

# Class 7: Web Scraping in R

## MAST5953: Web Scraping and Text Mining

Dr. Miriam Sorace

[www.miriamsorace.net](http://www.miriamsorace.net)

28 January 2022

# Outline of Today's Class

Web Scraping Recap

Web Scraping Methods

Scraping in R: Core Functions w. Examples

## Web Scraping Recap

# Web Scraping

- ▶ Scrapers: programmes that grab *specific* content from webpages
- ▶ Spiders (or Crawlers): programmes that index entire pages and follow every link.
- ▶ Web scraping involves some spidering when - for example - parsing entire web-pages. But scrapers are *selective/purposive*

# Web Scraping: Managing Expectations

- ▶ It is rarely a one-shot affair: websites change constantly
- ▶ Difficult to generalise: each website is different
- ▶ Writing a scraping code that works well takes a long time

# Web Scraping: Netiquette!

- ▶ Search for & respect terms of service
  - ▶ add “/robot.txt” at the end of URL
- ▶ Careful with personal/private data
- ▶ Check if there are APIs or ready-to-download files already
- ▶ Stay identifiable (do not mask your IP)
- ▶ Reduce traffic as much as possible
  - ▶ light/efficient scrapers
  - ▶ use pauses (`Sys.sleep()`)
  - ▶ avoid over-scraping

# Web Scraping Steps

1. Analyse the structure of the webpage (e.g. HTML/XML trees)
  - ▶ Identify relevant nodes
  - ▶ [www.selectorgadget.com](http://www.selectorgadget.com)
2. Test: write scraper and debug
3. Execute the download

## First Task: Install SelectorGadget!



## SelectorGadget

Offered by: [selectorgadget.com](http://selectorgadget.com)

★★★★★ 83 | [Developer Tools](#) |  100,000+ users

Remove from Chrome

## Overview

## Reviews

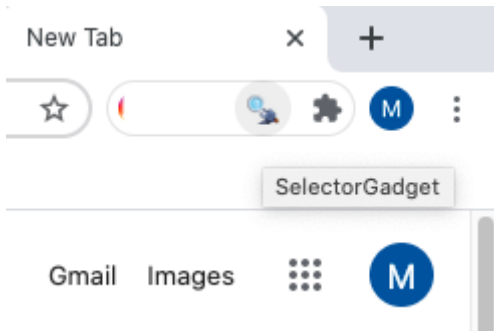
Support

## Related





## First Task: Install SelectorGadget!



## Web Scraping Methods

## Extraction Methods

1. Regular Expressions
2. Node queries
3. APIs (*Application Programming Interfaces*)
  - ▶ We'll explore them in the next class on social media data

# Regular Expressions

- ▶ Scraper works by extracting from meaningful alphanumeric patterns in the webpage text
  - ▶ E.g.: grab all text in-between quotes ("").

## Regular Expressions

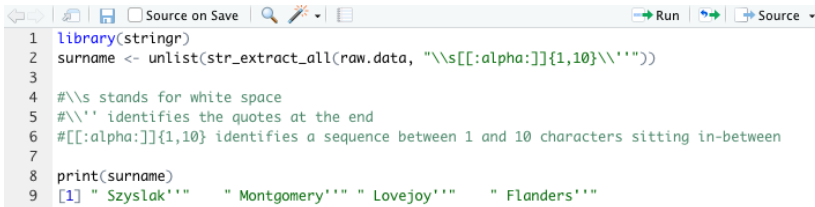
```
[1] "Moe Szyslak"      "Burns, C. Montgomery" "Rev. Timothy Lovejoy"  
[4] "Ned Flanders"
```

- ▶ Suppose from the web content above, you wanted to extract the surnames
- ▶ You can exploit the regular pattern of surnames sitting between a space and closing with "

# Regular Expressions

```
[1] "Moe Szyslak"          "Burns, C. Montgomery" "Rev. Timothy Lovejoy"  
[4] "Ned Flanders"
```

► You would write a regular expression scraper as the below



```
1 library(stringr)  
2 surname <- unlist(str_extract_all(raw.data, "\\s[:alpha:]{1,10}\\''"))  
3  
4 #\\s stands for white space  
5 #\\'' identifies the quotes at the end  
6 #[:alpha:]{1,10} identifies a sequence between 1 and 10 characters sitting in-between  
7  
8 print(surname)  
9 [1] " Szyslak'" " Montgomery'" " Lovejoy'" " Flanders'"
```

# Regular Expressions

---

<code>[:digit:]</code>	Digits: 0 1 2 3 4 5 6 7 8 9
<code>[:lower:]</code>	Lowercase characters: a–z
<code>[:upper:]</code>	Uppercase characters: A–Z
<code>[:alpha:]</code>	Alphabetic characters: a–z and A–Z
<code>[:alnum:]</code>	Digits and alphabetic characters

