A Fast-Track-Overview on Web Scraping with R

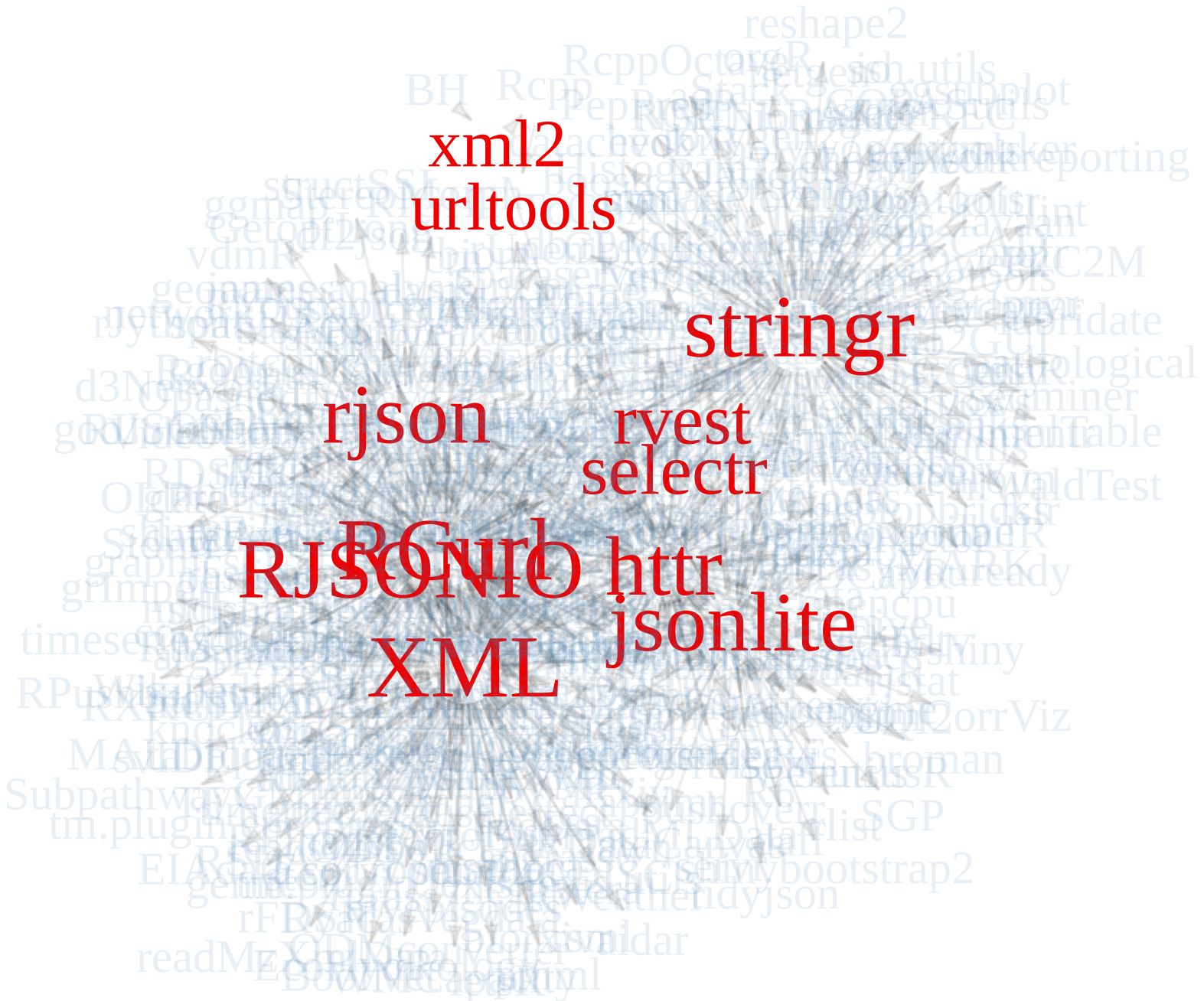
UseR! 2015

Peter Meißner
Comparative Parliamentary Politics Working Group
University of Konstanz

`https://github.com/petermeissner`
`http://pmeissner.com`
`http://www.r-datacollection.com/`

presented: 2015-07-01 / last update: 2015-06-30

Introduction



Introduction

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

Conventions

All code examples assume . . .

- `dplyr`
- `magrittr`

. . . to be loaded via . . .

```
library(dplyr)
library(magrittr)
```

. . . while all other package dependencies will be made explicit on an example by example base.

