Errata for ‘Automated Data Collection with R’

Last update: 2017-03-21 18:35:51

```r
library(stringr)
library(RCurl)
library(XML)
library(rvest)
```

**page 2**

*Credit: Suryapratim Sarkar (2015-06-25)*

Wikipedia changed its server communication from HTTP to HTTPS. As a result, the following lines on page 2 return an error:

```r
```


There are at least two solutions to the problem:

1. Use `getURL()` and specify the location of CA signatures (see Section 9.1.7 of our book).

2. Use Hadley Wickham’s `rvest` package, which came out after our book was published. It facilitates scraping with R considerably, in particular in such scenarios. In this specific example, use the following code instead:

```r
library(rvest)  # the new package, version 0.3.0
```

```r
tables <- html_table(heritage_parsed, fill = TRUE)  # html_table() from the rvest package, which replaces readHTMLTable() from the XML package
```

From thereon, the rest of the chapter code should work. If you want to learn more about the rvest package, have a look here. We are planning to cover it extensively in the next edition of our book.

**page 35**

*Credit: Jüri Kuusik (2015-01-22)*

Typo: change “parsed_doc” to “parsed_fortunes$children”

**page 96**

*Credit: Tobias Rosenberger (2017-03-14)*

Typo: `xmlParse("titles.xml")` should read `xmlParse("books.xml").`

**page 136**

*Credit: Laurent Franckx (2015-02-18)*
Due to (supposedly) a bug in the RCurl package (version 1.95-4.5, bug has been reported) the following lines on page 136 give an error:

```r
text <- getCurlHandle(customrequest = "HEAD")
res <- getURL(url = url, curl = handle, header = TRUE)
```

There are two workarounds at the moment.

1. To make the code simply run through you might give up the HTTP HEAD method used in the code above and use `customrequest = "GET"` instead:

```r
require(RCurl)
require(stringr)
url <- "http://www.r-datacollection.com/materials/http/helloworld.html"
res <- getURL(url = url, header = TRUE)
cat(str_split(res, "\r")[[1]])
handle <- getCurlHandle(customrequest = "GET")
res <- getURL(url = url, curl = handle, header = TRUE)
```

2. If you want to have a working example involving HTTP HEAD method you might switch to the httr package like this:

```r
require(httr)
url <- "http://www.r-datacollection.com/materials/http/helloworld.html"
res <- HEAD(url)

res
```

## Date: 2015-06-11 15:03
## Status: 200
## Content-Type: text/html
## <EMPTY BODY>
```

res$request
```

## $handle
## Host: http://www.r-datacollection.com/ <0x104ae8c00>
## $writer
## $method
## [1] "HEAD"
## $opts
## Config:
## List of 8
## $ followlocation:TRUE
## $ maxredirs :10
## $ encoding :"gzip"
## $ useragent :"curl/7.24.0 Rcurl/1.95.4.5 httr/0.6.1"
## $ httpheader :"application/json, text/xml, application/xml, */*"
## ..- attr(*, "names")="Accept"
## $ nobody :TRUE
```

## $ customrequest : "HEAD"
## $body
## NULL
res$headers[1:3]
## $date
## $server
## [1] "Apache"
## $vary
## [1] "Accept-Encoding"

### page 194

*Reported by: Laurent Franckx (2015-05-11)*

The URL on page 194 to the parlgov SQLite database has changed and does not work anymore. The new URL is:


### page 249

*Reported by: Laurent Franckx (2015-06-08)*

The page structure had changed and code did not work anymore.

```r
# define urls
search_url <- "www.biblio.com/search.php?keyisbn=data"
cart_url <- "www.biblio.com/cart.php"

# download and parse page
search_page <- htmlParse(getURL(url = search_url, curl = handle))

# identify form fields
xpathApply(search_page, "//div[@class='row-fixed'][position()<2]/form")
```
## extract book ids

```r
# attr("class")
# [1] "XMLNodeSet"

# extract book ids

xpath <- "//div[@class='row-fixed'][position()<=4]/form/input[@name='bid']/@value"
bids <- unlist(xpathApply(search_page, xpath, as.numeric))
```
bids

## [1] 564631291 579332617 578596993 741931398 585228051 587156394 582476338
## [8] 724420313 724419579 742043581 436242244 587152896 741844001 543068623
## [15] 631091376 631091307 578453413 742060670 741812280 578999967

# add items to shopping cart
for(i in seq_along(bids)) {
  res <- getForm(uri = cart_url,
                 curl = handle,
                 bid = bids[i],
                 add = 1,
                 int = "keyword_search")
}

# inspect shopping cart
cart <- htmlParse(getURL(url=cart_url, curl=handle))
clean <- function(x) str_replace_all(xmlValue(x),"\t|\n"," ")
xpathSApply(cart, "/h3/a", clean)

## [3] "Revolution by George Barna"
## [4] "Preparing Data for Sharing: Guide to Social Science Data Archiving by Data Archiving and Networking Center"
## [5] "Hydraulic Data by Howard Dorsey Coale"
## [8] "The Statistical Analysis of Experimental Data by John Mandel"
## [9] "Cryptonomicon by Stephenson, Neal"
## [10] "Big Data by Cukier, Kenneth,Mayer-Schonberger, Viktor"
## [12] "How to Pass Data Interpretation Tests by Bryan, Mike"
## [13] "Metalworker's Data Book by Hall, Harold"
## [14] "Preparing Data for Sharing: Guide to Social Science Data Archiving by Data Archiving and Networking Center"
## [16] "Oracle Data Dictionary Pocket Reference by David C. Kreines"
## [17] "The Statistical Analysis of Experimental Data by John Mandel"
## [18] "Cryptonomicon by Stephenson, Neal"
## [19] "Big Data by Cukier, Kenneth,Mayer-Schonberger, Viktor"
## [20] "Preparing Data for Sharing: Guide to Social Science Data Archiving by Data Archiving and Networking Center"

# request header
cat(str_split(info$value()["headerOut"],"\r")[[1]][1:13])

## GET /search.php?keyisbn=data HTTP/1.1
## Host: www.biblio.com
## Accept: */*
## from: eddie@r-datacollection.com
## user-agent: R Under development (unstable) (2015-01-13 r67443), x86_64-apple-darwin10.8.0
## GET /cart.php?bid=564631291&add=1&int=keyword_search HTTP/1.1
## Host: www.biblio.com
## Accept: */*
## Cookie: variation=res_b; vis=language%3Ach%7Ccountry%3A8%7Ccurrency%3A9%7Cvisitor%3ANjLmEQ8MSM9rnev41nvgev1bi6snPACORi1mN8FbRIdQjcfVV9Bc0Ni6NZpwam605203390014340277881306013052%7Cver%3A4
## from: eddie@r-datacollection.com
## user-agent: R Under development (unstable) (2015-01-13 r67443), x86_64-apple-darwin10.8.0
There has been a change to the install_github function of the devtools package. To install Rwebdriver use:

```r
library(devtools)
install_github("crubba/Rwebdriver")
```

The website holding the UK government press releases has been altered slightly. To get the date and organisation you need to change the XPaths here...

```r
library(XML)

organisation <- xpathSApply(tmp, "/dl[@data-trackposition='top']/a[@class='organisation-link']", xmlValue)
publication <- xpathSApply(tmp, "/dl[@class='primary-metadata']//abbr[@class='date']", xmlValue)

... and here...

for(i in 2:length(list.files("Press_Releases/"))){
  tmp <- readLines(str_c("Press_Releases/", i, ".html"))
  tmp <- str_c(tmp, collapse = "")
  tmp <- htmlParse(tmp)
  release <- xpathSApply(tmp, "/div[@class='block-4']", xmlValue)
  organisation <- xpathSApply(tmp, "/dl[@data-trackposition='top']/a[@class='organisation-link']", xmlValue)
  publication <- xpathSApply(tmp, "/dl[@class='primary-metadata']//abbr[@class='date']", xmlValue)
  if(length(release) != 0){
    n <- n + 1
    tmp_corpus <- Corpus(VectorSource(release))
    release_corpus <- c(release_corpus, tmp_corpus)
    meta(release_corpus[[n]], "organisation") <- organisation[1]
  }
}
```
The prescindMeta() function is defunct as of version 0.6 of the tm package. The meta data can now be gathered with the meta() function.

```r
meta_organisation <- meta(release_corpus, type = "local", tag = "organisation")
meta_publication <- meta(release_corpus, type = "local", tag = "publication")
```

```r
meta_data <- data.frame(
  organisation = unlist(meta_organisation),
  publication = unlist(meta_publication)
)
```

The sFilter() function is also defunct. You can filter the corpus using meta().

```r
release_corpus <- release_corpus[
  meta(release_corpus, tag = "organisation") == "Department for Business, Innovation & Skills" |
  meta(release_corpus, tag = "organisation") == "Department for Communities and Local Government" |
  meta(release_corpus, tag = "organisation") == "Department for Environment, Food & Rural Affairs" |
  meta(release_corpus, tag = "organisation") == "Foreign & Commonwealth Office" |
  meta(release_corpus, tag = "organisation") == "Ministry of Defence" |
  meta(release_corpus, tag = "organisation") == "Wales Office"
]
```

The stringr package also produces a hick-up with the updated version of the tm package, thus we switch to base R.

```r
tm_filter(release_corpus, FUN = function(x) any(grep("Afghanistan", content(x))))
```

We need to wrap the replace function with the new content_transformer()...

```r
release_corpus <- tm_map(
  release_corpus,
  content_transformer(
    function(x, pattern){
      gsub(
        pattern = "$[:punct:]$",
        replacement = " ",
        x
      )
    })
  )
)
```

Moreover, the tolower() function needs to be wrapped with the content_transformer()...

```r
release_corpus <- tm_map(release_corpus, content_transformer(tolower))
```

The prescindMeta() function is also defunct on page 310...

```r
org_labels <- unlist(meta(release_corpus, "organisation"))
```
Since the sFilter() and prescindMeta() functions are defunct as of version 0.6 of the tm package, you need to change the code on page 315 to filter the corpus.

```r
short_corpus <- release_corpus[c(
  which(
    meta(
      release_corpus, tag = "organisation"
    ) == "Department for Business, Innovation & Skills"
  )[1:20],
  which(
    meta(
      release_corpus, tag = "organisation"
    ) == "Wales Office"
  )[1:20],
  which(
    meta(
      release_corpus, tag = "organisation"
    ) == "Department for Environment, Food & Rural Affairs"
  )[1:20]
)]

table(unlist(meta(short_corpus, "organisation")))
```

---

**page 243 / 9.1.5.3**

*reported by Jane Yu*

The installation of RHTMLForms package would fail - it’s omgehat.net not omegahat.org. Use this instead ...

```r
install.packages("RHTMLForms", repos = "http://www.omegahat.net/R", type="source")
```