---

<code>[:punct:]</code>	Punctuation characters: . , ; etc.
<code>[:graph:]</code>	Graphical characters: <code>[:alnum:]</code> and <code>[:punct:]</code>
<code>[:blank:]</code>	Blank characters: Space and tab
<code>[:space:]</code>	Space characters: Space, tab, newline, and other space characters
<code>[:print:]</code>	Printable characters: <code>[:alnum:]</code> , <code>[:punct:]</code> and <code>[:space:]</code>

---

*Source:* Adapted from <http://stat.ethz.ch/R-manual/R-patched/library/base/html/regex.html>

# Regular Expressions

?	The preceding item is optional and will be matched at most once
*	The preceding item will be matched zero or more times
+	The preceding item will be matched one or more times
{n}	The preceding item is matched exactly <i>n</i> times
{n, }	The preceding item is matched <i>n</i> or more times
{n, m}	The preceding item is matched at least <i>n</i> times, but not more than <i>m</i> times

Source: Adapted from <http://stat.ethz.ch/R-manual/R-patched/library/base/html/regex.html>

**Table 8.3** Selected symbols with special meaning

\w	Word characters: <code>[[:alnum:]]_</code>
\W	No word characters: <code>[^[:alnum:]]_</code>
\s	Space characters: <code>[[:blank:]]</code>
\S	No space characters: <code>[^[:blank:]]</code>
\d	Digits: <code>[[:digit:]]</code>
\D	No digits: <code>[^[:digit:]]</code>
\b	Word edge
\B	No word edge
\<	Word beginning
\>	Word end



# Regular Expressions

Function	Description	Output
<i>Functions using regular expressions</i>		
<code>str_extract()</code>	Extracts first string that matches pattern	Character vector
<code>str_extract_all()</code>	Extracts all strings that match pattern	List of character vectors
<code>str_locate()</code>	Returns position of first pattern match	Matrix of start/end positions
<code>str_locate_all()</code>	Returns positions of all pattern matches	List of matrices
<code>str_replace()</code>	Replaces first pattern match	Character vector
<code>str_replace_all()</code>	Replaces all pattern matches	Character vector
<code>str_split()</code>	Splits string at pattern	List of character vectors
<code>str_split_fixed()</code>	Splits string at pattern into fixed number of pieces	Matrix of character vectors
<code>str_detect()</code>	Detects patterns in string	Boolean vector
<code>str_count()</code>	Counts number of pattern occurrences in string	Numeric vector
<i>Further functions</i>		
<code>str_sub()</code>	Extracts strings by position	Character vector
<code>str_dup()</code>	Duplicates strings	Character vector
<code>str_length()</code>	Returns length of string	Numeric vector
<code>str_pad()</code>	Pads a string	Character vector
<code>str_trim()</code>	Discards string padding	Character vector
<code>str_c()</code>	Concatenates strings	Character vector

# Regular Expressions

## Advantages

- ▶ Only useful when HTML/XML are malformed **and** clear patterns are retrievable in the page
- ▶ Mostly used in creating URL lists to scrape multiple pages
- ▶ Mostly used in data cleaning/variable creation

# Regular Expressions

## Disadvantages

- ▶ Building and interpreting a regular expression scraper is challenging: hard to test and debug
- ▶ Limited applicability: not all web pages have consistent repeated patterns that *uniquely* identify a specific content
- ▶ Ignores the useful and meaningful hierarchical structure of web pages
- ▶ Strongly discouraged for webscraping, best for data cleaning

## Node Queries

- ▶ This is the most common procedure
- ▶ The scraper is built by exploiting the tree structure of the web page
  - ▶ Inspected via web developer tools & SelectorGadget
- ▶ Utilises query functions from R library such as `rvest` to locate nodes and extract information

# Node Queries

## Advantages

- ▶ Intuitive to write and read
- ▶ Uniquely identifies the relevant content

# Node Queries

## Disadvantages

- ▶ Costly when the HTML/XML structure is malformed
- ▶ Costly when web pages get constantly updated

# Application Programming Interfaces

- ▶ Scraper is built by using APIs 'wrappers', exploiting the webpage's API, which provides the website's data in pure/cleaned form
- ▶ We'll see concrete examples of this in the next class where we'll look at social media data

# Application Programming Interfaces

## Advantages

- ▶ This is the scraping gold standard.
- ▶ Very easy: clean, standardised data
- ▶ APIs codes are robust and regularly updated



# Application Programming Interfaces

## Disadvantages

- ▶ Not many websites provide their APIs
- ▶ Websites that do, often have licences/fees/limits attached to API use
- ▶ API owners often limit their functionality from one day to the next, rendering the scraper useless.

