Web scraping

WICSS-Tucson

Thomas Davidson

Rutgers University

January 6, 2020

Plan

1. What is web-scraping?
2. When should I use it?
3. How to scrape a web page
4. Crawling websites
5. `selenium` and browser automation

What is web-scraping?

Terminology

- Web-scraping is a method to collect data from websites
  - We use the code underlying a webpage to collect data (**scraping**)
  - The process is then repeated for other pages on the same website in an automated fashion (**crawling**)

What is web-scraping?

Challenges

- Different websites have different structures, so a script used to scrape one website will likely have to be changed to scrape another
- Websites can be internally inconsistent, making them difficult to scrape
- Some websites are easier to crawl than others
- Some websites limit or prohibit scraping

When should I use it?

Commercial use cases

- Search engines
  - Google scrapes websites to create a searchable index of the internet
- Price comparison
  - Kayak scrape airlines to compare flight prices, other websites do the same for hotels and rental cars
- Recruitment
  - Recruitment companies scrape LinkedIn to get data on workers

When should I use it?

Social scientific use cases

- Web-scraping is a useful tool to collect data from websites without APIs
  - Large social media platforms and other sites have APIs but smaller websites do not
    - Local newspapers, forums, small businesses, educational institutions, etc.
- Often we want to collect data from a single website
  - e.g. All posts written on a forum
- Sometimes we might want to collect data from many websites
  - e.g. All schools in a school district

When should I use it?

Ethical and legal considerations

No Robots, Spiders, or Scrapers: Legal and Ethical Regulation of Data Collection Methods in Social Media Terms of Service

Casey Fiesler,[1]* Nathan Beard,[2] Brian C. Keegan[1]
[1]Department of Information Science, University of Colorado Boulder
[2]College of Information Studies, University of Maryland

Abstract

Researchers from many different disciplines rely on social media data as a resource. Whereas some platforms explicitly allow data collection, even facilitating it through an API, others explicitly forbid automated or manual collection processes. A current topic of debate within the social computing research community involves the ethical (or even legal) implications of collecting data in ways that violate Terms of Service (TOS). Using a sample of TOS from over one hundred social media sites from around the world, we analyze TOS language and content in order to better understand the landscape of prohibitions on this practice. Our findings show that

opportunities for digital social research, with new ways of collecting, analyzing, and visualizing data; it also allows for ordered collection, so that messy online data can become usable, well-ordered data sets (Marres and Weltevrede 2013).

However, even when data collection is possible technically, sometimes it is prohibited by terms of service (TOS), which restrict certain behaviors and uses of a site. Whether it is permissible, or ethical, for researchers to violate TOS in the course of collecting data is currently an open question within the social computing research community (Vaccaro et al. 2015; Vitak, Shilton, and Ashktorab 2016).

When should I use it?

Ethical and legal considerations

- Fiesler, Beard, and Keegan (2020) review the legal cases related to web-scraping and analyze website terms of service
  - "In short, it is an unsettled question as to whether it is explicitly illegal (or even a criminal act) to violate TOS."
  - No academic or journalist has ever been prosecuted for violating a website terms of service to collect data for research
- They analyze terms of service of over 100 social media websites
  - Terms of service are ambiguous, inconsistent, and lack context

