Web Scraping en R

Francisco Pautt Guzmán, Carlos Granadillo Díaz & Juan Ibáñez Cárdenas

28 de Agosto de 2020

Contents

Este documento sirve como una guía a los asistentes para tener las bases de la realización de un Web Scrapping en R, su aplicación y conceptos importantes relacionados al enlace Sistema-Web.

1. ¿Qué es el Web Scraping?

El ***Web Scraping*** ("raspado" de páginas web) consiste en la extracción de datos significativos de una o varias páginas web determinadas, para una manipulación o análisis posterior. El web scraping está muy relacionado con la indexación de la web, la cual indexa la información de la web utilizando un robot y es una técnica universal adoptada por la mayoría de los motores de búsqueda. Sin embargo, el web scraping se enfoca más en la transformación de datos sin estructura en la web (como el formato HTML) en datos estructurados que pueden ser almacenados y analizados en una base de datos central, en una hoja de cálculo o en alguna otra fuente de almacenamiento. Alguno de los usos del web scraping son la comparación de precios en tiendas, la monitorización de datos relacionados con el clima de cierta región, la detección de cambios en sitios webs y la integración de datos en sitios webs. También es utilizado para obtener información relevante de un sitio a través de los rich snippets.

En los últimos años el web scraping se ha convertido en una técnica muy utilizada dentro del sector del posicionamiento web gracias a su capacidad de generar grandes cantidades de datos para crear contenidos de calidad.

2. Importancia del Web Scraping y Aplicaciones

El Web Scraping es aplicable en distintos sectores:

- **Comercial y Ventas:** Cualificar bases de datos de manera automática. Permite añadir datos adicionales a nuestras bases de datos de clientes, prospectos, suscriptores, etc.
- **Monitorizar precios de la competencia:** Mantener un listado actualizado a tiempo real de los precios que tiene la competencia en determinadas referencias también de venta de nuestros partners y red de minoristas.
- **Marketing e investigación de mercado:** Investigar compradores, tendencias, monitorizar nuestra marca. Nos permite rastrear la web en busca de cualquier dato: redes sociales, foros, etc. Hay mucha información de nuestros potenciales consumidores a nuestra disposición.
- **Detectar influencers:** Sería una información muy útil para planificar tu campaña de marketing, y con web scraping podrías conocerlo y organizarlo.

3. Paquete en R: Rvest

Rvest es un paquete que permite la realización del Web Scraping en el software, dicho paquete permite la extracción de datos de la web y transformarlos en información util. Este paquete está diseñado para trabajar junto con ***magittr*** para facilitar labores en el web scraping.

4. Ejemplos

A continuación se presentan ejemplos básicos sobre la aplicación de Web Scraping en R.

Ejemplo (1)

En este primer ejemplo, se hace una simple extracción de texto de una pagina web. Primero se instala la libreria Rvest y una vez instalada se carga el paquete con la función *Library*.

```
# Se carga la librería:
library(rvest)
```

```
## Warning: package 'rvest' was built under R version 3.6.3
```

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

Hecho esto, guardamos en un objeto la página web de la cual se obtendrán los datos:

```
simple <- read_html("http://dataquestio.github.io/web-scraping-pages/simple.html")
```

Ahora, visualizamos el objeto.

```
simple
```

```
## {html_document}
## <html>
## [1] <head>\n<meta http-equiv="Content-Type" content="text/html; charset=UTF-8 ...
## [2] <body>\n        <p>Here is some simple content for this page.</p>\n    </ ...
```

Finalmente, buscamos y recuperamos el texto del tag

.

```
simple %>%
  html_nodes("p") %>%
  html_text()
```

```
## [1] "Here is some simple content for this page."
```

Ejemplo (2)

En este segundo ejemplo se busca extraer texto de formato *temp*, para esto primero guardamos en un objeto la página web de la cual se obtendrán los datos:

```
web_clima <- read_html("https://forecast.weather.gov/MapClick.php?lat=37.777120000000025&lon=-122.41963
web_clima
```

```
## {html_document}
## <html class="no-js">
## [1] <head>\n<meta http-equiv="Content-Type" content="text/html; charset=UTF-8 ...
## [2] <body>\n        <main class="container"><header class="row clearfix" id=" ...
```

Ahora buscamos y recuperamos el texto de la clase *temp*

```
pronosticos_clima <- web_clima %>%
  html_nodes(".temp") %>%
  html_text()
```

Por último, llevamos el vector al formato que necesitamos con la ayuda de la libreria ***readr***:

```
library(readr)
```

```
##
## Attaching package: 'readr'
```

```
## The following object is masked from 'package:rvest':
##
##     guess_encoding
```

```
parse_number(pronosticos_clima)
```

```
## [1] 56 66 57 74 58 72 58 73 58
```