## Scraping in R: Core Functions w. Examples

# Parsing the Page

## read\_html

### Read In .Html Content

Read in the content from a .html file. This is generalized, reading in all body text. For finer control the user should utilize the xml2 and rvest packages.

**Keywords** [html](#)

### Usage

```
read_html(file, skip = 0, remove.empty = TRUE, trim = TRUE, ...)  
read_xml(file, skip = 0, remove.empty = TRUE, trim = TRUE, ...)
```

### Arguments

- file** The path to the .html file.
- skip** The number of lines to skip.
- remove.empty** logical. If `TRUE` empty elements in the vector are removed.
- trim** logical. If `TRUE` the leading/trailing white space is removed.
- ...** Other arguments passed to [xml2::read\\_html\(\)](#).

### Value

Returns a character vector.

# Extracting Tables

## html\_table

From [rvest v0.3.2](#)  
by [Hadley Wickham](#)

### Parse An Html Table Into A Data Frame.

Parse an html table into a data frame.

### Usage

```
html_table(x, header = NA, trim = TRUE, fill = FALSE, dec = ".")
```

### Arguments

- x** A node, node set or document.
- header** Use first row as header? If `NA`, will use first row if it consists of `<th>` tags.
- trim** Remove leading and trailing whitespace within each cell?
- fill** If `TRUE`, automatically fill rows with fewer than the maximum number of columns with `NA`s.
- dec** The character used as decimal mark.

### Assumptions

`html_table` currently makes a few assumptions:

- No cells span multiple rows
- Headers are in the first row



### Example

- ```
► url<-"https:
  //finance.yahoo.com/quote/AAPL/history?p=AAPL"
► apple_finance<- read_html(url)
► tableaslist<-html_table(apple_finance)
► table<-as.data.frame(tableaslist)
```

# Extracting Tables

## Example

| Filter |              |               |               |  |
|--------|--------------|---------------|---------------|--|
|        | Date         | Open          | High          |  |
| 1      | Aug 11, 2020 | 447.88        | 449.93        |  |
| 2      | Aug 10, 2020 | 450.40        | 455.10        |  |
| 3      | Aug 07, 2020 | 452.82        | 454.70        |  |
| 4      | Aug 07, 2020 | 0.82 Dividend | 0.82 Dividend |  |
| 5      | Aug 06, 2020 | 441.62        | 457.65        |  |
| 6      | Aug 05, 2020 | 437.51        | 441.57        |  |
| 7      | Aug 04, 2020 | 436.53        | 443.16        |  |
| 8      | Aug 03, 2020 | 432.80        | 446.55        |  |
| 9      | Jul 31, 2020 | 411.54        | 425.66        |  |
| 10     | Jul 30, 2020 | 376.75        | 385.19        |  |
| 11     | Jul 29, 2020 | 375.00        | 380.92        |  |
| 12     | Jul 28, 2020 | 377.47        | 378.20        |  |
| 13     | Jul 27, 2020 | 374.84        | 379 378.20    |  |
| 14     | Jul 24, 2020 | 363.95        | 371.88        |  |
| 15     | Jul 23, 2020 | 387.99        | 388.31        |  |
| 16     | Jul 22, 2020 | 386.77        | 391.90        |  |
| 17     | Jul 21, 2020 | 396.69        | 397.00        |  |
| 18     | Jul 20, 2020 | 385.67        | 394.00        |  |
| 19     | Jul 17, 2020 | 387.05        | 388.50        |  |

# Extracting Text

## html\_text

From [rvest v0.3.6](#)  
by [Hadley Wickham](#)

### Extract Attributes, Text And Tag Name From Html.

Extract attributes, text and tag name from html.

#### Usage

```
html_text(x, trim = FALSE)

html_name(x)

html_children(x)

html_attrs(x)

html_attr(x, name, default = NA_character_)
```

#### Arguments

- x** A document, node, or node set.
- trim** If `TRUE` will trim leading and trailing spaces.
- name** Name of attribute to retrieve.
- default** A string used as a default value when the attribute does not exist in every node.

#### Value

`html_attr`, `html_tag` and `html_text`, a character vector; `html_attrs`, a list.

