

CONFERENCIA ONLINE GRATUITA EN NUESTRO CANAL DE YOUTUBE



Web scraping utilizando

Cómo realizar extracción automática

de datos en sitios web

Panelistas:

- Dr. Martín Masci
- Mg. Rodrigo Del Rosso

Moderador: Lic. Diego Parrás

Jueves 16 de Abril de 19 a 21 hs.





AGENDA



- ➔ Motivación y el significado de programar
- ➔ Lenguaje R... ¿cómo lo instalo?
- ➔ Introducción al web scrapping y html
- ➔ Casos de estudio



Introducción a la Programación



Cómo no entrar en pánico en el primer acercamiento...

¿Por qué buscamos aprender programación?

- La ciencia computacional toca nuestras vidas cada día más, sin importar el área de trabajo o estudio
- Herramientas estándar se vuelven obsoletas ante ciertos problemas (dead end)
- Búsqueda de eficiencia (programas simples resuelven grandes problemas)
- Otros: curiosidad, simplemente tiene sentido, etc.
- **Genera pensamiento crítico: lo más valioso de aprender a programar es aprender a pensar cómo ordenar una serie de pasos para resolver un problema**





Introducción a la Programación



¿Cómo relacionamos conceptos de programación con actividades comunes?

- ¿Qué tienen en común seguir una receta de cocina o ensamblar un mueble con programar?
- **Secuencia**
- ¿Qué tienen en común elegir entre ver televisión o estudiar, o elegir entre comer una ensalada o una hamburguesa con programar?
- **Selección**
- ¿Qué tiene en común tomar un examen multiple choice con programar?
- **Repetición**





Introducción a la Programación



¿Cómo relacionamos conceptos de programación con actividades comunes?

- Programar es una estructura secuencial (**instrucción 1, instrucción 2, ...**)
- Programar se cimenta sobre un formato de condiciones (**IF - THEN o similares**)
- Programar utiliza repeticiones para resolver las limitaciones en el manejo de datos (**WHILE - FOR**)
- Todo programa complejo es una combinación de estructuras simples





Introducción a la Programación



Sabiendo que aplicamos los mismos conceptos todo el tiempo en nuestras vidas



¿Qué entendemos por programar?

- La creación de un set de órdenes o instrucciones para resolver un problema, utilizando una computadora
- Resulta tan importante las instrucciones como su ordenamiento

¿Por qué usamos lenguajes de programación?

- La PC procesa instrucciones en **lenguaje binario** (lenguaje de "**bajo nivel**"), los lenguajes de programación permiten su procesamiento con palabras.
- Lenguajes de **alto nivel** (**VBA, Python, R, etc**) permiten una interfaz más simple para el usuario (leíble y entendible)
- Compiladores interpretan el lenguaje y lo transforman en computacional



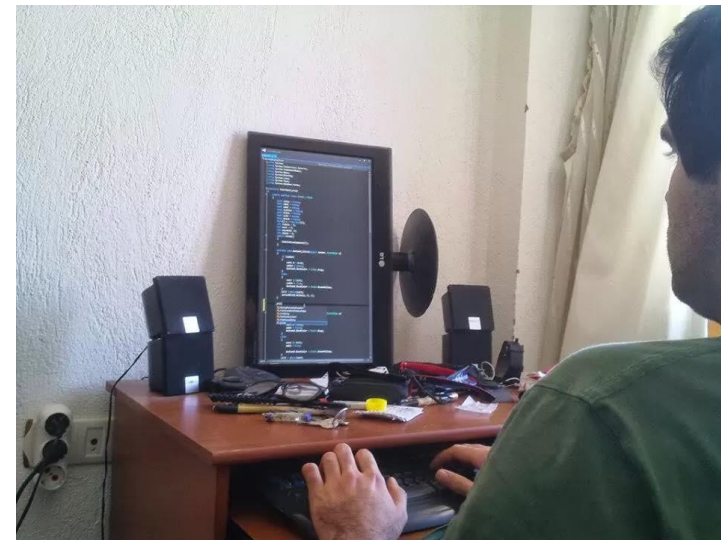
¿Qué es un programa?

Es una secuencia finita de instrucciones

¿Qué es una instrucción?

