Webscraping movie information off of IMDB

Lyne Cha

December 2, 2017

Introduction

Major Theme: Webscraping

After this weeks lecture and lab on XML and webscraping, I was insipired to use post02 as an opportunity to dig deeper into webscraping. Since we just learned webscraping in class, many of you might still not be familiar with what it exactly is. Web scraping is a technique for converting the data present in unstructured format (HTML tags) over the web to the structured format which can easily be accessed and used.

Why is webscraping important?

It might be hard to see the applications and the usefulness of webscraping at first glance. However, it has immense implications and can be found all around us. We can use webscraping to collect practically any type of data and analyze it. This has immense application in fields such as search engine optimization, market research, competitor analysis of reviews and sentiments. You might wonder how Google knows what type of ads to show you or what you meant to type when you make a typo. This has all to do with webscraping by big companies to collect data of our personal information and preferences.

Getting Started

For this post, I will try webscraping off of IMBD, a movie review website and try to access their data and compile it into a format that is easier to analyze.

We must first install the package rvest and load it.

```
#Loading the rvest package
library('rvest')
```

```
## Warning: package 'rvest' was built under R version 3.4.3
```

```
## Loading required package: xml2
```

```
## Warning: package 'xml2' was built under R version 3.4.2
```

```
library(magrittr)
```

Scraping IMDB

We will try to scrape the following data from the website

Ranking Name of the movie Rating

Step 1. Now, we will start with scraping the ranking. For that, we'll use the selector gadget to get the specific CSS selectors that encloses the name of the movie. You can click on the extension in your browser and select the ranking with cursor. If you don't have the selector gadget, you can download it here http://selectorgadget.com/

Step 2: Once you are sure that you have made the right selections, you need to copy the corresponding CSS selector that you can view in the bottom center.

Step 3. Once you know the CSS selector that contains the ranking, you can begin writing code to to extract the data.

Try using the selector gadget and selecting on the ranking of the movies. This should highlight it in green and yellow and the text ".text-primary" should pop up at the bottom.

```
#Specifying the url for desired website to be scrapped
url <- 'http://www.imdb.com/search/title?count=100&release_date=2016,2016&title_type=feature'

#Reading the HTML code from the website
webpage <- read_html(url)

#Using CSS selectors to scrap the rankings section
rank_data_html <- html_nodes(webpage,'.text-primary')

#Converting the ranking data to text
rank_data <- html_text(rank_data_html)

#Let's have a look at the rankings
rank_data
```

```
##   [1] "1."   "2."   "3."   "4."   "5."   "6."   "7."   "8."   "9."   "10."
##  [11] "11."  "12."  "13."  "14."  "15."  "16."  "17."  "18."  "19."  "20."
##  [21] "21."  "22."  "23."  "24."  "25."  "26."  "27."  "28."  "29."  "30."
##  [31] "31."  "32."  "33."  "34."  "35."  "36."  "37."  "38."  "39."  "40."
##  [41] "41."  "42."  "43."  "44."  "45."  "46."  "47."  "48."  "49."  "50."
##  [51] "51."  "52."  "53."  "54."  "55."  "56."  "57."  "58."  "59."  "60."
##  [61] "61."  "62."  "63."  "64."  "65."  "66."  "67."  "68."  "69."  "70."
##  [71] "71."  "72."  "73."  "74."  "75."  "76."  "77."  "78."  "79."  "80."
##  [81] "81."  "82."  "83."  "84."  "85."  "86."  "87."  "88."  "89."  "90."
##  [91] "91."  "92."  "93."  "94."  "95."  "96."  "97."  "98."  "99."  "100."
```