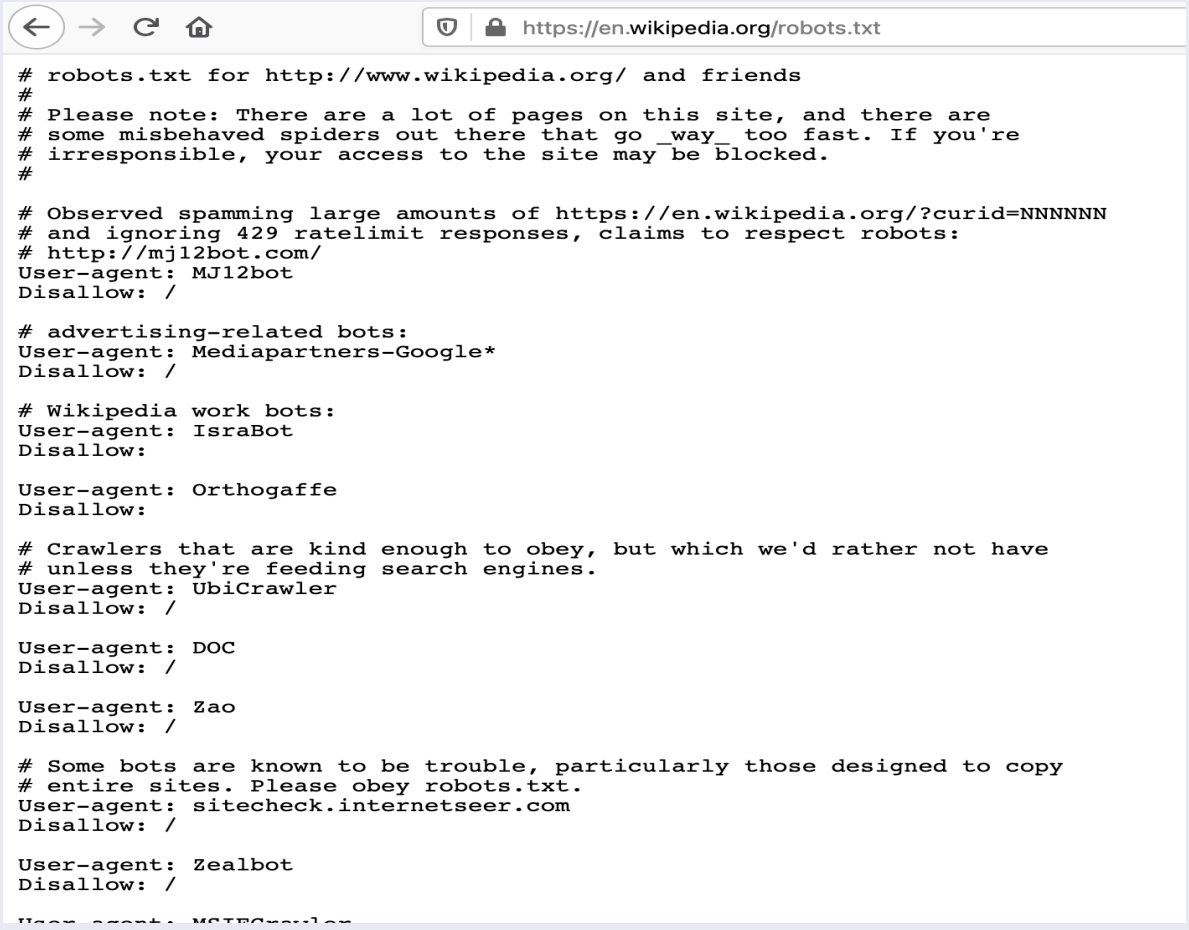
When should I use it?

Best-practices

- Only scrape publicly available data
  - i.e. You can access the page on the web without logging in
- Do not scrape copyright protected data
- Try not to violate website terms of service
- Do not burden the website
  - Limit the number of calls you make (similar to rate-limiting in APIs)
- Avoid using the data in a way that may interfere with business
  - i.e. Don't copy valuable data from a small business and share it on Github

How to scrape a web page

Start by looking up “robots.txt”



How to scrape a web page

Decoding `robots.txt`

- `User-agent` = the name of the scraper
  - `*` = All scrapers
- `Allow: /path/` = OK to scrape
- `Disallow: /path/` = Not OK to scrape
  - `Disallow: /` = Not OK to scrape any pages
- `Crawl-Delay: N` = Wait `N` miliseconds between each call to the website

