The rvest Package in R

Amber Brodeur

5/02/2017

Contents

Introduction **1**

Objective **1**

Dataset Description **2**
Variables 2

Goals **3**

Preparation **3**

Tutorial 1 - TripAdvisor **3**

Tutorial 2 **6**

Tutorial 3 - Zillow **6**

Conclusion **9**

Works Cited **9**

Introduction

Three tutorials are explored in this project that all use the `rvest` package. Each tutorial explores different commands to scrape a webpage in the `rvest` package. The tutorials use webpages from TripAdvisor, United, and Zillow. The project is focused on learning how to use the CSS selector tool, Selectorgadget, to select nodes in HTML.

Objective

The objective of this project is to master the `rvest` package and the CSS selector tool. This will be accomplished by going through the three tutorials available in the `rvest` package. The `rvest` package is used to scrape data from the Internet and is used in conjuction with the `magrittr` package. The CSS selector tool used in this tutorial is called Selectorgadget. Firstly, "CSS stands for Cascading Style Sheets and is a language used for describing the look and formatting of a document written in a markup language."[1] The Selectorgadget is used to extract specific components from HTML. More specifically, "Selectorgadget is a javascript bookmarklet that allows you to interactively figure out what css selector you need to extract desired components from a page."[2]

[1] https://en.wikipedia.org/wiki/Cascading_Style_Sheets
[2] Wickham, Hadley; Selectorgadget

Dataset Description

There are three datasets that represent the three webpages scraped in the three tutorials. The first tutorial scrapes reviews on the JW Marriot Indianapolis hotel from the TripAdvisor webpage. The second tutorial scrapes flights from the United webpage. The third tutorial scrapes data from the Zillow webpage.

Variables

There is one set of variables for each tutorial with a total of three sets of variables. Below are three tables representing each set.

TripAdvisor Variables

The table below is the data scraped of the reviews on the JW Marriot Indianapolis hotel from the TripAdvisor webpage.

Variable Name	Description	Unit
id	primary key / unique id	id
quote	quote from the hotel review	character
date	date of the review	month, day, year
review	hotel review	character

United Variables

The table below is the data scraped on the flights from the United webpage.

Variable Name	Description	Unit

Zillow Variables

The table below is the data scraped from the Zillow webpage.

Variable Name	Description	Unit
z_id	unique id	id
address	location	character
price	listing price of house	USD
beds	number of bedrooms	numeric
baths	number of bathrooms	numeric
house_area	area size of house	square feet

Goals

Preparation

In this tutorial we used the CSS selector tool (Selector gadget). To get started with webscraping, we will first take a look at how to use the Selector gadget. To grasp a basic understanding of how to use it in R, use the command `vignette("selectorgadget")` to return documentation about how to download and use the Selector gadget.

```
vignette("selectorgadget")
```

A tutorial to practice with the Selector Gadget can be found at the following website: http://flukeout.github.io/#.

Install the packages used in this project.

```
install.packages('rvest')
install.packages('magrittr')
```

Package descriptions:

- The `rvest` package is used to scrape data from the webpage.
- The `xml2` and `httr` packages are downloaded with the `rvest` package. These two packages are used to download and manipulate XML and HTML data.
- The `magrittr` package is used for the pipe operator.
-

Load the libraries into the workspace.

```
library(rvest)
library(magrittr)
library(tidyr) #used in zillow tutorial
```

We will explore the `rvest` package by reviewing demonstrations available in the `rvest` package. The command below lists the demonstrations in the `rvest` package.

```
demo(package = "rvest")
```

There are three demonstrations available for the `rvest` package:

tripadvisor united *zillow

Tutorial 1 - TripAdvisor

In this section, a tutorial from the `rvest` package on TripAdvisor is reviewed.

The command below explores the tripadvisor demonstration from the `rvest` package. When the `ask` setting is set to true, the demonstration pauses between pages. The results are not shown for the code-chunk below.

```
demo(tripadvisor, package = "rvest", ask = FALSE)
```

Next, the Tripadvisor url is saved as an object in the R environment.

```
url <- "http://www.tripadvisor.com/Hotel_Review-g37209-d1762915-Reviews-JW_Marriott_Indianapolis-Indiana
```

The `read_html` command is used to read the the Tripadvisor url. The pipe operator is used to forward the result of this expression to the `html_nodes` command. The `html_nodes` command is used to select nodes

from a HTML document. In this case, to select the nodes of the reviews from the Tripadvisor webpage. The Selector gadget is used to find the nodes. The result is assigned to `reviews`.

```
reviews <- url %>%
   read_html() %>%
   html_nodes("#REVIEWS .innerBubble")
```

The command `length` is a base R command. This command returns the length of an object. The line of code below returns the number of reviews extracted from the TripAdvisor website.

```
length(reviews)
```

```
## [1] 10
```

There were ten reviews extracted.