As you can see from the output, there are unwanted pesky periods at the end and each element is in the form of a string rather than a number. We will fix this now through the use of gsub

```
#getting rid of the period using gsub
rank_data <-gsub(".", "", rank_data, fixed = TRUE)
rank_data
```

```
##   [1] "1"   "2"   "3"   "4"   "5"   "6"   "7"   "8"   "9"   "10"  "11"
##  [12] "12"  "13"  "14"  "15"  "16"  "17"  "18"  "19"  "20"  "21"  "22"
##  [23] "23"  "24"  "25"  "26"  "27"  "28"  "29"  "30"  "31"  "32"  "33"
##  [34] "34"  "35"  "36"  "37"  "38"  "39"  "40"  "41"  "42"  "43"  "44"
##  [45] "45"  "46"  "47"  "48"  "49"  "50"  "51"  "52"  "53"  "54"  "55"
##  [56] "56"  "57"  "58"  "59"  "60"  "61"  "62"  "63"  "64"  "65"  "66"
##  [67] "67"  "68"  "69"  "70"  "71"  "72"  "73"  "74"  "75"  "76"  "77"
##  [78] "78"  "79"  "80"  "81"  "82"  "83"  "84"  "85"  "86"  "87"  "88"
##  [89] "89"  "90"  "91"  "92"  "93"  "94"  "95"  "96"  "97"  "98"  "99"
## [100] "100"
```

```
#changing it into a numeric form
rank_data<- as.numeric(rank_data)
rank_data
```

```
##   [1]   1   2   3   4   5   6   7   8   9  10  11  12  13  14  15  16  17
##  [18]  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34
##  [35]  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51
##  [52]  52  53  54  55  56  57  58  59  60  61  62  63  64  65  66  67  68
##  [69]  69  70  71  72  73  74  75  76  77  78  79  80  81  82  83  84  85
##  [86]  86  87  88  89  90  91  92  93  94  95  96  97  98  99 100
```

sidenote

I tried to webscrape the title of the movies using the same method off of rotten tomatoes but was not able to do so. After hours of looking up why I couldn't, I found out it was because the nodes are dynamically run and rvest cannot access them. You need to use RSelenium and have have phantomjs driver in your working directory. Since this seemed pretty advanced and out of scope of the context of our class, I decided not to include this in my post. You can read more about it here. https://stackoverflow.com/questions/31054826/rvest-output-returning-character0-instead-of-the-column-highlighted-with-sel

Scraping Cont.

