Webscraping with RSelenium

Automate your browser actions

Etienne Bacher

LISER

2022-08-04

Introduction

Scraping can be divided in two steps:

1. getting the HTML that contains the information
2. cleaning the HTML to extract the information we want

These 2 steps don't require the same tools, and *shouldn't be made at the same time*.

Here, we will focus on step 1: *how to get the HTML we need with dynamic webpages?*

Static vs dynamic

Static webpage: all the information is loaded with the page.

Example: Wikipedia.

Dynamic webpage: the website uses JavaScript to fetch data from their servers and *dynamically* update the page.

Example: see later.

(R)Selenium

Idea

Idea: control the browser from the command line.

I wish I could click on this button to open a modal

```
remote_driver$
  findElement(using = "css", value = ".my-button")$
  clickElement()
```

I wish I could fill these inputs to automatically connect

```
remote_driver$
  findElement(using = "id", value = "password")$
  sendKeysToElement(list("my_super_secret_password"))
```

Almost everything you can do "by hand" in a browser, you can reproduce with Selenium:

- open a browser
- click on something
- enter values
- go to previous/next page
- refresh the page
- get all the HTML that is currently displayed

- `open()` / `navigate()`
- `clickElement()`
- `sendKeysToElement()`
- `goBack()` / `goForward()`
- `refresh()`
- **`getPageSource()`**

...

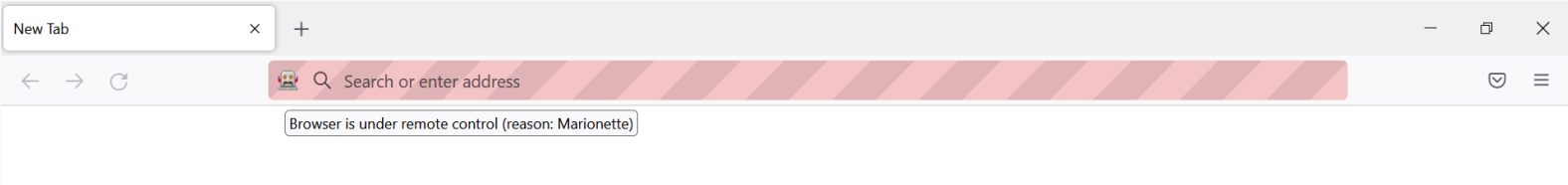
Get started

In the beginning there was ~~light~~ `rsDriver()`:

```r
# if not already installed
# install.packages("RSelenium")
library(RSelenium)

driver <- rsDriver(browser = "firefox") # can also be chrome
remote_driver <- driver[["client"]]
```

This will print a bunch of messages and open a "marionette browser".