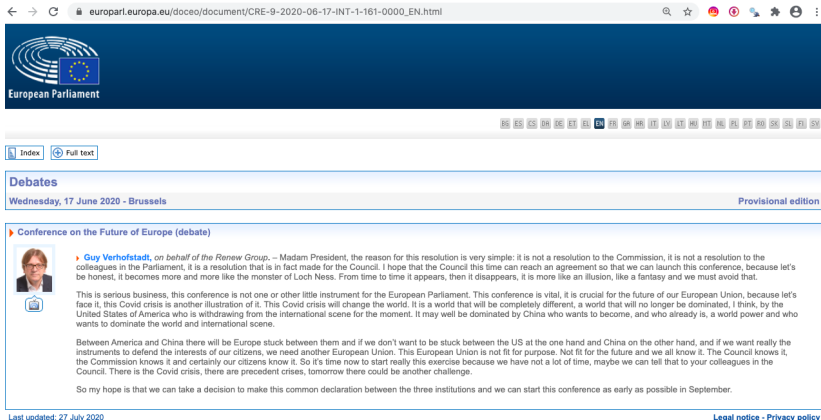
#### Examples

```
# NOT RUN {
movie <- read_html("https://en.wikipedia.org/wiki/The_Lego_Movie")
cast <- html_nodes(movie, "tr:nth-child(8) .plainlist a")
html_text(cast)
```



## Extracting Text

### Example



# Extracting Text

## Example

← → ↺ europa.eu/document/CRE-9-2020-06-17-INT-1-161-0000\_EN.html 🔍 ☆ 🌐 📄 🏠 👤 ⋮

  
European Parliament

BR ES CS DA DE ET EL EN FR GR HR IT LV LT HU MT NL PL PT RO SK SL FI SV

 Index  Full text

### Debates

Wednesday, 17 June 2020 - Brussels Provisional edition

#### Conference on the Future of Europe (debate)



**Guy Verhofstadt**, on behalf of the Renew Group. – Madam President, the reason for this resolution is very simple: it is not a resolution to the Commission, it is not a resolution to the colleagues in the Parliament, it is a resolution that is in fact made for the Council. I hope that the Council this time can reach an agreement so that we can launch this conference, because let's be honest, it becomes more and more like the monster of Loch Ness. From time to time it appears, then it disappears, it is more like an illusion, like a fantasy and we must avoid that.

Let us be serious business. This conference is not one or other like instrument for the European Parliament. This conference is what, it is crucial for the future of our European Union, because let's see it. This Council crisis is another indication of it. This Council crisis will change the world. It is a world that will be completely different, a world that will no longer be dominated, I think, by the United States of America who is withdrawing from the international scene for the moment. It may well be dominated by China who wants to become, and who already is, a world power and who wants to dominate the world and international system.

Between America and China there will be Europe stuck between them and if we don't want to be stuck between the US at the one hand and China on the other hand, and if we want really the instruments to defend the interests of our citizens, we need another European Union. This European Union is not fit for purpose. Not fit for the future and we all know it. The Council knows it, the Commission knows it and certainly our citizens know it. So it's time now to start really this exercise because we have not a lot of time, maybe we can tell that to your colleagues in the Council. There is the Covid crisis, there are precedent crises, tomorrow there could be another challenge.

So my hope is that we can take a decision to make this common declaration between the three institutions and we can start this conference as early as possible in September.

Last updated: 27 July 2020 Legal notice Privacy policy

.contents Clear (4) Toggle Position XPath ? X

# Extracting Text

## Example

- ▶ `url<-"https://www.europarl.europa.eu/doceo/document/CRE-9-2020-06-17-INT-1-161-0000_EN.html"`
- ▶ `v_speech1<-read_html(url)`
- ▶ `text <- v_speech1 %>%  
 html_nodes(".contents") %>%  
 html_text() %>%  
 as.data.frame()`

# Extracting Text

## Example

1                   Guy Verhofstadt,    on behalf of the Renew Group. – Madam President, the reason for this resolution is very simple: it is not a resolution to the Commission, it is not a resolution to the colleagues in the Parliament, it is a resolution that is in fact made for the Council. I hope that the Council this time can reach an agreement so that we can launch this conference, because let's be honest, it becomes more and more like the monster of Loch Ness. From time to time it appears, then it disappears, it is more like an illusion, like a fantasy and we must avoid that.

2                   This is serious business, this conference is not one or other little instrument for the European Parliament. This conference is vital, it is crucial for the future of our European Union, because let's face it, this Covid crisis is another illustration of it. This Covid crisis will change the world. It is a world that will be completely different, a world that will no longer be dominated, I think, by the United States of America who is withdrawing from the international scene for the moment. It may well be dominated by China who wants to become, and who already is, a world power and who wants to dominate the world and international scene.

3 Between America and China there will be Europe stuck between them and if we don't want to be stuck between the US at the one hand and China on the other hand, and if we want really the instruments to defend the interests of our citizens, we need another European Union. This European Union is not fit for purpose. Not fit for the future and we all know it. The Council knows it, the Commission knows it and certainly our citizens know it. So it's time now to start really this exercise because we have not a lot of time, maybe we can tell that to your colleagues in the Council. There here is the Covid crisis, there are precedent crises, tomorrow there could be another challenge.