Now lets scrape the name and rating of each respective movies.

```
#Using CSS selectors to scrap the title section
title_html <- html_nodes(webpage,'.lister-item-header a')

#Converting the title data to text
title_data <- html_text(title_html)

#title data
title_data
```

```
##   [1] "Batman v Superman: Dawn of Justice"
##   [2] "Suicide Squad"
##   [3] "Sing"
##   [4] "Split"
##   [5] "The Great Wall"
##   [6] "Fantastic Beasts and Where to Find Them"
##   [7] "Deadpool"
##   [8] "Moana"
##   [9] "Rogue One"
##  [10] "Trolls"
##  [11] "La La Land"
##  [12] "Captain America: Civil War"
##  [13] "Bad Moms"
##  [14] "Hacksaw Ridge"
##  [15] "Arrival"
##  [16] "Doctor Strange"
##  [17] "The Magnificent Seven"
##  [18] "Nocturnal Animals"
##  [19] "Hidden Figures"
```

```
##  [20] "Assassin's Creed"
##  [21] "Why Him?"
##  [22] "The Accountant"
##  [23] "A Cure for Wellness"
##  [24] "Almost Friends"
##  [25] "Allied"
##  [26] "Office Christmas Party"
##  [27] "Passengers"
##  [28] "The Bad Batch"
##  [29] "Zootopia"
##  [30] "Star Trek: Beyond"
##  [31] "Me Before You"
##  [32] "X-Men: Apocalypse"
##  [33] "The Autopsy of Jane Doe"
##  [34] "Brimstone"
##  [35] "Lion"
##  [36] "Kimi no na wa."
##  [37] "Moonlight"
##  [38] "Better Watch Out"
##  [39] "The Lost City of Z"
##  [40] "Hell or High Water"
##  [41] "Collateral Beauty"
##  [42] "Ghostbusters"
##  [43] "Keeping Up with the Joneses"
##  [44] "Manchester by the Sea"
##  [45] "Bedeviled"
##  [46] "Miss Peregrine's Home for Peculiar Children"
##  [47] "Birth of the Dragon"
##  [48] "Gold"
##  [49] "The Secret Life of Pets"
##  [50] "The Girl on the Train"
##  [51] "The Edge of Seventeen"
##  [52] "Contratiempo"
##  [53] "Leap!"
##  [54] "Jack Reacher: Never Go Back"
##  [55] "Now You See Me 2"
##  [56] "Allegiant"
##  [57] "Colossal"
##  [58] "13 Hours"
##  [59] "The Nice Guys"
##  [60] "The Jungle Book"
##  [61] "Silence"
##  [62] "Hunt for the Wilderpeople"
##  [63] "Warcraft"
##  [64] "The Legend of Tarzan"
##  [65] "Don't Breathe"
##  [66] "Sully"
##  [67] "Live by Night"
##  [68] "Jason Bourne"
##  [69] "True Crimes"
##  [70] "Captain Fantastic"
##  [71] "Gods of Egypt"
##  [72] "Killing Ground"
##  [73] "Deepwater Horizon"
##  [74] "Sausage Party"
##  [75] "The 5th Wave"
##  [76] "Almost Christmas"
##  [77] "10 Cloverfield Lane"
##  [78] "Ah-ga-ssi"
##  [79] "The Founder"
##  [80] "Finding Dory"
##  [81] "Mechanic: Resurrection"
##  [82] "Boo! A Madea Halloween"
##  [83] "The Conjuring 2"
##  [84] "In a Valley of Violence"
##  [85] "Underworld: Blood Wars"
##  [86] "Inferno"
##  [87] "The Neon Demon"
##  [88] "Fences"
##  [89] "A Monster Calls"
##  [90] "Below Her Mouth"
##  [91] "Grave"
##  [92] "Triple 9"
##  [93] "Patriots Day"
##  [94] "Storks"
##  [95] "Free Fire"
##  [96] "The Limehouse Golem"
##  [97] "Fallen"
##  [98] "Criminal"
##  [99] "Snowden"
## [100] "Nerve"
```

```
#Using CSS selectors to scrap the rating section
rating_data_html <- html_nodes(webpage,'.ratings-imdb-rating strong')

#Converting the rating data to text
rating_data <- html_text(rating_data_html)

#rating data of each movie
rating_data
```

```
##   [1] "6.6" "6.1" "7.1" "7.3" "6.0" "7.4" "8.0" "7.6" "7.8" "6.5" "8.1"
##  [12] "7.9" "6.2" "8.2" "8.0" "7.5" "6.9" "7.5" "7.8" "5.8" "6.2" "7.4"
##  [23] "6.4" "5.7" "7.1" "5.8" "7.0" "5.3" "8.0" "7.1" "7.4" "7.0" "6.8"
##  [34] "7.1" "8.1" "8.5" "7.5" "6.6" "6.6" "7.6" "6.8" "5.3" "5.9" "7.9"
##  [45] "4.4" "6.7" "5.0" "6.7" "6.6" "6.5" "7.4" "8.1" "6.8" "6.1" "6.5"
##  [56] "5.7" "6.2" "7.3" "7.4" "7.5" "7.2" "7.9" "6.9" "6.3" "7.1" "7.5"
##  [67] "6.4" "6.6" "7.2" "7.9" "5.4" "5.8" "7.2" "6.2" "5.2" "6.1" "7.2"
##  [78] "8.1" "7.2" "7.4" "5.7" "4.6" "7.4" "6.0" "5.8" "6.2" "6.2" "7.2"
##  [89] "7.5" "5.6" "7.0" "6.3" "7.4" "6.8" "6.5" "6.3" "5.5" "6.3" "7.3"
## [100] "6.6"
```