How to scrape a web page

Terminology

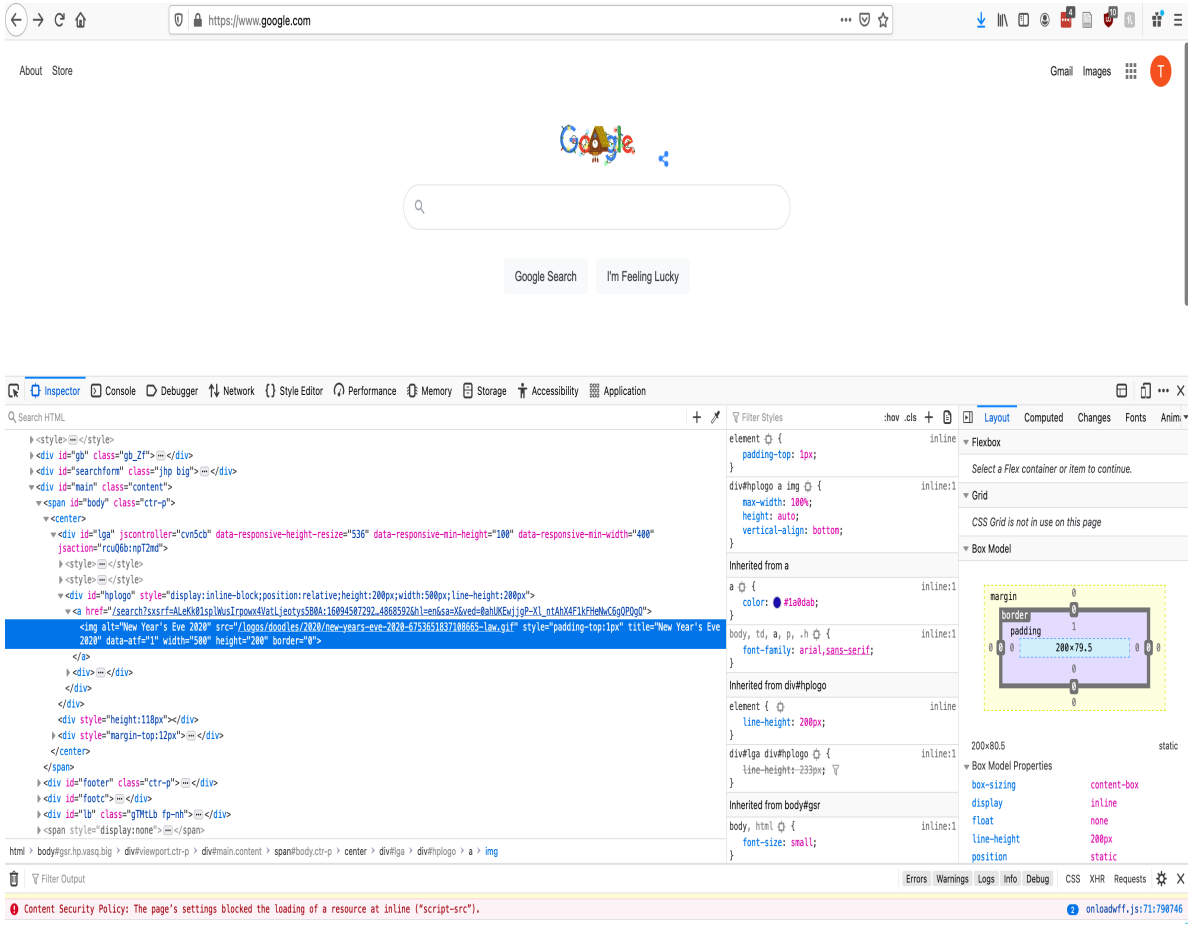
- A web-page is loaded using a **URL** (Uniform Resource Locator)
- The underlying code we are interested in is usually **HTML** (Hypertext Markup Language)
- Many websites use **CSS** (Cascading Style Sheets) to structure HTML
  - This will help us to find what we are interested in
    - See https://flukeout.github.io/ for an interactive tutorial on using CSS selectors
    - Chrome Plugin to help find CSS elements: https://selectorgadget.com/

How to scrape a web page

Inspecting HTML

- Open up a website and right click on any text or image on the screen
  - You should see an option `Inspect Element`
  - This will allow you to see the code used to generate the page