The `html_text` command extracts text from HTML. The code below extracts the text of the ten Tripadvisor reviews.

```
html_text(reviews)
```

```
##  [1] ""Lovely Hotel with Nice Views"Reviewed yesterday  NEWI stayed at this hotel during Solid Edge U
##  [2] ""Clean, contemporary hotel"Reviewed 2 days ago  NEWVery clean, contemporary hotel. Several on-s
##  [3] ""JW Marriott Indy is my choice"Reviewed 5 days ago  NEWI have stayed at the JW Indy several tir
##  [4] ""More Than a Conference Hotel"Reviewed 6 days ago  NEWWe've stayed at the JW previously for bus
##  [5] ""great conference location"Reviewed 6 days ago  NEWStayed there as a result of being in town fo
##  [6] ""Great Location for a Conference"Reviewed 1 week ago  I stayed on the 5th floor of the JW Marri
##  [7] ""Just for the conference..."Reviewed 1 week ago  via mobileThe skyways connect the hotels and p
##  [8] ""Just an ok hotel. Uncomfortable beds and poor plumbing "Reviewed 1 week ago  via mobileJust an
##  [9] ""Great stay"Reviewed 1 week ago  Got a decent mid-week rate and was very impressed. Very comfor
## [10] ""Perfect "Reviewed 1 week ago  via mobileStayed here for one night for a conference. From the r
```

The `reviews` are piped to the`html_node` command. The

With the Selector gadget, click on the title/quote of the review. When hovering over this selection, in small print you will see an `a` and `span`. These are two nodes for the quote. The `a` node is used to extract the id from the reviews. The `span` node will be used later. The `html_attr` command extracts attributes from HTML.

```
id <- reviews %>%
   html_node(".quote a") %>%
   html_attr("id")
```

```
length(id)
```

```
## [1] 10
```

The id attribute is returned below.

```
id
```

```
##  [1] "rn480609179" "rn480385625" "rn479491224" "rn479350237" "rn479249461"
##  [6] "rn478902916" "rn478834424" "rn478565051" "rn478563054" "rn478210836"
```

The `span` node mentioned above is used here. The `span` node extracts the quote from the Tripadvisor review. The `html_text` command extracts text from HTML.

```
quote <- reviews %>%
   html_node(".quote span") %>%
   html_text()
```

```
quote
```

```
##  [1] "Lovely Hotel with Nice Views"
##  [2] "Clean, contemporary hotel"
##  [3] "JW Marriott Indy is my choice"
##  [4] "More Than a Conference Hotel"
##  [5] "great conference location"
##  [6] "Great Location for a Conference"
##  [7] "Just for the conference..."
##  [8] "Just an ok hotel. Uncomfortable beds and poor plumbing "
##  [9] "Great stay"
## [10] "Perfect "
```

The rating is extracted from the reviews object. The nodes are found using the css Selector gadget. The `html_attr` command extracts the attribute from HTML.

```
rating <- reviews %>%
   html_node(".rating .rating_s_fill") %>%
   html_attr("alt") %>%
   gsub(" of 5 stars", "", .) %>%
   as.integer()
```

The rating is not correct in the tutorial. All values returned NA.

```
rating
```

```
##  [1] NA NA NA NA NA NA NA NA NA NA
```

The date is extracted from the reviews object. The nodes are found using the css Selector gadget. The `html_attr` command extracts the attribute from HTML. The `strptime` command converts character vector to class "POSIXlt" and each input string is processed for the specified format. "POSIXlt" represents calendar dates and times and is a list of vectors that represents: seconds, minutes, hours, day of the month, year, day of the week, day of the year, daylight saving time, and time zone.

```
date <- reviews %>%
   html_node(".rating .ratingDate") %>%
   html_attr("title") %>%
   strptime("%b %d, %Y") %>%
   as.POSIXct()
```

```
date
```

```
##  [1] "2017-05-01 EDT" "2017-04-30 EDT" "2017-04-27 EDT" "2017-04-26 EDT"
##  [5] "2017-04-26 EDT" "2017-04-25 EDT" "2017-04-25 EDT" "2017-04-24 EDT"
##  [9] "2017-04-24 EDT" "2017-04-23 EDT"
```

There are 10 dates with the time zone in `date`.