Es una operación que:

- Transforma los datos (el estado) o bien
- Modifica el flujo de ejecución



Ejemplo: Instrucciones de uso de un Shampoo

- i. Humedecer el cabello y aplicar el Shampoo "Fulanito"
- ii. Dar un ligero masaje en el cabello y el cuero cabelludo
- iii. Dejar reposar 5 minutos
- iv. Enjuagar
- v. Repetir la operación si se desea.



Nuevos desafíos profesionales



- ➔ Los profesionales de ciencias económicas deberán entender como funcionan los nuevos negocios digitales, como implementarlos, como controlarlos y como auditarlos.
- ➔ Deberán estar atentos a nuevos desafíos y a nuevos competidores que aparecen repentinamente.
- ➔ Los CIOs (Gerentes de sistemas) pasan a tener un rol cada vez mas importante. Dejan de ser quienes solo gestionan los sistemas y pasan a buscar el modo que las organizaciones hagan un uso estratégico de las TICs.

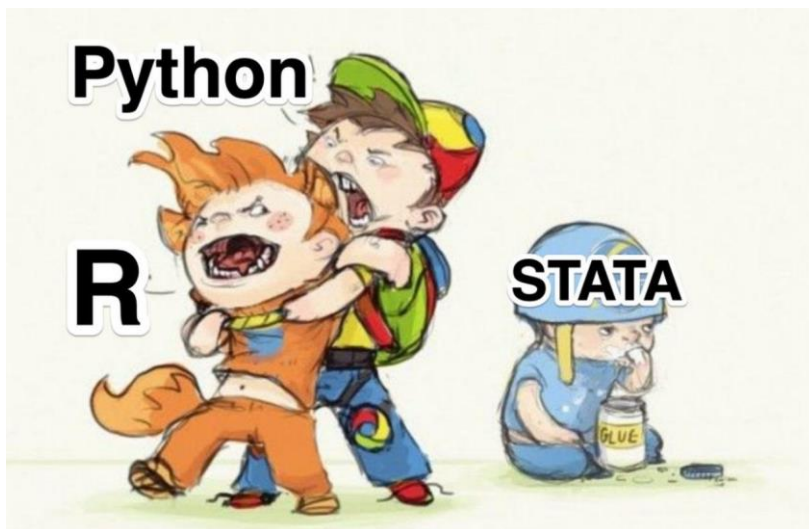




Vamos a usar el lenguaje



(veamos como instalar todo...)





Ventajas

- Gratis y Open source
- Comunidad muy activa (acceso a cualquier librería cargada en la web), o "estado del arte"
- Altamente flexible y extensible a cualquier necesidad
- Fácil implementación relativa a métodos estadísticos y amplia potencia de gráficos
- Interfaz disponible con SQL para manejo de datos

Desventajas

- Curva de aprendizaje "empinada" en el inicio
- No es un software de minería de datos, por lo que resulta lento ante grandes volúmenes
- Sin soporte comercial o ayuda útil para el usuario



¿Podemos hacer cualquier cosa con R?

Primero, debemos tener instalado todo lo necesario...



Instalación



Para instalar R bajo el sistema operativo Windows,

1. Ingresar a la siguiente página <https://www.r-project.org/>
2. Seleccionar "Download" y se abrirá la siguiente pantalla con distintos "Mirrors" (Espejos) para descargar el software.



[Home]

Download

CRAN

R Project

About R

Logo

Contributors

What's New?

Reporting Bugs

Development Site

Conferences

Help



The R Project for Statistical Computing

Getting Started

R is a free software environment for statistical computing and graphics. It compiles and runs on a wide variety of UNIX platforms, Windows and MacOS. To **download R**, please choose your preferred **CRAN mirror**.

If you have questions about R like how to download and install the software, or what the license terms are, please read our [answers to frequently asked questions](#) before you send an email.

News

- **The R Journal Volume 9/1** is available.
- **R version 3.4.1 (Single Candle)** has been released on Friday 2017-06-30.
- **R version 3.3.3 (Another Canoe)** has been released on Monday 2017-03-06.
- **The R Journal Volume 8/2** is available.