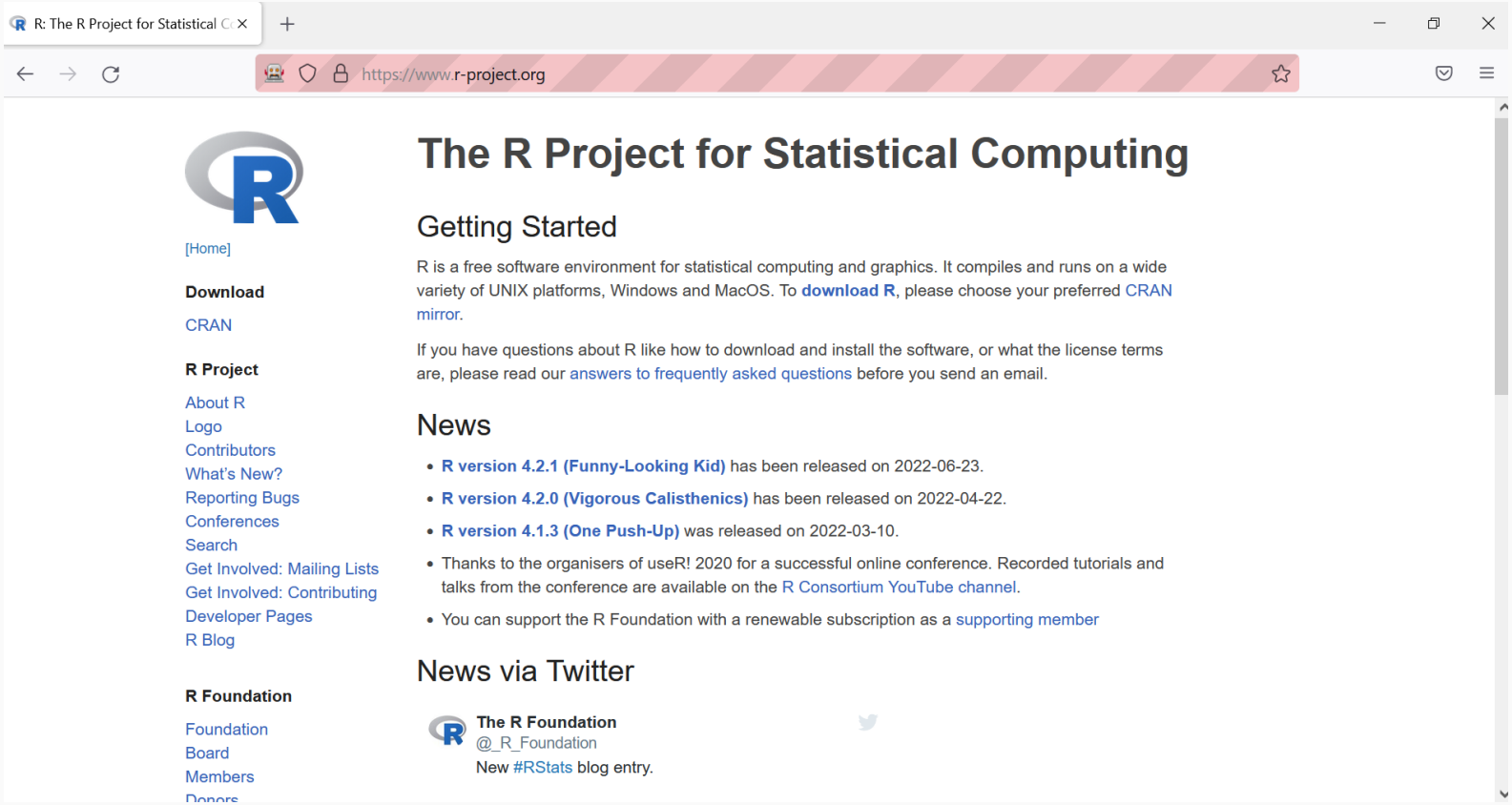
From now on, everything we do is calling `<function>()` starting with `remote_driver$`[1].

Objective: get the list of core contributors to R located here.

[1]: Or whatever you called it in the previous step

Navigate

```
remote_driver$navigate("https://r-project.org")
```



Click on "Contributors"

This requires two things:

1. find the element
2. click on it

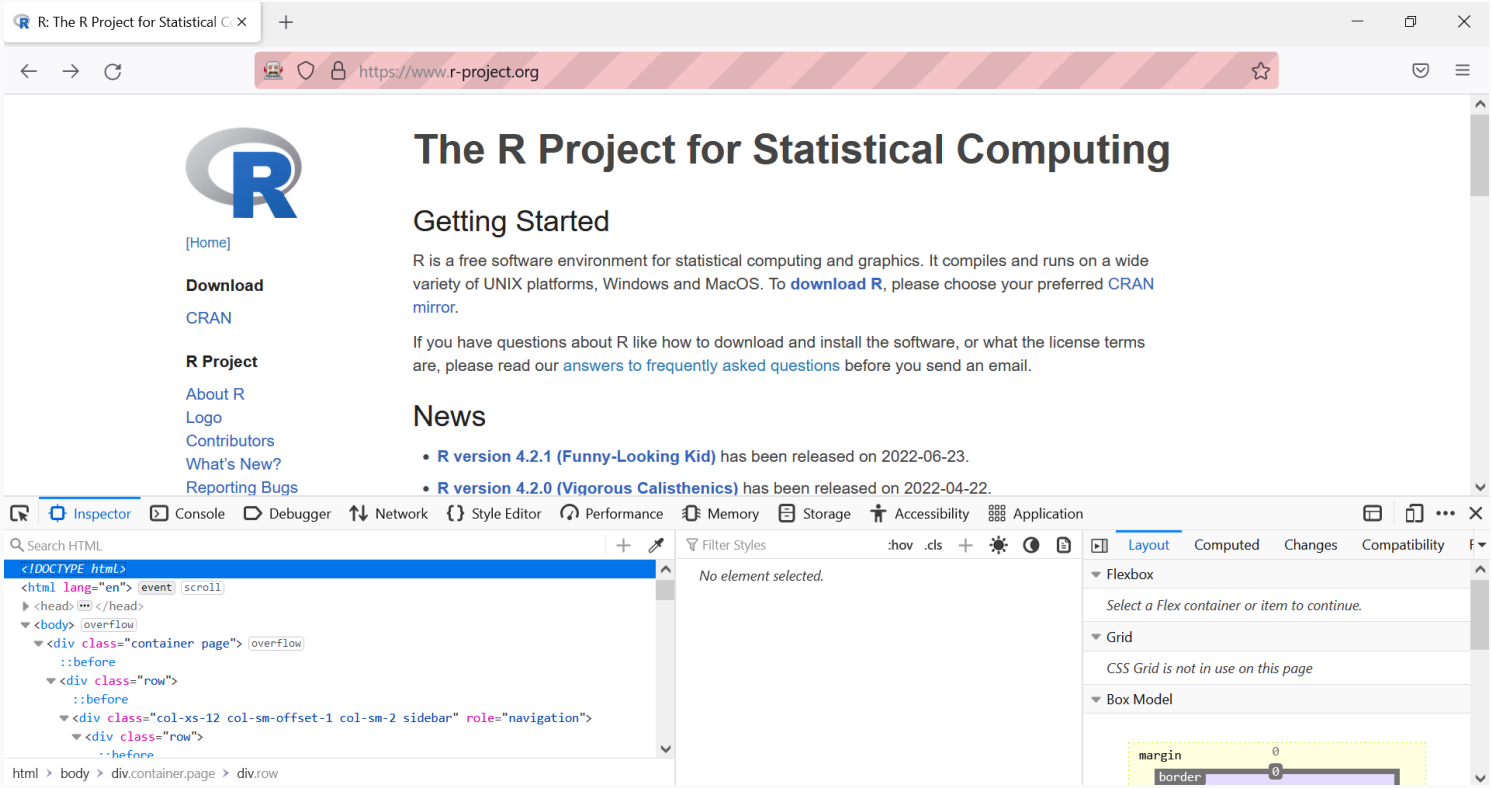
Humans -> eyes

Computers -> HTML/CSS

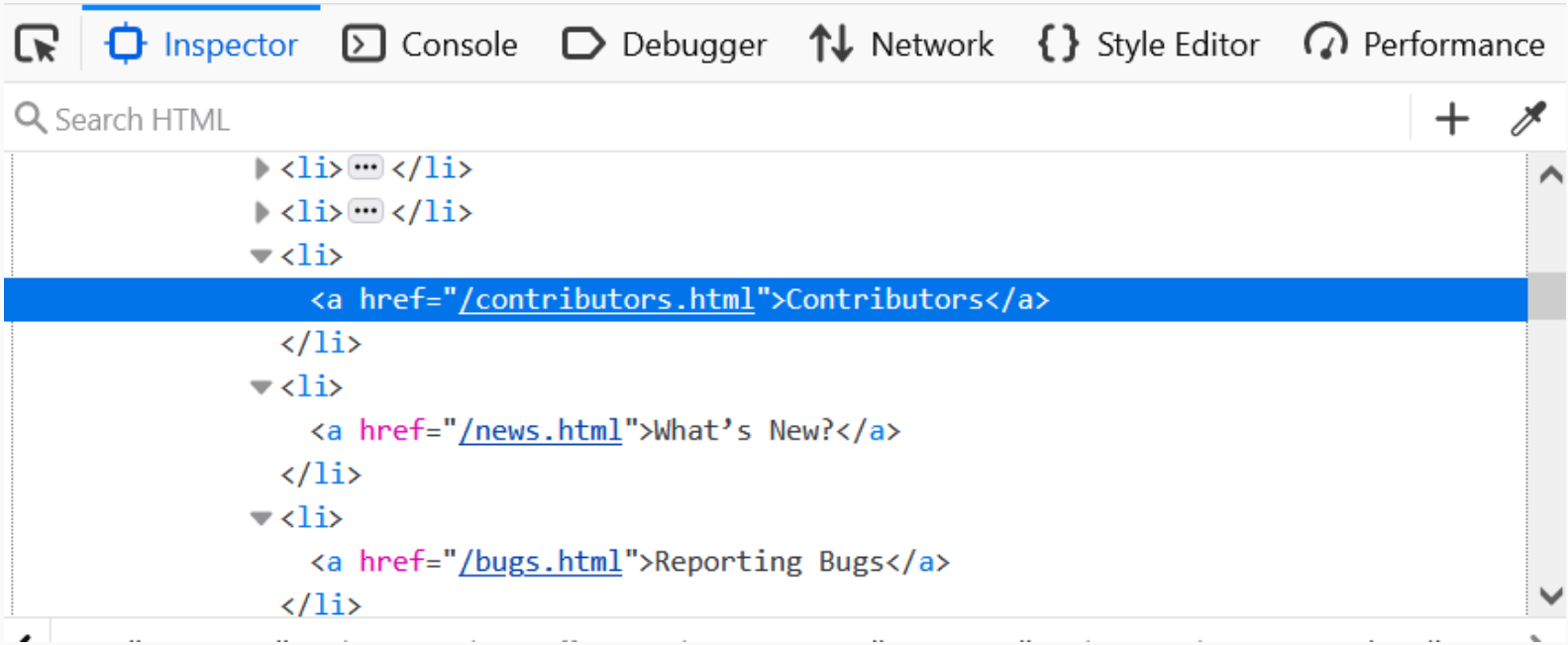
To find the element, we need to open the console to see the structure of the page.

Several ways to do it:

- right-click -> "Inspect"
- `Ctrl` + `Shift` + `C`



Then, hover the element we're interested in: the link "Contributors".



How can we find this with `RSelenium`?

```
?RSelenium::remoteDriver
```

-> `findElement`

- class name ❌
- id ❌
- name ❌
- tag name ❌

- css selector ✓
- link text ✓
- partial link text ✓
- xpath ✓

All of these work:

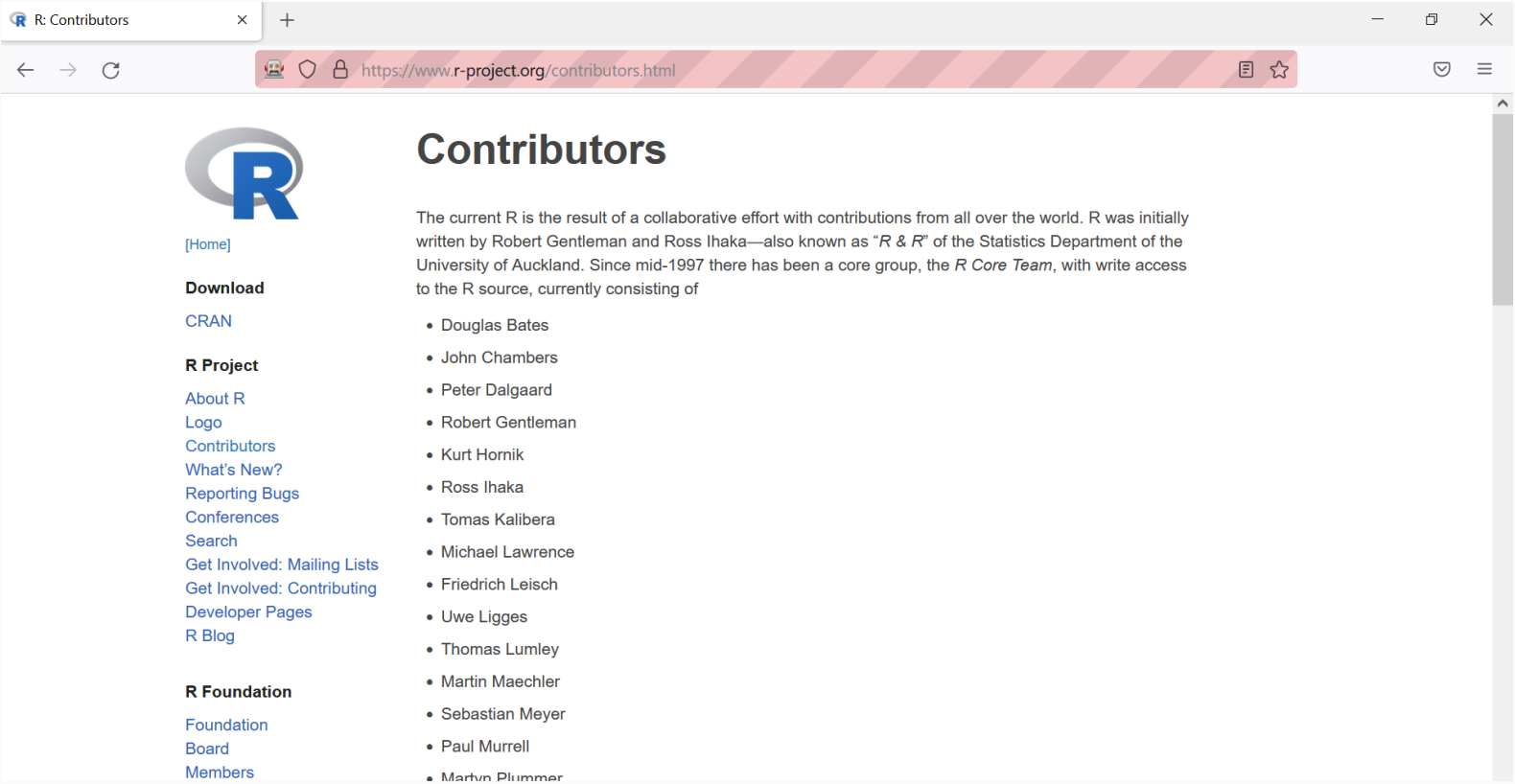
```
remote_driver$
  findElement("link text", "Contributors")$
  clickElement()

remote_driver$
  findElement("partial link text", "Contributors")$
  clickElement()

remote_driver$
  findElement("xpath", "/html/body/div/div[1]/div[1]/div/div[1]/ul/li[3]
  clickElement()

remote_driver$
  findElement("css selector", "div.col-xs-6:nth-child(1) > ul:nth-child(
  clickElement()
```

We are now on the right page!



Last step: obtain the HTML of the page.

```
remote_driver$getPageSource()
```

To read it with `rvest`:

```
x <- remote_driver$getPageSource()[[1]]
rvest::read_html(x)
```

Do we read the HTML and extract the information in the same script?

No!

Rather, we save the HTML in an external file, and we will be able to access it in another script (and offline) to manipulate it as we want[1].

```
write(x, file = "contributors.html")
# Later and in another script
rvest::read_html("contributors.html")
```

Click here to see the results.

[1]: Although, in this case, it wouldn't cost too much to treat it directly in the same script.

A harder & real-life example

The previous example was not a *dynamic* page: we could have used the link to the page and apply webscraping methods for static webpages.

Let's now dive into a more complex example, where RSelenium is the only way to scrape.