How to scrape a web page

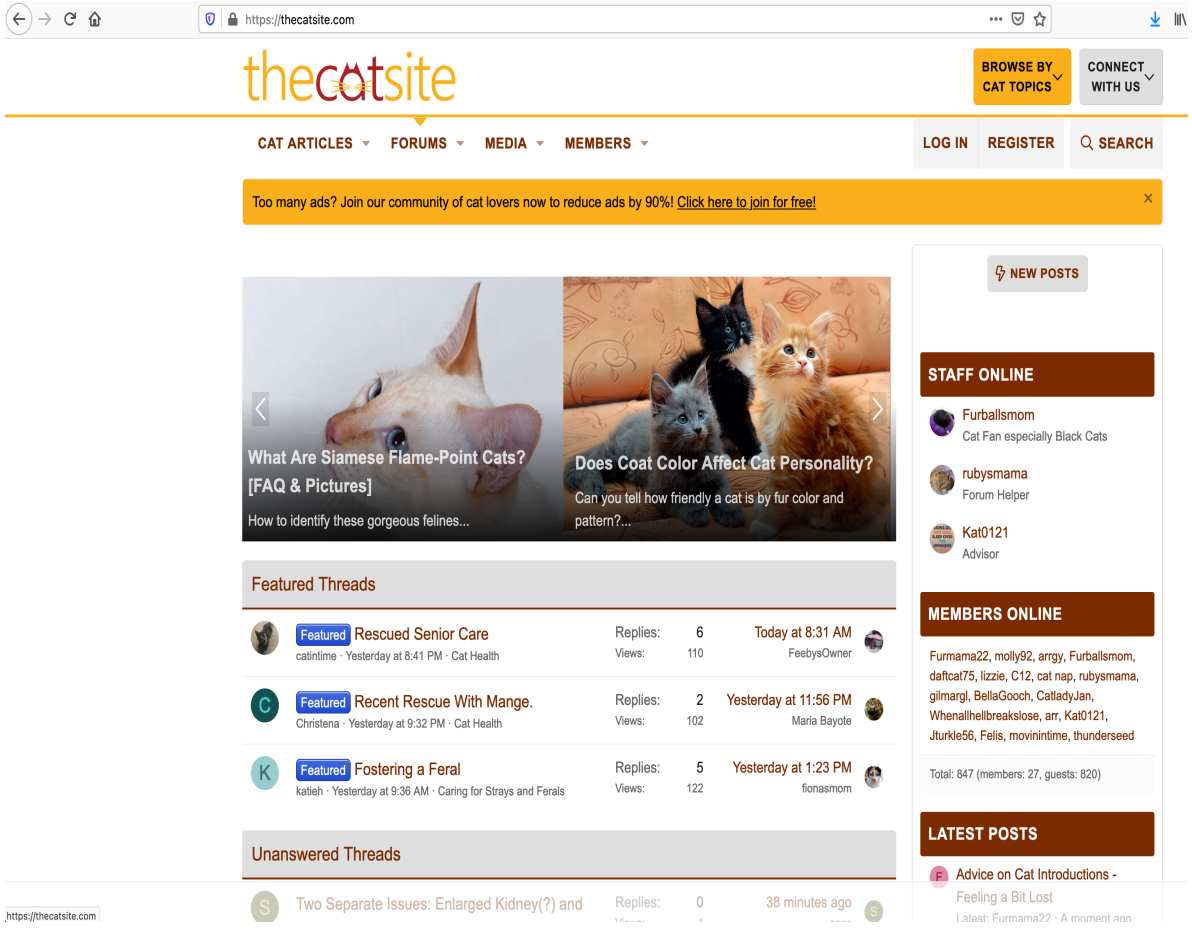


How to scrape a web page

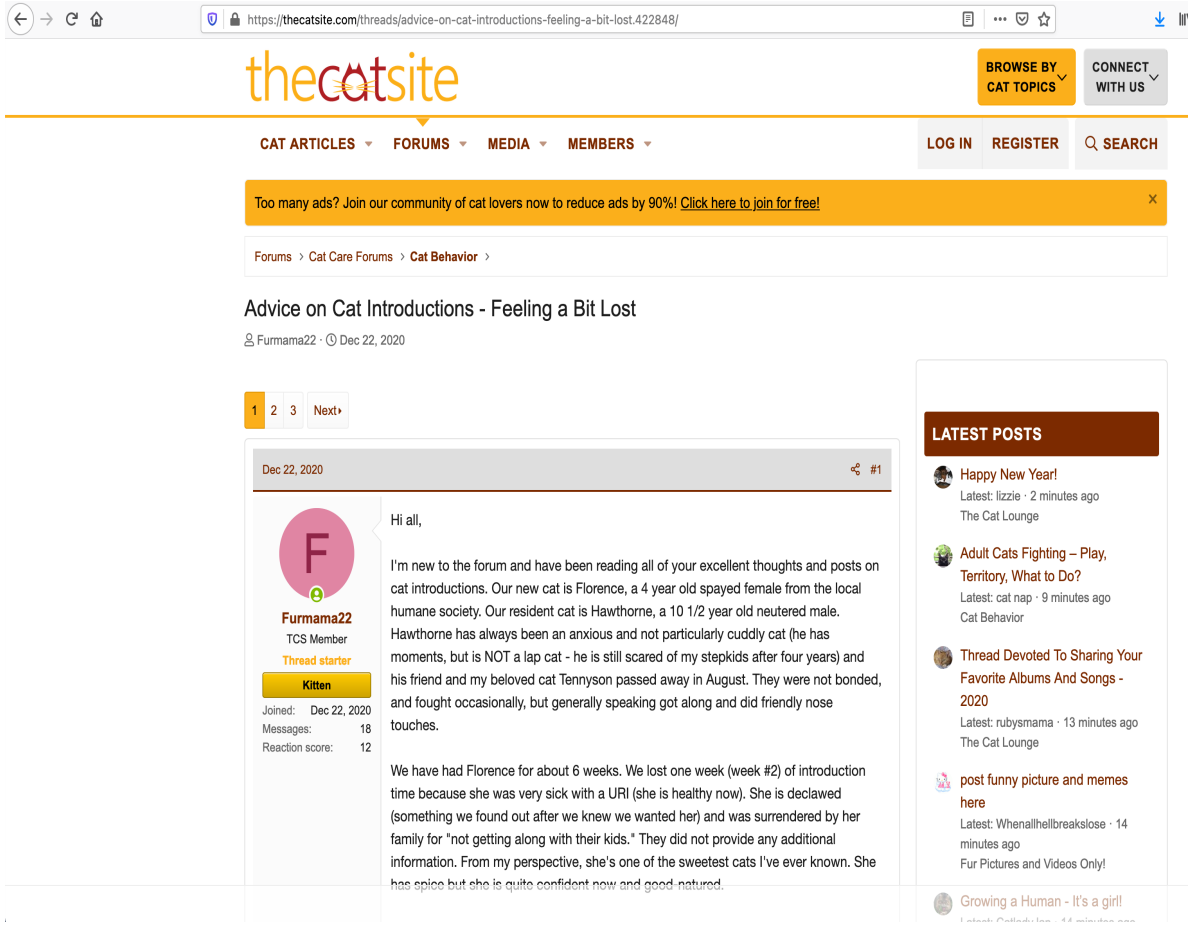
Using `rvest` to scrape HTML

```
library(rvest)
library(dplyr)
library(stringr)
```

How to scrape a web page



How to scrape a web page



How to scrape a web page

Using `rvest` to scrape HTML

```
url <- "https://thecatsite.com/threads/advice-on-cat-introductions-feel
thread <- read_html(url)
```

How to scrape a web page

Collecting messages

First, we parse the HTML to obtain the text of each message on the page. Here we use the CSS selector `.message-body`, which selects all elements with class `message-body`.

```
messages <- thread %>% html_nodes(".message-body") %>%
  html_text() %>% str_trim()
print(length(messages))
```

```
## [1] 20
```

```
print(substr(messages[[1]], 1, 50))
```

```
## [1] "Hi all,\nI'm new to the forum and have been reading"
```

