-center conventional HD. However, HHD may increase the risk for specific complications such as vascular access complications, infection, loss of residual kidney function and patient and caregiver burden.

Example—PubMed ix

In Japan, only 529 patients (0.2% of the total dialysis patients) were on maintenance HHD at the end of 2014. The most commonly perceived barriers to intensive HHD included lack of patient motivation, unwillingness to change from in-center modality, and fear of self-cannulation. However, these barriers can often be overcome by adequate predialysis education, motivational training of patient and caregiver, nurse-assisted cannulation, nurse-led home visits, a well-defined nursing/technical support system for patients, and provision of respite care.\n \n

Example—PubMed x

```
\n\n \n\n\n \n ”

# Let's create a function!
# Input: url
extract_abstract <- function(url) {
  read_html(url) %>%
    html_nodes(”div.abstract-content”) %>%
    html_text() %>%
    str_replace_all(”^\\s+|\\s+$”, ””)
}
```

Example—PubMed xi

```
extract_abstract(full_urls[1])
```

```
## [1] "Home hemodialysis (HHD) has been
available as a modality of renal replacement
therapy since the 1960s. HHD allows intensive
dialysis such as nocturnal hemodialysis or short
daily hemodialysis. Previous studies have shown
that patients receiving HHD have an increased
survival and better quality of life compared with
those receiving in-center conventional HD.
However, HHD may increase the risk for specific
```

Example—PubMed xii

complications such as vascular access complications, infection, loss of residual kidney function and patient and caregiver burden. In Japan, only 529 patients (0.2% of the total dialysis patients) were on maintenance HHD at the end of 2014. The most commonly perceived barriers to intensive HHD included lack of patient motivation, unwillingness to change from in-center modality, and fear of self-cannulation. However, these barriers can often be overcome by adequate predialysis education, motivational training of patient and caregiver, nurse-assisted

Example—PubMed xiii

```
cannulation, nurse-led home visits, a
well-defined nursing/technical support system for
patients, and provision of respite care."

# Different ways to loop
# We will use map from package purrr
library(purrr)

abstracts <- map(full_urls,
                 extract_abstract)

str(abstracts)
```

Example—PubMed xiv

```
## List of 10
## $ : chr "Home hemodialysis (HHD) has been
available as a modality of renal replacement
therapy since the 1960s. HHD allo"| __truncated__
## $ : chr "Prior small studies have shown
multiple benefits of frequent nocturnal
hemodialysis compared to conventional th"|
__truncated__
## $ : chr "Background:\n \n \n There is a
growing interest in home hemodialysis because of
its clinical b"| __truncated__
## $ : chr(0)
```

Example—PubMed xv

```
## $ : chr "Interest in home hemodialysis (HD) is
high because of the reported benefits and its
excellent safety record. Ho"| __truncated__
## $ : chr(0)
## $ : chr "Introduction:\n \n \n Only a minority
of patients with chronic kidney disease treated
by hemod"| __truncated__
## $ : chr "Home hemodialysis is regaining
popularity as a treatment choice for end-stage
kidney disease. This trend is fue"| __truncated__
## $ : chr "Background and objectives:\n \n \n
Canadian home hemodialysis guidelines highlight
```

Example—PubMed xvi

```
the potenti"| __truncated__
## $ : chr "Multiple observational studies along
with a limited number of randomized clinical
trials suggest that intensive"| __truncated__
```

Further examples of CSS selectors

- `p a` — finds all `a` tags inside of a `p` tag.
- `body p a` — finds all `a` tags inside of a `p` tag inside of a `body` tag.
- `p.outer-text` — finds all `p` tags with a *class* of `outer-text`.
- `p#first` — finds all `p` tags with an *id* of `first`.
- `body p.outer-text` — finds any `p` tags with a class of `outer-text` inside of a `body` tag.

Example—Weather forecast i

- We can find weather forecasts for Winnipeg at this URL: `https://weather.gc.ca/city/pages/mb-36_metric_e.html`
- The forecast is split into 2 parts:
  - Day: `div.row strong`
  - Temperature: `div.row span.wxo-metric-hide`
- We will extract each part separately and combine in a `data.frame`.

Example—Weather forecast ii

```
library(rvest)

url <- "https://weather.gc.ca/city/pages/mb-36_metric_e
forecast <- read_html(url)

days <- forecast %>%
  html_elements("div.row strong") %>%
  html_text()
days
```

Example—Weather forecast iii

```
## [1] "Tonight" "Tue, 30 Mar" " Night: "
"Wed, 31 Mar" " Night: "
## [6] "Thu, 1 Apr" " Night: " "Fri, 2 Apr" "
Night: " "Sat, 3 Apr"
## [11] " Night: " "Sun, 4 Apr"
```

Example—Weather forecast iv

```
# Not quite...
# Let's use regex to remove "Night:"
library(stringr)
days <- forecast %>%
  html_elements("div.row strong") %>%
  html_text() %>%
  str_subset("Night:", negate = TRUE)
days
```

Example—Weather forecast v

```
## [1] "Tonight" "Tue, 30 Mar" "Wed, 31 Mar"
"Thu, 1 Apr" "Fri, 2 Apr"
## [6] "Sat, 3 Apr" "Sun, 4 Apr"
```

```
temps <- forecast %>%
    html_elements("div.row span.wxo-metric-hide") %>%
    html_text()
temps
```