Now we have to change the rating data into a numeric form from a string.

```
#Converting rating to numerical
rating_data<- as.numeric(rating_data)
rating_data
```

```
##   [1] 6.6 6.1 7.1 7.3 6.0 7.4 8.0 7.6 7.8 6.5 8.1 7.9 6.2 8.2 8.0 7.5 6.9
##  [18] 7.5 7.8 5.8 6.2 7.4 6.4 5.7 7.1 5.8 7.0 5.3 8.0 7.1 7.4 7.0 6.8 7.1
##  [35] 8.1 8.5 7.5 6.6 6.6 7.6 6.8 5.3 5.9 7.9 4.4 6.7 5.0 6.7 6.6 6.5 7.4
##  [52] 8.1 6.8 6.1 6.5 5.7 6.2 7.3 7.4 7.5 7.2 7.9 6.9 6.3 7.1 7.5 6.4 6.6
##  [69] 7.2 7.9 5.4 5.8 7.2 6.2 5.2 6.1 7.2 8.1 7.2 7.4 5.7 4.6 7.4 6.0 5.8
##  [86] 6.2 6.2 7.2 7.5 5.6 7.0 6.3 7.4 6.8 6.5 6.3 5.5 6.3 7.3 6.6
```

Creating a Data Table

since now we have successfully extracted several attributes from the IMDB website, let's make a data table using these elements.

```
#Combining all the lists to form a data frame
movies_df<-data.frame(Rank = rank_data, Title = title_data, Rating = rating_data)
movies_df
```

```
##     Rank                                     Title Rating
## 1      1          Batman v Superman: Dawn of Justice    6.6
## 2      2                                Suicide Squad    6.1
## 3      3                                         Sing    7.1
## 4      4                                        Split    7.3
## 5      5                                The Great Wall    6.0
## 6      6     Fantastic Beasts and Where to Find Them    7.4
## 7      7                                     Deadpool    8.0
## 8      8                                        Moana    7.6
## 9      9                                    Rogue One    7.8
## 10    10                                       Trolls    6.5
## 11    11                                    La La Land    8.1
## 12    12                  Captain America: Civil War    7.9
## 13    13                                     Bad Moms    6.2
## 14    14                                Hacksaw Ridge    8.2
## 15    15                                      Arrival    8.0
## 16    16                               Doctor Strange    7.5
## 17    17                        The Magnificent Seven    6.9
## 18    18                            Nocturnal Animals    7.5
## 19    19                               Hidden Figures    7.8
## 20    20                             Assassin's Creed    5.8
## 21    21                                      Why Him?    6.2
## 22    22                               The Accountant    7.4
## 23    23                          A Cure for Wellness    6.4
## 24    24                               Almost Friends    5.7
## 25    25                                       Allied    7.1
## 26    26                       Office Christmas Party    5.8
## 27    27                                   Passengers    7.0
## 28    28                                The Bad Batch    5.3
## 29    29                                    Zootopia    8.0
## 30    30                          Star Trek: Beyond    7.1
## 31    31                              Me Before You    7.4
## 32    32                          X-Men: Apocalypse    7.0
## 33    33                       The Autopsy of Jane Doe    6.8
## 34    34                                   Brimstone    7.1
## 35    35                                        Lion    8.1
## 36    36                                Kimi no na wa.    8.5
## 37    37                                    Moonlight    7.5
## 38    38                              Better Watch Out    6.6
## 39    39                            The Lost City of Z    6.6
```

```
## 40    40                          Hell or High Water    7.6
## 41    41                           Collateral Beauty    6.8
## 42    42                                Ghostbusters    5.3
## 43    43                 Keeping Up with the Joneses    5.9
## 44    44                       Manchester by the Sea    7.9
## 45    45                                  Bedeviled    4.4
## 46    46 Miss Peregrine's Home for Peculiar Children    6.7
## 47    47                           Birth of the Dragon    5.0
## 48    48                                        Gold    6.7
## 49    49                     The Secret Life of Pets    6.6
## 50    50                       The Girl on the Train    6.5
## 51    51                       The Edge of Seventeen    7.4
## 52    52                                Contratiempo    8.1
## 53    53                                       