How to scrape a web page

Getting user names

Next we collect the name of each user. User information is found by parsing the `.message-userDetails` node.

```
users <- thread %>% html_nodes(".message-userDetails") %>%
  html_text() %>% str_trim()
print(length(users))
```

```
## [1] 20
```

```
users[[1]]
```

```
## [1] "Furmama22\nTCS Member\nThread starter\nYoung Cat"
```

```
users <- thread %>% html_nodes(".message-userDetails") %>%
  html_text() %>% str_trim() %>% str_split('\n') %>% pluck(1)
users[[1]]
```

```
## [1] "Furmama22"
```

We do some string manipulation to get the user names.

How to scrape a web page

Collecting timestamps

Finally, we also want to get the time-stamp of each message. While the forum only displays dates, we can actually get the full timestamp.

```
dates <- thread %>% html_nodes("time.u-dt")
length(dates)
```

```
## [1] 27
```

```
dates <- thread %>% html_nodes(".u-concealed .u-dt")
length(dates)
```

```
## [1] 21
```

```
#dates <- date[-1]
dates <- dates %>% html_attr("datetime")
dates[[1]]
```

```
## [1] "2020-12-22T11:26:12-0800"
```

Note how `time.u-dt` returns too much information, so `.u-concealed .u-dt` is selected instead.

How to scrape a web page

Putting it all together

```
get.posts <- function(thread) {
  messages <- thread %>% html_nodes(".message-body") %>%
    html_text() %>% str_trim()  %>%
    str_trunc(15, "right") # only get 1st 15 chars
  users <- thread %>% html_nodes(".message-userDetails") %>%
    html_text() %>% str_trim() %>%
    str_split('\n') %>% pluck(1)
  timestamps <- thread %>% html_nodes(".u-concealed .u-dt") %>%
    html_attr("datetime")
  timestamps <- timestamps[-1]
  df <- data.frame(messages, unlist(users), timestamps)
  colnames(df) <- c("message","user", "timestamp")
  return(df)
}
```

How to scrape a web page

Putting it all together

```
results <- get.posts(thread)
results[1:5,]
```

```
##                message             user                 timestamp
## 1 Hi all,\nI'm ...        Furmama22 2020-12-22T11:26:12-0800
## 2  Furmama22 sa... calicosrspecial 2020-12-22T13:13:04-0800
## 3  Thank you SO...        Furmama22 2020-12-22T14:01:53-0800
## 4  I don't thin...     Mamanyt1953 2020-12-22T18:21:00-0800
## 5  Thanks so mu...        Furmama22 2020-12-22T18:52:26-0800
```

How to scrape a web page

Pagination

```
links <- thread %>% html_nodes(".pageNav-jump") %>%
  html_attr("href")
desc <- thread %>% html_nodes(".pageNav-jump") %>%
  html_text()
pagination.info <- data.frame(links, desc) %>%
  filter(str_detect(desc, "Next")) %>% distinct()
base <- "https://thecatsite.com"
next.page <- paste(base, pagination.info$links, sep = '')
```

How to scrape a web page

Pagination

```
results <- get.posts(read_html(next.page))
results[1:5,]
```

```
##           message        user                 timestamp
## 1 Thank you al...   Furmama22 2020-12-25T10:57:49-0800
## 2 Sounds like ... Mamanyt1953 2020-12-25T19:04:14-0800
## 3 Well I suppo...   Furmama22 2020-12-26T07:25:30-0800
## 4 AWWWWWWWWW! ... Mamanyt1953 2020-12-26T07:39:55-0800
## 5      Thank you!   Furmama22 2020-12-26T07:50:41-0800
```