The reviews are extracted from the reviews object. The nodes are found using the css Selector gadget. The `html_text` command extracts text from HTML.

```
review <- reviews %>%
   html_node(".entry .partial_entry") %>%
   html_text()
```

```
review
```

```
##  [1] "I stayed at this hotel during Solid Edge University last year and really enjoyed it for a multi
##  [2] "Very clean, contemporary hotel. Several on-site restaurants/bars. Fairly easy to access for a c
##  [3] "I have stayed at the JW Indy several times and would not think of staying anywhere else. The st
```

```
##  [4] "We've stayed at the JW previously for business and pleasure and eagerly looked forward to our
##  [5] "Stayed there as a result of being in town for a conference. The rooms were good sized, clean a
##  [6] "I stayed on the 5th floor of the JW Marriott Indianapolis from 4/22-4/24 for a conference that
##  [7] "The skyways connect the hotels and parking garages with the rest the city. While I did not sta
##  [8] "Just an OK hotel. Stayed here for three nights and were not very happy with the service or the
##  [9] "Got a decent mid-week rate and was very impressed. Very comfortable, close to downtown, nice r
## [10] "Stayed here for one night for a conference. From the minute you walk into the door till the mi
```

The `id`, `quote`, `date`, and `review` are combined to form a dataframe named `ta`.

```
ta <- data.frame(id, quote, date, review, stringsAsFactors = FALSE)
```

Review the structure of the `ta` dataframe.

```
str(ta)
```

```
## 'data.frame':    10 obs. of  4 variables:
##  $ id    : chr  "rn480609179" "rn480385625" "rn479491224" "rn479350237" ...
##  $ quote : chr  "Lovely Hotel with Nice Views" "Clean, contemporary hotel" "JW Marriott Indy is my c
##  $ date  : POSIXct, format: "2017-05-01" "2017-04-30" ...
##  $ review: chr  "I stayed at this hotel during Solid Edge University last year and really enjoyed it
```

The dataframe has 10 observations and four variables. Conveniently, the data looks good and does not need any clean-up.

In order to save the data set as a csv file, we set the working directory and then used the write.csv command to export the data to the file path. Below, the csv file was saved on the desktop.

```
setwd("/Users/amberbrodeur/Desktop/Web Scrape")
write.csv(ta, 'tripadvisor.csv', row.names=FALSE)
```

Tutorial 2

In this section, a tutorial from the `rvest` package on United Airlines is reviewed.

The command below explores the `united` demonstration from the `rvest` package. When the `ask` setting is set to true, the demonstration pauses between pages.

```
demo(united, package = "rvest", ask = FALSE)
```

NEED A UNITED MILEAGE PLUS USERNAME AND PASSWORD.

To Be Continued...

Tutorial 3 - Zillow

In this section, a tutorial from the `rvest` package on Zillow is reviewed.

The command below explores the `zillow` demonstration from the `rvest` package. When the `ask` setting is set to true, the demonstration pauses between pages.

```
demo(zillow, package = "rvest", ask = FALSE)
```

The `read_html` command is used to read the Zillow url into R. The url is saved as an object named `page`.

```
page <- read_html("http://www.zillow.com/homes/for_sale/Greenwood-IN/fsba,fsbo,fore,cmsn_lt/house_type/
```

The Zillow url `page` is read. The pipe operator is used to forward the Zillow webpage to the `html_nodes` command. The `html_nodes` command is used to select nodes from a HTML document. In this case, to select the nodes of the houses from the Zillow webpage. The Selector gadget is used to find the nodes. The result is assigned to `houses`.

```
houses <- page %>%
  html_nodes(".photo-cards li article")
```

```
head(houses)
```

```
## {xml_nodeset (6)}
## [1] <article data-photocount="38" data-audiencetesteventlabel="" data-gr ...
## [2] <article data-photocount="49" data-audiencetesteventlabel="" data-gr ...
## [3] <article data-photocount="43" data-audiencetesteventlabel="" data-gr ...
## [4] <article data-photocount="13" data-audiencetesteventlabel="" data-gr ...
## [5] <article data-photocount="1" data-audiencetesteventlabel="" data-gro ...
## [6] <article data-photocount="24" data-audiencetesteventlabel="" data-gr ...
```