Leap!    6.8
## 54    54                Jack Reacher: Never Go Back    6.1
## 55    55                             Now You See Me 2    6.5
## 56    56                                   Allegiant    5.7
## 57    57                                    Colossal    6.2
## 58    58                                    13 Hours    7.3
## 59    59                                The Nice Guys    7.4
## 60    60                              The Jungle Book    7.5
## 61    61                                     Silence    7.2
## 62    62                  Hunt for the Wilderpeople    7.9
## 63    63                                    Warcraft    6.9
## 64    64                         The Legend of Tarzan    6.3
## 65    65                                Don't Breathe    7.1
## 66    66                                        Sully    7.5
## 67    67                                Live by Night    6.4
## 68    68                                 Jason Bourne    6.6
## 69    69                                  True Crimes    7.2
## 70    70                            Captain Fantastic    7.9
## 71    71                                Gods of Egypt    5.4
## 72    72                              Killing Ground    5.8
## 73    73                            Deepwater Horizon    7.2
## 74    74                                Sausage Party    6.2
## 75    75                                  The 5th Wave    5.2
## 76    76                             Almost Christmas    6.1
## 77    77                           10 Cloverfield Lane    7.2
## 78    78                                     Ah-ga-ssi    8.1
## 79    79                                   The Founder    7.2
## 80    80                                Finding Dory    7.4
## 81    81                      Mechanic: Resurrection    5.7
## 82    82                      Boo! A Madea Halloween    4.6
## 83    83                              The Conjuring 2    7.4
## 84    84                      In a Valley of Violence    6.0
## 85    85                      Underworld: Blood Wars    5.8
## 86    86                                     Inferno    6.2
## 87    87                               The Neon Demon    6.2
## 88    88                                      Fences    7.2
## 89    89                               A Monster Calls    7.5
## 90    90                               Below Her Mouth    5.6
## 91    91                                        Grave    7.0
## 92    92                                    Triple 9    6.3
## 93    93                                Patriots Day    7.4
## 94    94                                      Storks    6.8
## 95    95                                   Free Fire    6.5
## 96    96                          The Limehouse Golem    6.3
## 97    97                                      Fallen    5.5
## 98    98                                    Criminal    6.3
## 99    99                                     Snowden    7.3
## 100  100                                       Nerve    6.6
```

Now we have a data table out of the attributes we extracted from a website. This allows to manipulate the data in any way we deem fit using the tools and methods we learned earlier in this class such as dplyr.

Conculsion

I struggled a lot trying to write this post on webscraping. I learned through this expereince that there was still a lot that I did not know and how complex webscraping can really be. Although I barely just scratched the surface, I hope this gives you guys a better understanding of how to go about webscraping and its useful applications in our daily lives. The selector gadget in particular is a very powerful tool that can help save a lot of time in webscraping so learn how to use it to the fullest extent. The internet is a source of limitless information and webscraping equips us with the tools to collect and analzye information from the web.

References

1. https://www.import.io/post/13-ways-use-web-scraping-tools/
2. https://stackoverflow.com/questions/31518150/gsub-in-r-not-replacing-dot
3. http://www.imdb.com/search/title?count=100&release_date=2016,2016&title_type=feature
4. http://selectorgadget.com/
5. https://blog.rstudio.com/2014/11/24/rvest-easy-web-scraping-with-r/
6. https://towardsdatascience.com/web-scraping-tutorial-in-r-5e71fd107f32
7. /https://github.com/hadley/rvest