How to scrape a web page

Pagination function

```
get.next.page <- function(thread){
  links <- thread %>% html_nodes(".pageNav-jump") %>%
    html_attr("href")
  desc <- thread %>% html_nodes(".pageNav-jump") %>%
    html_text()
  pagination.info <- data.frame(links, desc) %>%
    filter(str_detect(desc, "Next")) %>% distinct()
  base <- "https://thecatsite.com"
  next.page <- paste(base, pagination.info$links, sep = '')
  return(next.page)
}
get.next.page(thread)
```

```
## [1] "https://thecatsite.com/threads/advice-on-cat-introductions-feel
```

How to scrape a web page

Testing the pagination function

```
thread.2 <- read_html(get.next.page(thread))
pagination.2 <- get.next.page(thread.2)
pagination.2
```

```
## [1] "https://thecatsite.com/threads/advice-on-cat-introductions-feel
```

How to scrape a web page

Testing the pagination function

```
thread.3 <- read_html(get.next.page(thread.2))
pagination.3 <- get.next.page(thread.3)
pagination.3
```

```
## [1] "https://thecatsite.com/threads/advice-on-cat-introductions-feel
```

How to scrape a web page

Improving the function

```
get.next.page <- function(thread){
  links <- thread %>% html_nodes(".pageNav-jump") %>%
    html_attr("href")
  desc <- thread %>% html_nodes(".pageNav-jump") %>%
    html_text()
  pagination.info <- data.frame(links, desc) %>%
    filter(str_detect(desc, "Next")) %>% distinct()
  if (dim(pagination.info)[1] == 1) {
  base <- "https://thecatsite.com"
  next.page <- paste(base, pagination.info$links, sep = '')
  return(next.page)} else {
    return("Final page")
  }
}
```

How to scrape a web page

Testing the pagination function

```
pagination.3 <- get.next.page(thread.3)
pagination.3
```

```
## [1] "https://thecatsite.com/threads/advice-on-cat-introductions-feel
```

```
pagination.2 <- get.next.page(thread.2)
pagination.2
```

```
## [1] "https://thecatsite.com/threads/advice-on-cat-introductions-feel
```

How to scrape a web page

Paginate and scrape

```
paginate.and.scrape <- function(url){
  thread <- read_html(url)
  posts <- get.posts(thread)
  next.page <- get.next.page(thread)
  while (!str_detect(next.page, "Final page"))
  {
    thread <- read_html(next.page)
    posts <- rbind(posts, get.posts(thread))
    next.page <- get.next.page(thread)
    Sys.sleep(1) # wait 1 second
  }
  return(posts)
}
```

How to scrape a web page

Paginate and scrape

```
full.thread <- paginate.and.scrape(url)
length(full.thread)
```

```
## [1] 3
```

```
print(full.thread)
```

```
##                message              user                 timestamp
## 1   Hi all,\nI'm ...       Furmama22 2020-12-22T11:26:12-0800
## 2    Furmama22 sa... calicosrspecial 2020-12-22T13:13:04-0800
## 3    Thank you SO...       Furmama22 2020-12-22T14:01:53-0800
## 4    I don't thin...     Mamanyt1953 2020-12-22T18:21:00-0800
## 5    Thanks so mu...       Furmama22 2020-12-22T18:52:26-0800
## 6    You certainl...     Mamanyt1953 2020-12-22T19:06:23-0800
## 7    Furmama22 sa... calicosrspecial 2020-12-23T08:46:09-0800
## 8    This is grea...       Furmama22 2020-12-23T08:57:29-0800
## 9    Furmama22 sa...          pearl99 2020-12-23T09:17:28-0800
## 10   Furmama22 sa... calicosrspecial 2020-12-23T09:19:15-0800
## 11  Thank you \nC...       Furmama22 2020-12-23T14:23:36-0800
```

How to scrape a web page

Crawling a website

```
get.threads <- function(url) {
  f <- read_html(url)
  title <- f %>% html_nodes(".structItem-title") %>%
    html_text() %>% str_trim()
  link <- f %>% html_nodes(".structItem-title a") %>%
    html_attr("href")  %>% str_trim()
  link <- data.frame(link)
  link <- link %>% filter(str_detect(link, "/threads/"))
  threads <- data.frame(title, link)
  return(threads)
}
```