The `html_attr` command extracts attributes from HTML. In the command below, the `id` attribute is extracted from the `houses` object. The object is saved as `z_id`.

```
z_id <- houses %>% html_attr("id")
```

```
head(z_id)
```

```
## [1] "zpid_85450049"  "zpid_85449900"  "zpid_124612912" "zpid_85465290"
## [5] "zpid_85440083"  "zpid_85464853"
```

The `html_nodes` command is used to select the photo card address from the `houses` object. The `html_text` command extracts text from HTML. The code below extracts the text from the node that was extracted. The object is saved as `address`.

```
address <- houses %>%
  html_node(".zsg-photo-card-address") %>%
  html_text()
```

```
head(address)
```

```
## [1] "2124 Cheviot Ct, Greenwood, IN"
## [2] "1819 Dockside Dr, Greenwood, IN"
## [3] "1312 Brentford Ln, Greenwood, IN"
## [4] "595 Conifer Way, Greenwood, IN"
## [5] "601 Georgetown Rd, Greenwood, IN"
## [6] "1868 Harvest Meadow Dr, Greenwood, IN"
```

The `html_nodes` command is used to select the photo card price from the `houses` object. The `html_text` command extracts text from HTML node. The `readr::parse_number()` command... The object is saved as `price`.

```
price <- houses %>%
  html_node(".zsg-photo-card-price") %>%
  html_text() %>%
  readr::parse_number()
```

```
head(price)
```

```
## [1] 489900 650000 369900 144900 225000 165000
```

The `html_nodes` command is used to select the photo card information from the `houses` object. The `html_text` command extracts text from HTML node. The object is saved as `price`. The `strplit` command...

The object is saved as `params`.

```
params <- houses %>%
  html_node(".zsg-photo-card-info") %>%
  html_text() %>%
  strsplit("\u00b7")
```

```
head(params)
```

```
## [[1]]
## [1] "6 bds "      " 5 ba "      " 4,960 sqft"
##
## [[2]]
## [1] "5 bds "      " 3 ba "      " 5,204 sqft"
##
## [[3]]
## [1] "4 bds "      " 3 ba "      " 3,385 sqft"
##
## [[4]]
## [1] "3 bds "      " 2 ba "      " 1,196 sqft"
##
## [[5]]
## [1] "4 bds "      " 2 ba "      " 2,185 sqft"
##
## [[6]]
## [1] "3 bds "      " 2.5 ba "    " 2,233 sqft"
```

The `purrr::map_chr(1)` command... The `readr::parse_number()` command... The object is saved as `beds`.

```
beds <- params %>%
  purrr::map_chr(1) %>%
  readr::parse_number()
```

```
head(beds)
```

```
## [1] 6 5 4 3 4 3
```

The `purrr::map_chr(2)` command... The `readr::parse_number()` command... The object is saved as `baths`.

```
baths <- params %>%
  purrr::map_chr(2) %>%
  readr::parse_number()
```

```
head(baths)
```

```
## [1] 5.0 3.0 3.0 2.0 2.0 2.5
```

The `purrr::map_chr(3)` command... The `readr::parse_number()` command... The object is saved as `house_area`.

```
house_area <- params %>%
  purrr::map_chr(3) %>%
  readr::parse_number()
```

```
head(house_area)
```

```
## [1] 4960 5204 3385 1196 2185 2233
```

The z_id, address, price, beds, baths, and house_area are combined to form a dataframe named z.

```
z <- data.frame(z_id, address, price, beds, baths, house_area, stringsAsFactors = FALSE)
```

Review the structure of the z dataframe.

```
str(z)
```

```
## 'data.frame':    27 obs. of  6 variables:
##  $ z_id      : chr  "zpid_85450049" "zpid_85449900" "zpid_124612912" "zpid_85465290" ...
##  $ address   : chr  "2124 Cheviot Ct, Greenwood, IN" "1819 Dockside Dr, Greenwood, IN" "1312 Brentfo
##  $ price     : num  489900 650000 369900 144900 225000 ...
##  $ beds      : num  6 5 4 3 4 3 4 2 3 3 ...
##  $ baths     : num  5 3 3 2 2 2.5 2.5 2 3 2 ...
##  $ house_area: num  4960 5204 3385 1196 2185 ...
```

The dataframe has 27 observations and 6 variables.

In order to save the data set as a csv file, we set the working directory and then used the write.csv command to export the data to the file path. Below, the csv file was saved on the desktop.

```
setwd("/Users/amberbrodeur/Desktop/Web Scrape")
write.csv(z, 'zillow.csv', row.names=FALSE)
```

Conclusion

Works Cited