# Extracting Links

- Use `html_attr("href")` after identifying the relevant node

```
html_attrs(x)

html_attr(x, name, default = NA_character_)
```

## Arguments

- x** A document, node, or node set.
- trim** If `TRUE` will trim leading and trailing spaces.
- name** Name of attribute to retrieve.
- default** A string used as a default value when the attribute does not exist in every node.

## Value

`html_attr`, `html_tag` and `html_text`, a character vector; `html_attrs`, a list.

## Examples


```
# NOT RUN {
movie <- read_html("https://en.wikipedia.org/wiki/The_Lego_Movie")
cast <- html_nodes(movie, "tr:nth-child(8) .plainlist a")
html_text(cast)
html_name(cast)
html_attrs(cast)
html_attr(cast, "href")
# }
```

# Extracting Links

## Example

europarl.europa.eu/meps/en/97058/GUY\_VERHOFSTADT/main-activities/plenary-speeches#detailedcardmep

European Parliament



**Guy VERHOFSTADT** ♦♦

Renew Europe Group  
Member

Belgium - Open Vlaamse Liberalen en Democraten (Belgium)

Date of birth: 11-04-1963, Dendermonde

### Contributions to plenary debates

Speeches made during the plenary session and written declarations relating to plenary debates. Rules 154a (20) and 173(3)

Conclusions of the extraordinary European Council meeting of 17-21 July 2020 (continuation of debate)  
20-07-2020 - PS\_CNG-PRCV(202007-02)(140-0000)

Conference on the Future of Europe (debate)  
17-08-2020 - PS\_CNG-PRCV(202008-17)(1-0000)

Conclusions of the extraordinary European Council meeting of 23 April 2020 - New MFF, own resources and Recovery plan (debate)  
13-05-2020 - PS\_CNG-RECV(202005-13)(179-0000)

EU coordinated action to combat the COVID-19 pandemic and its consequences (continuation of debate)  
19-04-2020 - PS\_CNG-RECV(202004-19)(181-0000)

Withdrawal Agreement of the United Kingdom of Great Britain and Northern Ireland from the European Union and the European Atomic Energy Community (debate)  
20-01-2020 - PS\_CNG-RECV(202001-20)(189-0000)

European Parliament's position on the Conference on the Future of Europe (debate)  
19-07-2019 - PS\_CNG-RECV(201907-19)(029-0000)

European Parliament's position on the Conference on the Future of Europe (debate)  
15-01-2019 - PS\_CNG-RECV(2019-01-15)(029-0000)

Home

Main parliamentary activities

- Contributions to plenary debates
- Reports - as rapporteur
- Motions for resolutions

Other parliamentary activities

- Curriculum vitae
- Declarations
- Assistants
- History of parliamentary service