How to scrape a web page

Crawling a website

```
forum.url <- "https://thecatsite.com/forums/cat-behavior.5/"

threads <- get.threads(forum.url)
```

How to scrape a web page

Crawling a website

```
print(threads$title)
```

```
##  [1] "I need help with a kitten I'm cat sitting for two weeks"
##  [2] "One cat won't share a litter box, the other uses all of them h
##  [3] "Advice on Cat Introductions - Feeling a Bit Lost"
##  [4] "My cat is very vocal. I cannot figure him out..."
##  [5] "Should We Get a Second Cat?"
##  [6] "3 Week business trip and I'm super stressed"
##  [7] "New Cat - Very Timid"
##  [8] "Reintroducing lost kitty"
##  [9] "Resident cat refuses to eat after hearing the new cat"
## [10] "Siamese 5 year old male changed sleeping habits?"
## [11] "Featured\nAdvice on Next Steps with Timid Cat"
## [12] "Is my cat depressed because of new kitten?"
## [13] "Kitten in heat: litterbox problem?"
## [14] "Cat suddenly changed."
## [15] "Does Prozac really work?"
## [16] "Stressed cat with no obvious solution"
```

How to scrape a web page

Crawling a website

```
print(threads$link)
```

```
##  [1] "/threads/i-need-help-with-a-kitten-im-cat-sitting-for-two-week
##  [2] "/threads/one-cat-wont-share-a-litter-box-the-other-uses-all-of
##  [3] "/threads/advice-on-cat-introductions-feeling-a-bit-lost.422848
##  [4] "/threads/my-cat-is-very-vocal-i-cannot-figure-him-out.422683/"
##  [5] "/threads/should-we-get-a-second-cat.423393/"
##  [6] "/threads/3-week-business-trip-and-im-super-stressed.423394/"
##  [7] "/threads/new-cat-very-timid.422782/"
##  [8] "/threads/reintroducing-lost-kitty.423373/"
##  [9] "/threads/resident-cat-refuses-to-eat-after-hearing-the-new-cat
## [10] "/threads/siamese-5-year-old-male-changed-sleeping-habits.42320
## [11] "/threads/advice-on-next-steps-with-timid-cat.423315/"
## [12] "/threads/is-my-cat-depressed-because-of-new-kitten.423377/"
## [13] "/threads/kitten-in-heat-litterbox-problem.423354/"
## [14] "/threads/cat-suddenly-changed.423370/"
## [15] "/threads/does-prozac-really-work.423174/"
## [16] "/threads/stressed-cat-with-no-obvious-solution.423345/"
```

How to scrape a web page

Crawling a website

Exercise: Write a function to iterate through the 10 pages of threads, each time calling `get.threads` to collect all threads and then `paginate.and.scrape` for each thread. Store the results in a single dataframe.

```
# Complete function here
```

How to scrape a web page

Javascript and browser automation

- Many websites use Javascript
  - This can cause problems for web-scrapers as it cannot directly be parsed to HTML
- Rather than loading HTML directly into R, we can use R to automate a browser
  - Selenium WebDriver and the package `RSelenium` (https://github.com/ropensci/RSelenium) is the most popular approach

How to scrape a web page

What can RSelenium do?

- Extract HTML from Javascript-based websites
- Interact with web-based content
  - e.g., Click "OK" to a warning, complete a search box
- **However**, `RSelenium` does not currently work directly in RMarkdown and requires a more complicated set up using a *Docker container* so I will not demo it today

How to scrape a web page

Data storage and logging

- If collecting a lot of data, use a server to run a scrape and store scraped data
- Write output to a database
  - This helps to organize the data and makes it easier to query and manage
- Keep a log file with a record of which pages you have scraped
  - You could also use Slack to send progress updates

References

- Fiesler, Casey, Nate Beard, and Brian C Keegan. 2020. "No Robots, Spiders, or Scrapers: Legal and Ethical Regulation of Data Collection Methods in Social Media Terms of Service." In Proceedings of the Fourteenth International AAAI Conference on Web and Social Media, 187–96. AAAI.

Questions