Instalación



CRAN Mirrors	
The Comprehensive R Archive Network is available at the following URLs, please choose a location close to you. Some statistics on the status of the mirrors can be found here: main page , windows re	
If you want to host a new mirror at your institution, please have a look at the CRAN Mirror HOWTO .	
0-Cloud https://cloud.r-project.org/ http://cloud.r-project.org/	Automatic redirection to servers worldwide, currently sponsored by Rstudio Automatic redirection to servers worldwide, currently sponsored by Rstudio
Algeria https://cran.usthb.dz/ http://cran.usthb.dz/	University of Science and Technology Houari Boumediene University of Science and Technology Houari Boumediene
Argentina http://mirror.fcaglp.unlp.edu.ar/CRAN/	Universidad Nacional de La Plata
Australia https://cran.csiro.au/ http://cran.csiro.au/ https://mirror.aarnet.edu.au/pub/CRAN/ https://cran.ms.unimelb.edu.au/ https://cran.curtin.edu.au/	CSIRO CSIRO AARNET School of Mathematics and Statistics, University of Melbourne Curtin University of Technology
Austria https://cran.wu.ac.at/ http://cran.wu.ac.at/	Wirtschaftsuniversität Wien Wirtschaftsuniversität Wien

3. Seleccionar el espejo más cercano a su localidad. En nuestro caso, Argentina, Chile o Brasil. Hay un espejo en la Universidad Nacional de La Plata (UNLP)

4. Seleccionar el Sistema Operativo bajo el cual correrán "R".



Instalación



Subdirectories:

[base](#)

Binaries for base distribution (managed by Duncan Murdoch). This is what you want to [install R for the first time](#).

[contrib](#)

Binaries of contributed CRAN packages (for R \geq 2.11.x; managed by Uwe Ligges). There is also information on [third services](#) and corresponding environment and make variables.

[old contrib](#)

Binaries of contributed CRAN packages for outdated versions of R (for R $<$ 2.11.x; managed by Uwe Ligges).

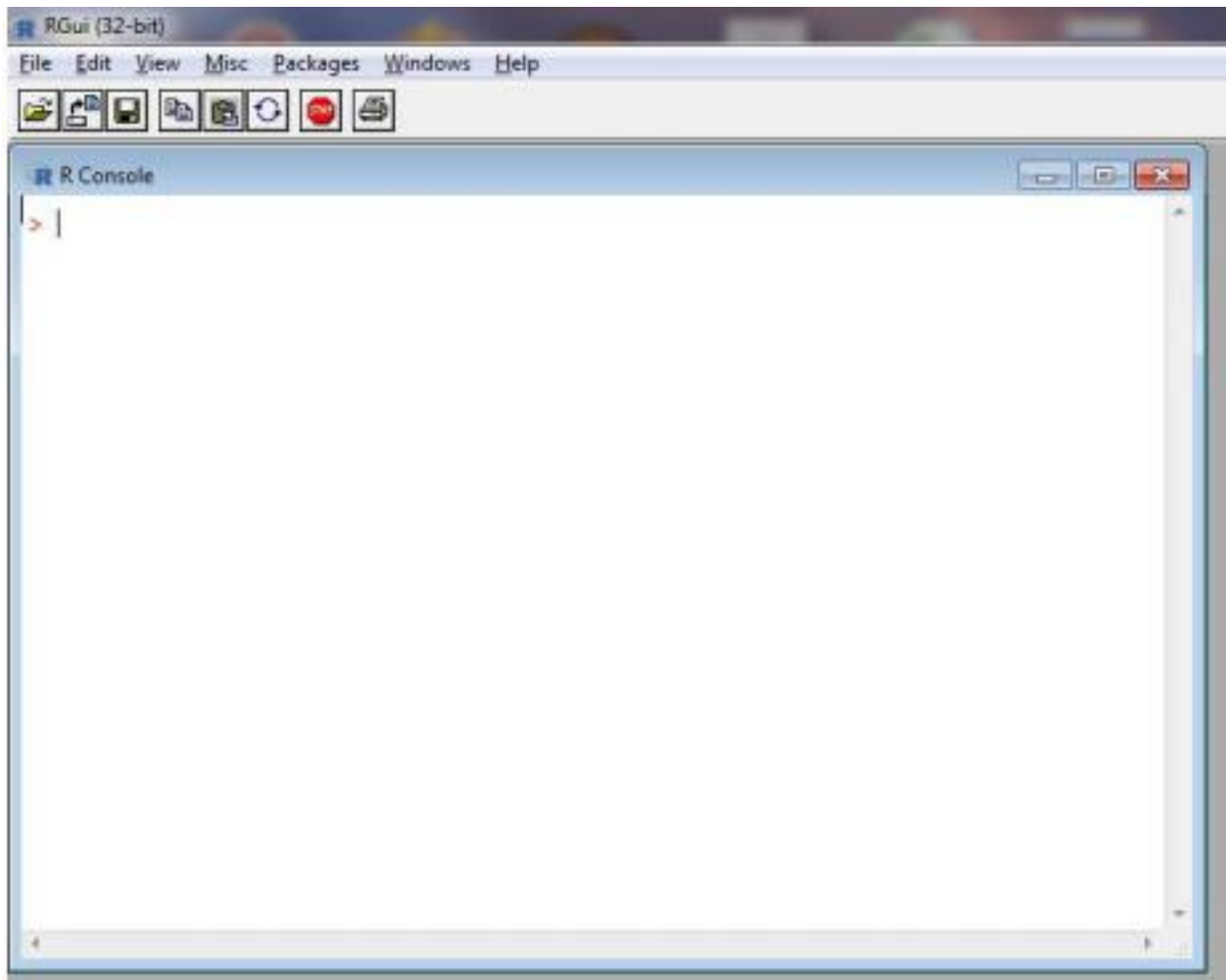
[Rtools](#)

