Web Scraping With Ralger

Promothesh Chatterjee*

Web Scraping with Ralger

Instead of using package rvest, let's use an interesting new package, Ralger. As usual we will follow polite webscraping.

```
library(polite)
bow("https://amazon.com/s?k=best+seller+book+list&page=1")
```

```
## <polite session> https://amazon.com/s?k=best+seller+book+list&page=1
##     User-agent: polite R package
##     robots.txt: 152 rules are defined for 4 bots
##    Crawl delay: 5 sec
##   The path is scrapable for this user-agent
```

Let's start with best sellers on book category in Amazon.

```
library(ralger)
my_link <- "https://amazon.com/s?k=best+seller+book+list&page=1"
my_node<-".a-color-base.a-text-normal"# selector gadget will give these element ID
best_books <- scrap(link = my_link, node = my_node)
head(best_books,5)
```

```
## [1] "Speak the Blessing: Send Your Words in the Direction You Want Your Life to Go"
## [2] "Best of Children's Classics (Deluxe Hardbound Edition)"
## [3] "A Full Moon in August"
## [4] "The Complete Novels of Sherlock Holmes (Deluxe Hardbound)"
## [5] "Lessons in Chemistry: A Novel"
```

That was only one page, if we want multiple pages, we can use scrap() in conjunction with paste0().

```
base_link <- "https://amazon.com/s?k=best+seller+book+list&page="
links <- paste0(base_link, 1:7) # there are 7 pages
node<-".a-color-base.a-text-normal"

all_books<-scrap(links, node)
head(all_books)
```

```
## [1] "The Abduction of Smith and Smith: A Novel"
## [2] "The Complete Novels of Sherlock Holmes (Deluxe Hardbound)"
## [3] "A Full Moon in August"
## [4] "Best of Children's Classics (Deluxe Hardbound Edition)"
## [5] "Lessons in Chemistry: A Novel"
## [6] "The Heaven & Earth Grocery Store: A Novel"
```

Suppose we want the respective ratings, reviews, and price along with the bestseller books. We simply select the elements using selector gadget and use tidy_scrap function. The tidy_scrap function returns a tibble. If you query the amazon website many times, they will stop sending data (what happened to me and hence I used the .in version).

```
base_link <- "https://www.amazon.com/s?k=best+seller+book+list+2021&page="
links <- paste0(base_link, 1:4) # let's take 4 pages

my_nodes <- c(
  ".a-color-base.a-text-normal", # The title
  ".aok-align-bottom", # Rating
  ".a-size-small .a-link-normal .a-size-base", #Number of ratings
  ".a-price-whole") # Price whole

names <- c("title", "rating", "number of ratings", "price ") # respect the nodes order

fullds<-tidy_scrap(link = links, nodes = my_nodes, colnames = names)
head(fullds,5)
```

```
## # A tibble: 5 x 4
##   title                                    rating ‘number of ratings‘ ‘price ‘
##   <chr>                                    <chr>  <chr>               <chr>
## 1 The Family Across the Street: A totally u~ 4.2 o~ 22,589             9.
## 2 Girl A: A Novel                          4.7 o~ 25,251             3.
## 3 Vortex: An FBI Thriller                  4.1 o~ 10,231             6.
## 4 When You Trap a Tiger: (Newbery Medal Win~ 4.7 o~ 7,374              6.
## 5 Nine Lives: A Novel                      4.1 o~ 489                12.
```

You can even scrape an html table using the function table_scrap.Let's get table of top 50 stocks as per market capitalization. First, let's check if the website permits scraping.

```
bow("https://www.iweblists.com/us/commerce/MarketCapitalization.html")
```

```
## <polite session> https://www.iweblists.com/us/commerce/MarketCapitalization.html
##     User-agent: polite R package
##     robots.txt: 1 rules are defined for 1 bots
##    Crawl delay: 5 sec
##   The path is scrapable for this user-agent
```

Now, we can scrape the table using the function table_scrap.

```
data <- table_scrap(link ="https://www.iweblists.com/us/commerce/MarketCapitalization.html")
head(data)
```

```
## # A tibble: 6 x 4
##   `Rank` `Company Name`        Symbol `Market Cap ($B)`
##    <int> <chr>                 <chr>              <dbl>
## 1      1 MICROSOFT CORPORATION MSFT               3074.
## 2      2 APPLE INC.            AAPL               2667.
## 3      3 NVIDIA CORPORATION    NVDA               2150.
## 4      4 ALPHABET INC.         GOOG               1944.
## 5      5 AMAZON.COM, INC.      AMZN               1907.
## 6      6 Meta Platforms, Inc.  META               1275.
```

Overall, package ralger looks really useful and easy, feel free to peruse the github paper of the author:

https://github.com/feddelegrand7/ralger