

# Quantitative Content Analysis: Lecture 11

Matthias Haber

26 April 2017

# Today's outline

- Text data sources and scripts
- Wordfish

# Online text data sources

- **web pages** (e.g. <http://example.com>)
- **web formats** (XML, HTML, JSON, ...)
- **web frameworks** (HTTP, URL, APIs, ...)
- **social media** (Twitter, Facebook, LinkedIn, Snapchat, Tumblr, ...)
- **data in the web** (speeches, laws, policy reports, news, ...)
- **web data** (page views, page ranks, IP-addresses, ...)

# The Problems

phase	problems	examples
<b>download</b>	protocols procedures	HTTP, HTTPS, POST, GET, ... cookies, authentication, forms, ...
<b>extraction</b>	parsing extraction cleansing	translating HTML (XML, JSON, ...) into R getting the relevant parts cleaning up, restructure, combine

# Before scraping, do some googling!

- If the resource is well-known, someone else has probably built a tool which solves the problem for you.
- ropensci has a ton of R packages providing easy-to-use interfaces to open data.
- The Web Technologies and Services CRAN Task View is a great overview of various tools for working with data that lives on the web in R.

# Example

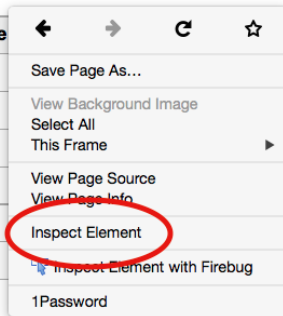
# Inspecting elements

## Simple table [\[edit\]](#)

The following illustrates a simple table with three columns and six rows. display the column names. This is traditionally called a "header row".

**Age table**

First name	Last name	Age
Bielat	Adamczak	24
Blaszczyk	Kostrzewski	25
Olatunkboh	Chijiaku	22
Adrienne	Anthoula	22
Axelia	Athanasios	22
Jon-Kabat	Zinn	22



# Hover to find desired elements

Simple table [\[edit\]](#)

The following illustrates a simple table with three columns and six rows. The first row is not counted, because it is only used to display the column names. This is traditionally called a "header row".

First name	Last name	Age
Bielat	Adamczak	24
Blaszczyk	Kostrzewski	25
Olatunkboh	Chijiaku	22
Adrienne	Anthoula	22
Axelia	Athanasios	22
Jon-Kabat	Zinn	22

Multi-dimensional table [\[edit\]](#)

The concept of dimension is also a part of basic terminology.<sup>[7]</sup> Any "simple" table can

The screenshot shows a web browser interface. On the left, there's a sidebar with language options: Ruina Simi, Simple English, Slovenčina, Svenska, தமிழ், ไทย, and Українська. Below this is a section for '中文' with an 'Edit links' button. The main content area displays a 'Simple table' with a caption 'table.wikitable 253 x 245' and a description. Below the description is a table with 3 columns and 7 rows. The first row is a header row. Below the table is a 'Multi-dimensional table' section with a description. At the bottom, a developer console is open, showing the HTML structure of the table and the CSS rules for the 'table.wikitable' class.

```
tbody > tr > td > table.wikitable
```

```
<table class="wikitable" style="text-align:center">
  <caption></caption>
  <tbody>
    <tr></tr>
    <tr>
      <td></td>
      <td></td>
      <td></td>
    </tr>
    <tr></tr>
    <tr></tr>
    <tr></tr>
  </tbody>
</table>
```

```
Rules
Computed
Fonts
Box Model
```

```
element {
  text-align: center;
}

table.wikitable {
  margin: 1em 8px;
  background-color: #F0F0F0;
  border: 1px solid #AAA;
  border-collapse: collapse;
  color: #000;
}

table {
```



# Wikitable

```
library(rvest)
src <- html("http://en.wikipedia.org/wiki/Table_(information)")
node <- html_node(src, css = ".wikitable")
```