Example—Weather forecast vi

```
## [1] "19°C" "19°C" "19°C" "19°C" "19°C"
## [6] "19°C" "-7°C" "-7°C" "-3°C" "9°C"
## [11] "17°C" "11°C" "13°C" "10.7°C" "-20.8°C"
## [16] "12.5 mm" "2.4 mm" "1.9 cm" "13.0°C"
"-20.0°C"
## [21] "10.7 mm" "0.0 mm" "0.0 cm" "9.7°C"
"-15.4°C"
## [26] "4.2 mm" "1.8 mm" "0.0 cm" "9.4°C"
"-17.4°C"
## [31] "36.6 mm" "36.6 mm" "0.0 cm" "13.2°C"
"-16.3°C"
## [36] "9.6 mm" "9.6 mm" "0.0 cm" "17.5°C"
```

Example—Weather forecast vii

```
"-15.0°C"
## [41] "29.9 mm" "17.4 mm" "12.5 cm" "14.6°C"
"-16.5°C"
## [46] "13.2 mm" "3.0 mm" "13.2 cm" "18.7°C"
"-15.6°C"
## [51] "16.1 mm" "7.0 mm" "9.1 cm" "0.0 cm"
"18.1°C"
## [56] "-14.3°C" "11.4 mm" "0.2 mm" "11.4 cm"
"0.0 cm"
## [61] "13.3°C" "-17.9°C" "7.2 mm" "1.2 mm" "0.0
cm"
## [66] "0.0 cm" "14.6°C" "-16.6°C" "12.0 mm"
```

Example—Weather forecast viii

```
"0.0 mm"
## [71] "5.4 cm" "0.0 cm" "5.8\n\t\t°C"
"-5.7\n\t\t°C" "0.0 mm"

# We also picked up precipitations!
# And other stuff...
# Better: we want span with title attribute
temps <- forecast %>%
    html_elements("div.row span.wxo-metric-hide[title]")
    html_text()
temps
```

Example—Weather forecast ix

```
## [1] "-7°C" "-7°C" "-3°C" "9°C" "17°C" "11°C"
"13°C"
```

```
# Put into nice table with caption
library(knitr)
data.frame(
  Day = days,
  Forecast = temps
) %>%
  kable(caption = "Weather Forecast for Winnipeg")
```

Example—Weather forecast x

Table 4: Weather Forecast for Winnipeg

Day	Forecast
Tonight	-7°C
Tue, 30 Mar	-7°C
Wed, 31 Mar	-3°C
Thu, 1 Apr	9°C
Fri, 2 Apr	17°C
Sat, 3 Apr	11°C
Sun, 4 Apr	13°C

Exercise

IMDd can provide a list of the top 50 movies by user ratings. Here's where you can find it: `https://www.imdb.com/search/title/?groups=top_250&sort=user_rating`

Using web scraping, extract the title of these movies.

Bonus: Try to also extract the year the movie came out.

Solution i

- Movie titles appear inside `a` elements, which are themselves inside `h3` elements of class `lister-item-header`.
- Movie years appear inside `span` elements of class `lister-item-year`, which are themselves inside `h3` elements of class `lister-item-header`.

Solution ii

```
library(rvest)
url <- paste0("https://www.imdb.com/search/title/",
              "?groups=top_250&sort=user_rating")

imdb_page <- read_html(url)

top50_titles <- imdb_page %>%
  html_elements("h3.lister-item-header a") %>%
  html_text()
head(top50_titles)
```

Solution iii

```
## [1] "The Shawshank Redemption"
## [2] "The Godfather"
## [3] "The Dark Knight"
## [4] "The Godfather: Part II"
## [5] "12 Angry Men"
## [6] "The Lord of the Rings: The Return of the
King"
```

Solution iv

```
csssel <- "h3.lister-item-header span.lister-item-year"
top50_years <- imdb_page %>%
  html_elements(csssel) %>%
  html_text()
head(top50_years)
```

```
## [1] "(1994)" "(1972)" "(2008)" "(1974)"
"(1957)" "(2003)"
```

Solution v

```
library(tidyverse)

# Combine into tibble
tibble(Movie = top50_titles,
       Year = top50_years) %>%
  mutate(Year = str_replace_all(Year,
                                "\\(|\\)",
                                ""),
         Year = as.numeric(Year))
```

Solution vi

```
## # A tibble: 50 x 2
## Movie Year
## <chr> <dbl>
## 1 The Shawshank Redemption 1994
## 2 The Godfather 1972
## 3 The Dark Knight 2008
## 4 The Godfather: Part II 1974
## 5 12 Angry Men 1957
## 6 The Lord of the Rings: The Return of the
King 2003
## 7 Pulp Fiction 1994
## 8 Schindler's List 1993
```

Solution vii

```
## 9 Inception 2010
## 10 Fight Club 1999
## # ... with 40 more rows
```

Ethics of web scraping i

- So far, we've only run quick toy examples.
- But keep in mind: you're making requests to a web server every time you tun `GET()` or `read_html()`.
- Websites may deny you access if they think you are excessive.
- Some websites list sections of the website you shouldn't scrape in a file called `robots.txt`.
    - E.g. Look up `https://weather.gc.ca/robots.txt`
- If you are parsing the same HTML file many times, it's better to save the output of `read_html()` first, and then parse the output multiple times.
    - See previous example.

Ethics of web scraping ii

- When making multiple requests to the same server (cf. PubMed example), consider adding a delay between requests.
    - E.g. using `Sys.sleep(time = 1)` inside your function.
- Have a look at the `polite` package, which helps implement some of these strategies.