Reading Text from the Web

```
news <-
  "http://cran.r-project.org/web/packages/base64enc/NEWS"  %>%
  readLines(url)
```

```
news %>% extract(1:10) %>%  cat(sep="\n")
```

```
## 0.1-2    2014-06-26
##    o bugfix: encoding content of more than 65536 bytes without
##  linebreaks produced padding characters between chunks because
##  chunk size was not divisible by three.
##
##
## 0.1-1    2012-11-05
##    o fix a bug in base64decode where output is a file name
##
##    o add base64decode(file=...) as a (non-leaking) shorthand for
```

Extracting Information from Text

. . . with base R

```
news  %>%
  substring(7, 16)  %>%
  grep("\\d{4}.\\d{1,2}.\\d{1,2}", ., value=T)
```

```
## [1] "2014-06-26" "2012-11-05" "2012-09-07"
```

Extracting Information from Text

... with stringr

```
library(stringr)
news  %>%
  str_extract("\\d{4}.\\d{1,2}.\\d{1,2}")
```

```
##  [1] "2014-06-26" NA           NA           NA
##  [5] NA           NA           "2012-11-05" NA
##  [9] NA           NA           NA           NA
## [13] NA           "2012-09-07" NA
```

HTML / XML

... with rvest

```
library(rvest)


rpack_html <-
  "http://cran.r-project.org/web/packages" %>%
  html()
```

```
rpack_html %>% class()
```

```
## [1] "HTMLInternalDocument" "HTMLInternalDocument"
## [3] "XMLInternalDocument"  "XMLAbstractDocument"
```

HTML / XML

. . . with rvest

```
rpack_html %>% xml_structure(indent = 2)
```

```
##   {DTD}
##   <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}
##         <a [href]> {text}
##         {text}
```

HTML / XML

... with rvest

```
rpack_html %>% html_text()  %>% cat()
```

```
## CRAN - Contributed Packages
## Contributed Packages
##
## Available Packages
##  Currently, the CRAN package repository features 6803 available pack
##   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)
```

Extraction from HTML / XML

... with rvest and XPath

```
rpack_html %>%
  html_node(xpath="//p/a[contains(@href, 'views')]/..")
```

```
## <p>
## <a href="../views/">CRAN Task Views</a>
## allow you to browse packages by topic and provide tools to
## automatically install all packages for special areas of
## interest.
## Currently, 33 views are available.
## </p>
```

Extraction from HTML / XML

. . . with rvest and XPath

```
rpack_html %>%
  html_nodes(xpath="//a") %>%
  html_attr("href")  %>%
  extract(1:6)
```

```
## [1] "available_packages_by_date.html"
## [2] "available_packages_by_name.html"
## [3] "../../manuals.html#R-admin"
## [4] "../views/"
## [5] "http://www.debian.org/"
## [6] "http://www.fedoraproject.org/"
```

Extraction from HTML / XML

... with rvest convenience functions

```
"http://cran.r-project.org/web/packages/multiplex/index.html"  %>%
  html() %>%
  html_table() %>%
  extract2(1) %>%
  filter(X1 %in% c("Version:", "Published:", "Author:"))
```

```
##           X1                          X2
## 1   Version:                         1.6
## 2 Published:                  2015-05-19
## 3    Author: J. Antonio Rivero Ostoic
```

JSON

```
"https://api.github.com/users/daroczig/repos" %>%
  readLines(warn=F) %>%
  substring(1,300) %>%
  str_wrap(60) %>%
  cat()
```

```
## [{"id":
## 12325008,"name":"AndroidInAppBilling","full_name":"daroczig/
## AndroidInAppBilling","owner":{"login":"daroczig","id":
## 495736,"avatar_url":"https://avatars.githubusercontent.com/
## u/495736?v=3","gravatar_id":"","url":"https://
## api.github.com/users/daroczig","html_url":"https://
## github.com/daroczig","f
```

JSON

. . . with jsonlite

```
library(jsonlite)
fromJSON("https://api.github.com/users/daroczig/repos") %>%
  select(language) %>%
  table() %>%
  sort(decreasing=TRUE)
```

```
## .
##          R JavaScript Emacs Lisp      Groff      Jasmin
##         16          4          1          1          1
##       Java        PHP     Python
##          1          1          1
```

HTML forms / HTTP methods

. . . with rvest and httr

```
library(rvest)
library(httr)

text    <-
  "Quirky spud boys can jam after zapping five worthy Polysixes."

mainpage <- html("http://read-able.com")
```

