

Lab 6 : Web Scrapping

Isabel Casas

Objective

After this lab, you should have learnt:

1. To find a product in Amazon using its ASIN
2. To scrape Amazon reviews avoiding to be bot detected

Finding the url of a product in Amazon

We follow the tutorial by [Marting Chan Blog](#)

- In Lecture 8, we learned about R's powerful functionality for parsing HTML, which enables us to scrape data from any website.
- Amazon uniquely identifies its products using an ASIN (Amazon Standard Identification Number), a unique alphanumeric code assigned to each item.
- For example, consider the 'Game of Thrones' saga with the ASIN number 0007477155 on Amazon.co.uk. Its URL is: <https://www.amazon.co.uk/Song-Ice-Fire-Volumes/dp/0007477155>.
- To find the URL of any other product on Amazon.co.uk, simply perform a search for the desired item. You will find the ASIN in its URL.
- In this way, we can efficiently gather data from specific product pages on Amazon.co.uk for various analytical or research purposes.

Exercise 1

From www.amazon.co.uk, find the ASIN of the following products:

1. Fender American Ultra Stratocaster Electric Guitar - Mocha Burst
2. PRO 11 WELLBEING Adjustable Ergonomic Kneeling Chair 3 Colours (Grey)
3. Introduction to Machine Learning with R: Rigorous Mathematical Analysis

Scraping Amazon review comments

-To access the reviews of the 'Game of Thrones' product on Amazon.co.uk, you can use the following URL: <https://www.amazon.co.uk/Song-Ice-Fire-Volumes/product-reviews/0007477155>.

- You may notice that the URL structure has changed slightly compared to the product page URL. This time, it includes 'product-reviews' in the link.
- With over 9000 reviews for this product, each page displays 10 reviews. To collect more than 10 reviews, we need to navigate through different pages. Each page has a unique URL:
 - Page 2: <https://www.amazon.co.uk/Song-Ice-Fire-Volumes/product-reviews/0007477155/?pageNumber=2>

– Page 3: <https://www.amazon.co.uk/Song-Ice-Fire-Volumes/product-reviews/0007477155/?pageNumber=3>

- To collect the review title, review text, and star rating from each page, we use the `html_nodes` function in R, which requires the HTML node containing this information. You can find and copy these nodes from the source code of the website.
- Below is an R function named `scrape_amazon()` with two parameters: `ASIN` and `page_num`. This function parses through the HTML of the pages and extracts the desired data. You can use this function to automate the process of scraping reviews from multiple pages.

```
library(tidyverse)
library(rvest)
scrape_amazon <- function(ASIN, page_num){
  url_reviews <- paste0("https://www.amazon.co.uk/product-reviews/", ASIN, "/?pageNumber=",
                        page_num)
  doc <- read_html(url_reviews) ## Assign results to `doc`
  ## Review Title
  doc %>%
    html_nodes("[class='a-size-base a-link-normal review-title a-color-base review-title-content a-text-b")
    html_text() -> review_title
  ## Review Text
  doc %>%
    html_nodes("[class='a-size-base review-text review-text-content']") %>%
    html_text() -> review_text
  ## Number of stars in review
  doc %>%
    html_nodes("[data-hook='review-star-rating']") %>%
    html_text() -> review_star
  ## Review date
  doc %>% html_nodes("[class='a-size-base a-color-secondary review-date']") %>%
    html_text() -> review_date
  ## Return a tibble with the title, review text, number of stars and page number
  tibble(review_title,
         review_text,
         review_star,
         page = page_num) %>% return()
}
```

- The chunk below shows the first rows (2 reviews) of the *tibble* (table) returned by `scrape_amazon()` for `ASIN = 0007477155` and the 5th page of reviews.

```
example <- scrape_amazon(ASIN = "0007477155", page_num = 5)
head(example$review_text, 2)
```

```
## [1] "\n\n\n\n\n\n\n\n \n \n My favourite book series. Bought to replace the ones that are fallin
## [2] "\n\n\n\n\n\n\n\n \n \n Bought as Christmas present\n \n"
```

Exercise 2

1. Check that what this R code returns corresponds to the 2 first reviews of page 5. How many pages of reviews does the saga have?
2. Use the `scrape_amazon()` function to obtain comments from the first page of the other 3 products from [amazon.co.uk](https://www.amazon.co.uk) that you looked for in Exercise 1.
3. How many pages of reviews do the other three product have?

Avoiding bot detection

- Calling the `scrape_amazon()` function 10 times implies that we are scraping 10 pages of the Amazon product-review website for ASIN =0007477155. So we are going to use a loop to do it for us using `lapply()`
- We have introduced delays of 3 seconds (`Sys.sleep()`) every 3 pages that we read. In this way, we avoid overloading web servers in a short space of time, which at the same time also helps avoid yourself being picked up as *suspicious webscraping behaviour*.
- With the same intention, we are scraping the pages in a random order – e.g. instead of scraping pages 1 – 2 – 3, you can scrape in a random order like for example 9 – 3 – 2.
- In the R chunk below, we read the 10 review pages in a random way.

```
ASIN <- "0007477155" ## Specify ASIN
page_range <- 1:10 ## Let's say we want to scrape pages 1 to 10

## Create a table that scrambles page numbers using `sample()`
## For randomising page reads!

match_key <- tibble(n = page_range,
                    key = sample(page_range,length(page_range)))
output_list <- lapply(page_range, function(i){
  j <- match_key[match_key$n==i,]$key
  message("Getting ", i, "th page of ",length(page_range), "; Actual: page ",j) ## Progress bar
  Sys.sleep(3) ## Take a three second break
  if((i %% 3) == 0){ ## After every three scrapes... take another two second break

    message("Taking a break...") ## Prints a 'taking a break' message on your console

    Sys.sleep(2) ## Take an additional two second break
  }
  return(scrape_amazon(ASIN = ASIN, page_num = j)) ## Scrape and return the value
})
```

- Let us take a look at `output_list` that contains the scraped data:

```
class(output_list)
length(output_list)
output_list[[2]]$review_text[1]
```

- Finally, we will save our data to analyse it later

```
save(output_list, file = "Lab6_2024.Rdata")
```

Exercise 3

Modify the above R code to collect several review pages automatically from the product you chose in Exercise 2.