- ".wikitable" is a CSS selector which says: "grab nodes (aka elements) with a class of wikitable".
- `html_table()` converts a single `<table>` node to a data frame.

```
html_table(node)
##   First name    Last name Age
## 1      Tinu     Elejogun  14
## 2 Blaszczyk Kostrzewski  25
## 3      Lily    McGarrett  16
## 4 Olatunkboh   Chijiaku  22
## 5   Adrienne   Anthoula  22
## 6     Axelia  Athanasios  22
## 7   Jon-Kabat      Zinn   22
```

# Pipeable!

```
html("http://en.wikipedia.org/wiki/Table_(information)") %>%  
  html_node(".wikitable") %>% html_table()  
##   First name   Last name Age  
## 1      Tinu    Elejogun  14  
## 2 Blaszczyk Kostrzewski  25  
## 3      Lily    McGarrett  16  
## 4 Olatunkboh  Chijiaku   22  
## 5   Adrienne   Anthoula  22  
## 6     Axelia  Athanasios  22  
## 7   Jon-Kabat      Zinn   22
```

# Rvest

rvest is a nice R package for web-scraping by (you guessed it) Hadley Wickham.

```
library(dplyr)
library(rvest)
library(magrittr)
# First, grab the page source
html("http://en.wikipedia.org/wiki/Table_(information)") %>%
  # then extract the first node with class of wikitable
  html_node(".wikitable") %>%
  # then convert the HTML table into a data frame
  html_table()
##   First name   Last name Age
## 1      Tinu    Elejogun  14
## 2 Blaszczyk Kostrzewski  25
## 3      Lily    McGarrett  16
## 4 Olatunkboh  Chijiaku   22
## 5   Adrienne  Anthoula   22
## 6     Axelia  Athanasios  22
## 7   Jon-Kabat      Zinn   22
```

# HTML / XML with rvest

```
rpacak_html <-  
"http://cran.r-project.org/web/packages" %>%  
html()  
rpacak_html %>% class()  
## [1] "xml_document" "xml_node"
```

# HTML / XML with rvest (II)

```
rpacak_html %>% xml_structure(indent = 2)
## <html [xmlns]>
##   <head>
##     <title>
##       {text}
##     <link [rel, type, href]>
##     <meta [http-equiv, content]>
##   <body>
##     {text}
##     <h1>
##       {text}
##       {text}
##     <h3 [id]>
##       {text}
##       {text}
##     <p>
##       {text}
##       {text}
##     <p>
##       {text}
```

## HTML / XML with rvest (III)

```
rpacak_html %>% html_text() %>% cat()
## CRAN - Contributed Packages
## Contributed Packages
##
## Available Packages
## Currently, the CRAN package repository features 10472 available package
## Table of available packages, sorted by date of publication
## Table of available packages, sorted by name
## Installation of Packages
##
## Please type
##   help("INSTALL")
## or
##   help("install.packages")
## in R for information on how to install packages from this
## repository. The manual
##
## R Installation and Administration
## (also contained in the R base sources)
## explains the process in detail.
```

# Technologies and Packages

- Regular Expressions / String Handling
  - **stringr**, stringi
- HTML / XML / XPath / CSS Selectors
  - **rvest**, xml2, XML
- JSON
  - **jsonlite**, RJSONIO, rjson
- HTTP / HTTPS
  - **httr**, curl, Rcurl
- Javascript / Browser Automation
  - **RSelenium**
- URL
  - **urltools**

# Readings

- Basics on HTML, XML, JSON, HTTP, RegEx, XPath
  - Munzert et al. (2014): Automated Data Collection with R. Wiley.  
<http://www.r-datacollection.com/>
- curl / libcurl
  - [http://curl.haxx.se/libcurl/c/curl\\_easy\\_setopt.html](http://curl.haxx.se/libcurl/c/curl_easy_setopt.html)
- CSS Selectors
  - W3Schools:  
[http://www.w3schools.com/cssref/css\\_selectors.asp](http://www.w3schools.com/cssref/css_selectors.asp)
- Packages: httr, rvest, jsonlite, xml2, curl
  - Readmes, demos and vignettes accompanying the packages
- Packages: RCurl and XML
  - Munzert et al. (2014): Automated Data Collection with R. Wiley. Nolan and Temple-Lang (2013): XML and Web Technologies for Data Science with R. Springer



# Next Session

- Topic modelling