Appendix

For reference, here's the code to extract the list of contributors:

```r
library(rvest)

html <- read_html("contributors.html")

bullet_points <- html %>%
  html_elements(css = "div.col-xs-12 > ul > li") %>%
  html_text()

blockquote <- html %>%
  html_elements(css = "div.col-xs-12.col-sm-7 > blockquote") %>%
  html_text() %>%
  strsplit(., split = ", ")

blockquote <- blockquote[[1]] %>%
  gsub("\\r|\\n|\\.|and", "", .)

others <- html %>%
  html_elements(xpath = "/html/body/div/div[1]/div[2]/p[5]") %>%
  html_text() %>%
  strsplit(., split = ", ")

others <- others[[1]] %>%
  gsub("\\r|\\n|\\.|and", "", .)

all_contributors <- c(bullet_points, blockquote, others)
```

Appendix

```
##   [1] "Douglas Bates"          "John Chambers"          "Peter Dalgaard"         "Robert Gentleman"
##   [6] "Ross Ihaka"             "Tomas Kalibera"         "Michael Lawrence"       "Friedrich Leisch"
##  [11] "Thomas Lumley"          "Martin Maechler"        "Sebastian Meyer"        "Paul Murrell"
##  [16] "Brian Ripley"           "Deepayan Sarkar"        "Duncan Temple Lang"     "Luke Tierney"
##  [21] "Valerio Aimale"         "Suharto Anggono"        "Thomas Baier"           "Gabe Becker"
##  [26] "Roger Biv"              "Ben Bolker"             "David Brahm"            "Göran Broström"
##  [31] "Vince Carey"            "Saikat DebRoy"          "Matt Dowle"             "Brian D'Urso"
##  [36] "Dirk Eddelbuettel"      "Claus Ekstrom"          "Sebastian Fischmeister" "John Fox"
##  [41] "Yu Gong"                "Gabor Grothendieck"     "Frank E Harrell Jr"     "Peter M Haverty"
##  [46] "Robert King"            "Kjetil Kjernsmo"        "Roger Koenker"          "Philippe Lambert"
##  [51] "Jim Lindsey"            "Patrick Lindsey"        "Catherine Loader"       "Gordon Maclean"
##  [56] "John Maindonald"        "David Meyer"            "Ei-ji Nakama"           "Jens Oehlschägel"
##  [61] "Richard O'Keefe"        "Hubert Palme"           "Roger D Peng"           "José C Pinheiro"
##  [66] "Anthony Rossini"        "Jonathan Rougier"       "Petr Savicky"           "Günther Sawitzki"
##  [71] "Arun Srinivasan"        "Detlef Steuer"          "Bill Simpson"           "Gordon Smyth"
##  [76] "Terry Therneau"         "Rolf Turner"            "Bill Venables"          "Gregory R Warnes"
##  [81] "Morten Welinder"        "James Wettenhall"       "Simon Wood"             " Achim Zeileis"
##  [86] "David J Best"           "Richard Brent"          "Kevin Buhr"             "Michael A Covington"
##  [91] "Robert Clevel,"         "G W Cran"               "C G Ding"               "Ulrich Drepper"
##  [96] "J O Evans"              "David M Gay"            "H Frick"                "G W Hill"
## [101] "Eric Grosse"            "Shelby Haberman"        "Bruno Haible"           "John Hartigan"
## [106] "Trevor Hastie"          "Min Long Lam"           "George Marsaglia"       "K J Martin"
## [111] "C R Mckenzie"           "Jean McRae"             "Cyrus Mehta"            "Fionn Murtagh"
## [116] "Finbarr O'Sullivan"     "R E Odeh"               "William Patefield"      "Nitin Patel"
## [121] "D E Roberts"            "Patrick Royston"        "Russell Lenth"          "Ming-Jen Shyu"
## [126] "S G Springer"           "Supoj Sutanthavibul"    "Irma Terpenning"        "G E Thomas"
## [131] "Wai Wan Tsang"          "Berwin Turlach"         "Gary V Vaughan"         "Michael Wichura"
## [136] "M A Wong"
```

Back