# Extracting Links

## Example

The image shows a profile page for Guy Verhofstadt, a member of the Renew Europe Group. Below the profile information, there is a section titled "Contributions to plenary debates" which lists several speeches. The first entry is highlighted in green and contains a link to a document. The second entry is highlighted in yellow. The third entry is also highlighted in yellow. The fourth entry is highlighted in yellow. The fifth entry is highlighted in yellow. The sixth entry is highlighted in yellow. The seventh entry is highlighted in yellow. The eighth entry is highlighted in yellow. The ninth entry is highlighted in yellow. The tenth entry is highlighted in yellow. The eleventh entry is highlighted in yellow. The twelfth entry is highlighted in yellow. The thirteenth entry is highlighted in yellow. The fourteenth entry is highlighted in yellow. The fifteenth entry is highlighted in yellow. The sixteenth entry is highlighted in yellow. The seventeenth entry is highlighted in yellow. The eighteenth entry is highlighted in yellow. The nineteenth entry is highlighted in yellow. The twentieth entry is highlighted in yellow. The twenty-first entry is highlighted in yellow. The twenty-second entry is highlighted in yellow. The twenty-third entry is highlighted in yellow. The twenty-fourth entry is highlighted in yellow. The twenty-fifth entry is highlighted in yellow. The twenty-sixth entry is highlighted in yellow. The twenty-seventh entry is highlighted in yellow. The twenty-eighth entry is highlighted in yellow. The twenty-ninth entry is highlighted in yellow. The thirtieth entry is highlighted in yellow. The thirty-first entry is highlighted in yellow. The thirty-second entry is highlighted in yellow. The thirty-third entry is highlighted in yellow. The thirty-fourth entry is highlighted in yellow. The thirty-fifth entry is highlighted in yellow. The thirty-sixth entry is highlighted in yellow. The thirty-seventh entry is highlighted in yellow. The thirty-eighth entry is highlighted in yellow. The thirty-ninth entry is highlighted in yellow. The fortieth entry is highlighted in yellow. The forty-first entry is highlighted in yellow. The forty-second entry is highlighted in yellow. The forty-third entry is highlighted in yellow. The forty-fourth entry is highlighted in yellow. The forty-fifth entry is highlighted in yellow. The forty-sixth entry is highlighted in yellow. The forty-seventh entry is highlighted in yellow. The forty-eighth entry is highlighted in yellow. The forty-ninth entry is highlighted in yellow. The fiftieth entry is highlighted in yellow. The fifty-first entry is highlighted in yellow. The fifty-second entry is highlighted in yellow. The fifty-third entry is highlighted in yellow. The fifty-fourth entry is highlighted in yellow. The fifty-fifth entry is highlighted in yellow. The fifty-sixth entry is highlighted in yellow. The fifty-seventh entry is highlighted in yellow. The fifty-eighth entry is highlighted in yellow. The fifty-ninth entry is highlighted in yellow. The sixtieth entry is highlighted in yellow. The sixty-first entry is highlighted in yellow. The sixty-second entry is highlighted in yellow. The sixty-third entry is highlighted in yellow. The sixty-fourth entry is highlighted in yellow. The sixty-fifth entry is highlighted in yellow. The sixty-sixth entry is highlighted in yellow. The sixty-seventh entry is highlighted in yellow. The sixty-eighth entry is highlighted in yellow. The sixty-ninth entry is highlighted in yellow. The seventieth entry is highlighted in yellow. The seventy-first entry is highlighted in yellow. The seventy-second entry is highlighted in yellow. The seventy-third entry is highlighted in yellow. The seventy-fourth entry is highlighted in yellow. The seventy-fifth entry is highlighted in yellow. The seventy-sixth entry is highlighted in yellow. The seventy-seventh entry is highlighted in yellow. The seventy-eighth entry is highlighted in yellow. The seventy-ninth entry is highlighted in yellow. The eightieth entry is highlighted in yellow. The eighty-first entry is highlighted in yellow. The eighty-second entry is highlighted in yellow. The eighty-third entry is highlighted in yellow. The eighty-fourth entry is highlighted in yellow. The eighty-fifth entry is highlighted in yellow. The eighty-sixth entry is highlighted in yellow. The eighty-seventh entry is highlighted in yellow. The eighty-eighth entry is highlighted in yellow. The eighty-ninth entry is highlighted in yellow. The ninetieth entry is highlighted in yellow. The ninety-first entry is highlighted in yellow. The ninety-second entry is highlighted in yellow. The ninety-third entry is highlighted in yellow. The ninety-fourth entry is highlighted in yellow. The ninety-fifth entry is highlighted in yellow. The ninety-sixth entry is highlighted in yellow. The ninety-seventh entry is highlighted in yellow. The ninety-eighth entry is highlighted in yellow. The ninety-ninth entry is highlighted in yellow. The hundredth entry is highlighted in yellow.

**Contributions to plenary debates**

Speeches made during the plenary session and written declarations relating to plenary debates. Rules [Rule 204](#) and [175\(1\)](#)

**Conclusions of the European Council meeting of 15 and 16 October 2020, in particular the negotiations of the future relations with the UK (debate)** [\[en\]](#)  
21-10-2020 - PB\_CRE-PROV(2020)10-21(3-045-0000)

**State of the Union (topical debate) (debate)** [\[en\]](#)  
16-09-2020 - PB\_CRE-PROV(2020)08-16(3-044-0000)

**Conclusions of the extraordinary European Council meeting of 17-21 July 2020 (continuation of debate)**  
25-07-2020 - PB\_CRE-PROV(2020)07-25(1-063-0000)

**Conference on the Future of Europe (debate)**  
17-06-2020 - PB\_CRE-REV(2020)06-17(1-161-0000)

**Conclusions of the extraordinary European Council meeting of 23 April 2020 - NE-MPF, own resources and Recovery plan (debate)** [\[en\]](#)

**Home**

**Main parliamentary activities**

Contributions to plenary debates

Reports - as rapporteur

Motions for resolutions

**Other parliamentary activities**

Curriculum vitae

Declarations

Assistants

# Extracting Links

## Example

- ▶ `url<-"https://www.europarl.europa.eu/meps/en/97058/GUY_VERHOFSTADT/main-activities/plenary-speeches#detailedcardmep"`
- ▶ `verhof_speech <- read_html(url)`
- ▶ `speech_links <- verhof_speech %>%  
 html_nodes(".erpl-search-results-list-expandable-block  
a") %>%  
 html_attr("href") %>%  
 as.data.frame()`