HTML forms / HTTP methods

... with rvest and httr

```
mainpage %>%
  html_nodes(xpath="//form") %>%
  html_attrs()
```

```
## [[1]]
##      method      action
##       "get" "check.php"
##
## [[2]]
##      method      action
##      "post" "check.php"
```

HTML forms / HTTP methods

... with rvest and httr

```
mainpage  %>%
  html_nodes(
    xpath="//form[@method='post']//*[self::textarea or self::input]"
  )
```

```
## [[1]]
## <textarea id="directInput" name="directInput" rows="10" cols="60"></
##
## [[2]]
## <input type="submit" value="Calculate Readability" />
##
## attr(,"class")
## [1] "XMLNodeSet"
```

HTML forms / HTTP methods

. . . with rvest and httr

```
response <-
  POST(
    "http://read-able.com/check.php",
    body=list(directInput = text),
    encode="form"
  )
```

HTML forms / HTTP methods

. . . with rvest and httr

```
response  %>%
  extract2("content") %>%
  rawToChar() %>%
  html() %>%
  html_table() %>%
  extract2(1)
```

```
##                                X1   X2 X3
## 1 Flesch Kincaid Reading Ease 61.3 NA
## 2  Flesch Kincaid Grade Level  7.2 NA
## 3           Gunning Fog Score  4.0 NA
## 4                  SMOG Index  6.0 NA
## 5          Coleman Liau Index 14.2 NA
## 6 Automated Readability Index  7.6 NA
```

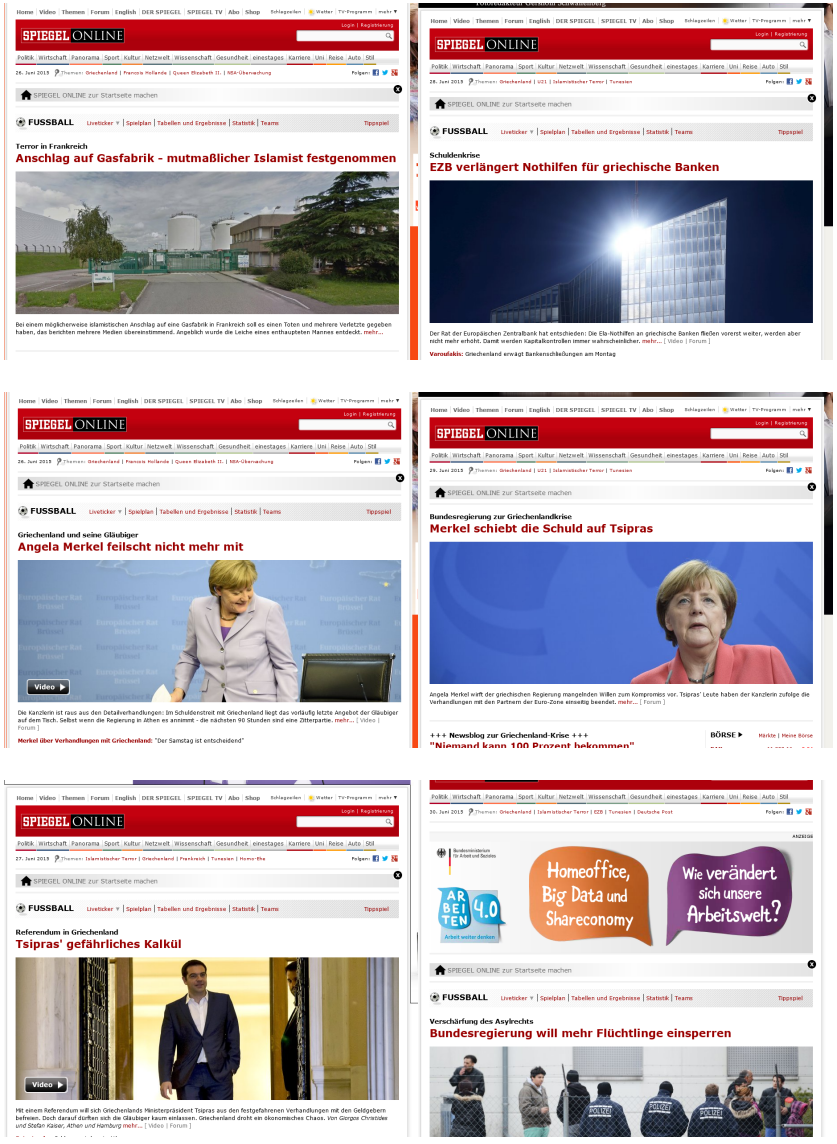
Overcoming the Javascript Barrier

... with RSelenium browser automation

```
library(RSelenium)
checkForServer() # make sure Selenium Server is installed
startServer()
remDr <- remoteDriver() # defaults firefox
dings <- remDr$open(silent=T) # see: ?remoteDriver !!
remDr$navigate("https://spiegel.de")
remDr$screenshot(
  display    = F,
  useViewer = F,
  file       = paste0("spiegel_", Sys.Date(),".png")
)
```