# Extracting Links

## Example

|    |                                                                                                                     |
|----|---------------------------------------------------------------------------------------------------------------------|
|    | .                                                                                                                   |
| 1  | <a href="https://www.europarl.europa.eu/doceo/document/C...">https://www.europarl.europa.eu/doceo/document/C...</a> |
| 2  | <a href="https://www.europarl.europa.eu/doceo/document/C...">https://www.europarl.europa.eu/doceo/document/C...</a> |
| 3  | <a href="https://www.europarl.europa.eu/doceo/document/C...">https://www.europarl.europa.eu/doceo/document/C...</a> |
| 4  | <a href="https://www.europarl.europa.eu/doceo/document/C...">https://www.europarl.europa.eu/doceo/document/C...</a> |
| 5  | <a href="https://www.europarl.europa.eu/doceo/document/C...">https://www.europarl.europa.eu/doceo/document/C...</a> |
| 6  | <a href="https://www.europarl.europa.eu/doceo/document/C...">https://www.europarl.europa.eu/doceo/document/C...</a> |
| 7  | <a href="https://www.europarl.europa.eu/doceo/document/C...">https://www.europarl.europa.eu/doceo/document/C...</a> |
| 8  | <a href="https://www.europarl.europa.eu/doceo/document/C...">https://www.europarl.europa.eu/doceo/document/C...</a> |
| 9  | <a href="https://www.europarl.europa.eu/doceo/document/C...">https://www.europarl.europa.eu/doceo/document/C...</a> |
| 10 | <a href="https://www.europarl.europa.eu/doceo/document/C...">https://www.europarl.europa.eu/doceo/document/C...</a> |

## Looping over list of URLs

- ▶ This is useful in situations where urls have patterns
- ▶ E.g. In EP website, MEPs urls are identical barring from MEP number and name.
- ▶ A scraper can exploit this and loop over a series of webpages with identical content
- ▶ This 'simply' requires creating a list of univocal url codes, pasting those to the unchanging - generic - url and running a standard for loop
- ▶ The web scraper we built will be inside the for loop

# Looping over list of URLs

## Example

► Consider the following urls:

1. `https://www.europarl.europa.eu/meps/en/97058/GUY_VERHOFSTADT/main-activities/plenary-speeches#detailedcardmep/`
2. `https://www.europarl.europa.eu/meps/en/197395/ALICE_KUHNKE/main-activities/plenary-speeches#detailedcardmep`
3. `https://www.europarl.europa.eu/meps/en/124846/PINA_PICIERNO/main-activities/plenary-speeches#detailedcardmep`

# Looping over list of URLs

## Example

```
#### Loop through URLs

MEPcodes <- c("97058/GUY_VERHOFSTADT", "197395/ALICE_KUHNKE", "124846/PINA_PICIERNO")

urls <- paste0("https://www.europarl.europa.eu/meps/en/", MEPcodes, "/main-activities/plenary-speeches#detailedcardmep" )



catcherlist<-list()

for (i in urls) {
  page <- read_html(i)
  Name <- page %>% html_nodes("#presentationmep .erpl_title-h1") %>% html_text() %>% as.character()
  Title <- page %>% html_nodes(".erpl_search-results-list-expandable-block .t-item") %>% html_text() %>% as.character()
  Link <- page %>% html_nodes(".erpl_search-results-list-expandable-block a") %>% html_attr("href") %>% as.character()
  temp <- list(Name, Title, Link)
  catcherlist <- rbind(catcherlist,temp)
}

df<-as.data.frame(catcherlist)
```

# Looping over list of links

## Example

|   Filter |                                                         |                                                    |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------|----------------------------------------------------|
|                                                                                                                                                                            | title                                                   | url                                                |
| 1                                                                                                                                                                          | Conclusions of the extraordinary European Council m...  | https://www.europarl.europa.eu/doceo/document/C... |
| 2                                                                                                                                                                          | Conference on the Future of Europe (debate)             | https://www.europarl.europa.eu/doceo/document/C... |
| 3                                                                                                                                                                          | Conclusions of the extraordinary European Council m...  | https://www.europarl.europa.eu/doceo/document/C... |
| 4                                                                                                                                                                          | EU coordinated action to combat the COVID-19 pand...    | https://www.europarl.europa.eu/doceo/document/C... |
| 5                                                                                                                                                                          | Withdrawal Agreement of the United Kingdom of Grea...   | https://www.europarl.europa.eu/doceo/document/C... |
| 6                                                                                                                                                                          | European Parliament's position on the Conference on ... | https://www.europarl.europa.eu/doceo/document/C... |
| 7                                                                                                                                                                          | European Parliament's position on the Conference on ... | https://www.europarl.europa.eu/doceo/document/C... |
| 8                                                                                                                                                                          | Implementing and monitoring the provisions on citize... | https://www.europarl.europa.eu/doceo/document/C... |
| 9                                                                                                                                                                          | Conclusions of the European Council meeting of 12 a...  | https://www.europarl.europa.eu/doceo/document/C... |
| 10                                                                                                                                                                         | Conclusions of the European Council meeting of 17 a...  | https://www.europarl.europa.eu/doceo/document/C... |