Overcoming the Javascript Barrier

. . . with RSelenium browser automation



Authentication

. . . with httr and httpuv

```
library(httpuv)
library(httr)

twitter_token <- oauth1.0_token(
  oauth_endpoints("twitter"),
  twitter_app <- oauth_app(
    "petermeissneruser2015",
    key = "fP7WB5CcoZNLVQ2Xh8nAdFVAN",
    secret = "PQG1eEJZ65Mb8ANHz8q7yp4MqgAmiAVED90F4ZvQUSTHxiGzPT"
  )
)
```

Authentication

. . . with httr and httpuv

```
req <-
  GET(
    paste0(
      "https://api.twitter.com/1.1/search/tweets.json",
      "?q=%23user2015&result_type=recent&count=100"
    ),
    config(token = twitter_token)
  )
```

Authentication

. . . with httr and httpuv

```
tweets <-
  req  %>%
  content("parsed")  %>%
  extract2("statuses") %>%
  lapply(`[`, "text") %>%
  unlist(use.names=FALSE) %>%
  subset(!grepl("^RT ", tweets)) %>%
  extract(1:15)
```

Authentication

. . . with httr and httpuv

```
tweets %>% substr(1,60) %>% cat(sep="\n")
```

```
## We're almost ready! #useR2015 @RevolutionR http://t.co/Lxov7
## The booth is getting ready for you! See you soon! #useR2015
## jra kzvettetni prblunk: 4+ magyaroszgi ltogat a #user2015 ko
## Congress centre only a stones throw from my room #UseR2015 h
## On my way to #useR2015 Wup, wup!
## My first day at #useR2015 is about to get underway! I hope y
## TIBCO: RT ianmcook: In Denmark at user2015aalborg #useR2015
## And please all Hungarian attendees of #user2015 ping me to g
## @MangoTheCat has arrived! #rstats #user2015 #aalborg http://
## great experience in  #DataMeetsViz, now ready for #useR2015
## Are you looking forward to see this year's t-shirt? Reg. ope
## Trying the Danish hospitality at #useR2015 http://t.co/Qrsvb
## Nice to be in the beautiful #Aalborg for #useR2015. See you
## Time to get some sleep... need to be alert for #useR2015 #rs
## In Denmark at @user2015aalborg #useR2015 with @TIBCO #Spotfi
```

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**

Reads

- **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`
- **CSS Selectors**
  - W3Schools: `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

Conclusion

- **Use Mac or Linux** because there will come the time when special characters punch you in the face on R/Windows and according to R-devel this is unlikely to change any time soon.
- Do not listen to guys saying you should use some other language for Web-Scraping. **If you like R, use R** - for any job.
- Use **stringr, rvest and jsonlite** first and the other packages if needed.
- If you want to do scraping learn Regular Expressions, file manipulation with R (file.create(), file.remove(), . . .), XPath or CSS Selectors and a little HTML-XML-JSON.
- Web scraping in R has evolved to a convenience state but still is a moving target within a year there might be even more powerful and/or more convenience packages.
- Before scraping data: (1) Watch for the download button; (2) Have a look at CRAN Web Technologies Task View; Look for an API or if maybe someone else has done it before. k

Thanks

```
thanks()
```

```
## Alex Couture-Beil, Duncan Temple Lang, Duncan Temple Lang,
## Duncan Temple Lang, Duncan Temple Lang, Hadley Wickham,
## Hadley Wickham, Hadley Wickham, Hadley Wickham, Ian
## Bicking, Inc., Jeroen Ooms, Jeroen Ooms, John Harrison,
## Lloyd Hilaiel, Mark Greenaway, Oliver Keyes, R Foundation,
## RStudio, RStudio, RStudio, RStudio, RStudio, See AUTHORS
## file. igraph author details, Simon Potter, Simon Sapin,
## Simon Urbanek, the CRAN Team
```

. . . and the R Community and all the others.