# Looping over list of links

## Example

```
##follow links

#Setup empty data frame

catcher_text <- data.frame(Name=character(),Text=character())

for (i in my_data$links) {

  page <- read_html(i)
  Name<-page %>% html_nodes(".doc_subtitle_level1_bis .bold") %>% html_text() %>% as.character()
  Text<-page %>% html_nodes(".contents") %>% html_text() %>% as.character()
  temp_text <- data.frame(Name, Text) #fill temporary repository
  catcher_text <- rbind(catcher_text,temp_text) #convert into dataframe
}
```

# Clicking Buttons

E.g. “Load More”

The screenshot shows the European Parliament website. The main content area lists several debates, each with a title and a date. The 'Load more' button is highlighted with a red box. The right sidebar contains a navigation menu with links to Home, Main parliamentary activities, Contributions to plenary debates, Reports - as rapporteur, Motions for resolutions, Other parliamentary activities, Curriculum vitae, Declarations, Assistants, and History of parliamentary service. The bottom of the page shows the URL: https://www.europarl.europa.eu/meps/en/97058/GUY\_VERHOFFSTADT/main-activities/plenary-speeches# and a search bar.

European Parliament

Conclusions of the Extraordinary European Council meeting of 25 April 2020 - New MFF, own resources and Recovery plan (debate)  
19-06-2020 - P9\_CRC-REV(2020)05-13(1-079-0000)

EU coordinated action to combat the COVID-19 pandemic and its consequences (continuation of debate)  
16-04-2020 - P9\_CRC-REV(2020)04-16(1-067-0000)

Withdrawal Agreement of the United Kingdom of Great Britain and Northern Ireland from the European Union and the European Atomic Energy Community (debate)  
29-01-2020 - P9\_CRC-REV(2020)01-09(1-065-0000)

European Parliament's position on the Conference on the Future of Europe (debate)  
15-01-2020 - P9\_CRC-REV(2020)01-15(3-025-0000)

European Parliament's position on the Conference on the Future of Europe (debate)  
15-01-2020 - P9\_CRC-REV(2020)01-15(3-029-0000)

Implementing and monitoring the provisions on citizens' rights in the Withdrawal Agreement (debate)  
14-01-2020 - P9\_CRC-REV(2020)01-14(2-086-0000)

Conclusions of the European Council meeting of 12 and 13 December 2019 (debate)  
16-12-2019 - P9\_CRC-REV(2019)12-16(3-055-0000)

Conclusions of the European Council meeting of 17 and 18 October 2019 (debate)  
20-10-2019 - P9\_CRC-REV(2019)10-20(2-025-0000)

Load more

https://www.europarl.europa.eu/meps/en/97058/GUY\_VERHOFFSTADT/main-activities/plenary-speeches#

.btn-default Clear (1) Toggle Position XPath ? X

## RSelenium

- ▶ *Dynamic* interaction (filling fields, clicking buttons) with the website is not possible through rvest
- ▶ You need RSelenium for this: but **outside of the scopes of this course**
- ▶ RSelenium requires a bit of environment preparation:
  - ▶ It requires selenium initiation, by opening a browser session (not simply the specification of a url)
  - ▶ Only after setting up the 'Remote Server/Driver' it can navigate the specific web page ... v. time consuming
- ▶ The core functions to perform a button click are RSelenium's `$findElement` and `$clickElement`
- ▶ If you want to know more: [http://joshuamccrain.com/tutorials/web\\_scraping\\_R\\_selenium.html](http://joshuamccrain.com/tutorials/web_scraping_R_selenium.html)



## What we have learnt today ...

- ▶ SelectorGadget can make the first step of scraping (web-page analysis) so much easier
- ▶ There are 3 core strategies in scraping
  - ▶ Regular expression (worst case)
  - ▶ Node queries (typical case)
  - ▶ APIs (gold standard but rare)
- ▶ The core functions in R to perform **node query** scraping - using the package rvest
- ▶ Awareness of RSelenium and the problems in scraping dynamic pages
- ▶ Next week: more on scraping via APIs: we're going to learn how to scrape Twitter data