Tools to build R and R packages (managed by Duncan Murdoch). This is what you want to build your own packages or

5. Seleccionar-> base ->install R for the first time
6. Download R-3.6.0 for Windows (32/64 bit) (figurará la última versión al 29/02/2020 es la 3.6.3)
7. Guardar el archivo en algún directorio de su PC.
8. Buscar el archivo y ejecutarlo. Al realizar esta operación podrán seleccionar el idioma durante la instalación, como así también otras opciones, pero recomendamos que al instalarlo por primera vez mantengan la configuración por default.
9. Al finalizar la instalación se creará un icono en el Escritorio. Al hacer doble clic, aparece la siguiente pantalla con la consola de "R".



Instalación



¿Esta consola no es muy amigable no?

Bienvenidos

Nuevo
Espacio



AL
FUTURO





Instalación



Existen diversas interfaces gráficas de usuario (GUI) para "R". La más utilizada es "RStudio".

1. Ingresar a la siguiente dirección <https://www.rstudio.com/>

2. Seleccionar "Download" y aparecerá la siguiente pantalla,

The screenshot shows the RStudio website's download page. At the top, there's a navigation bar with links: 'rstudio::conf', 'Products', 'Resources', 'Pricing', 'About Us', 'Blogs', and a search icon. Below the navigation bar, the 'R Studio' logo is on the left. The main heading is 'Choose Your Version of RStudio'. A subheading describes RStudio as a set of integrated tools for R, including a console, editor, and plotting tools. Below this, there are five product cards arranged in a row:

Product	License	Price
RStudio Desktop	Open Source License	FREE
RStudio Desktop	Commercial License	\$995 per year
RStudio Server	Open Source License	FREE
RStudio Server Pro	Commercial License	\$9,995 per year
RStudio Server Pro + RStudio Connect	Commercial License	\$29,995 per year



3. Seleccionar la descarga gratuita para Escritorio -Rstudio Desktop- (FREE), nuevamente en función de su Sistema Operativo.

Installers for Supported Platforms

Installers

RStudio 1.0.153 - Windows Vista/7/8/10

RStudio 1.0.153 - Mac OS X 10.6+ (64-bit)

RStudio 1.0.153 - Ubuntu 12.04-15.10/Debian 8 (32-bit)

RStudio 1.0.153 - Ubuntu 12.04-15.10/Debian 8 (64-bit)

RStudio 1.0.153 - Ubuntu 16.04+/Debian 9+ (64-bit)

RStudio 1.0.153 - Fedora 19+/RedHat 7+/openSUSE 13.1+ (32-bit)

RStudio 1.0.153 - Fedora 19+/RedHat 7+/openSUSE 13.1+ (64-bit)

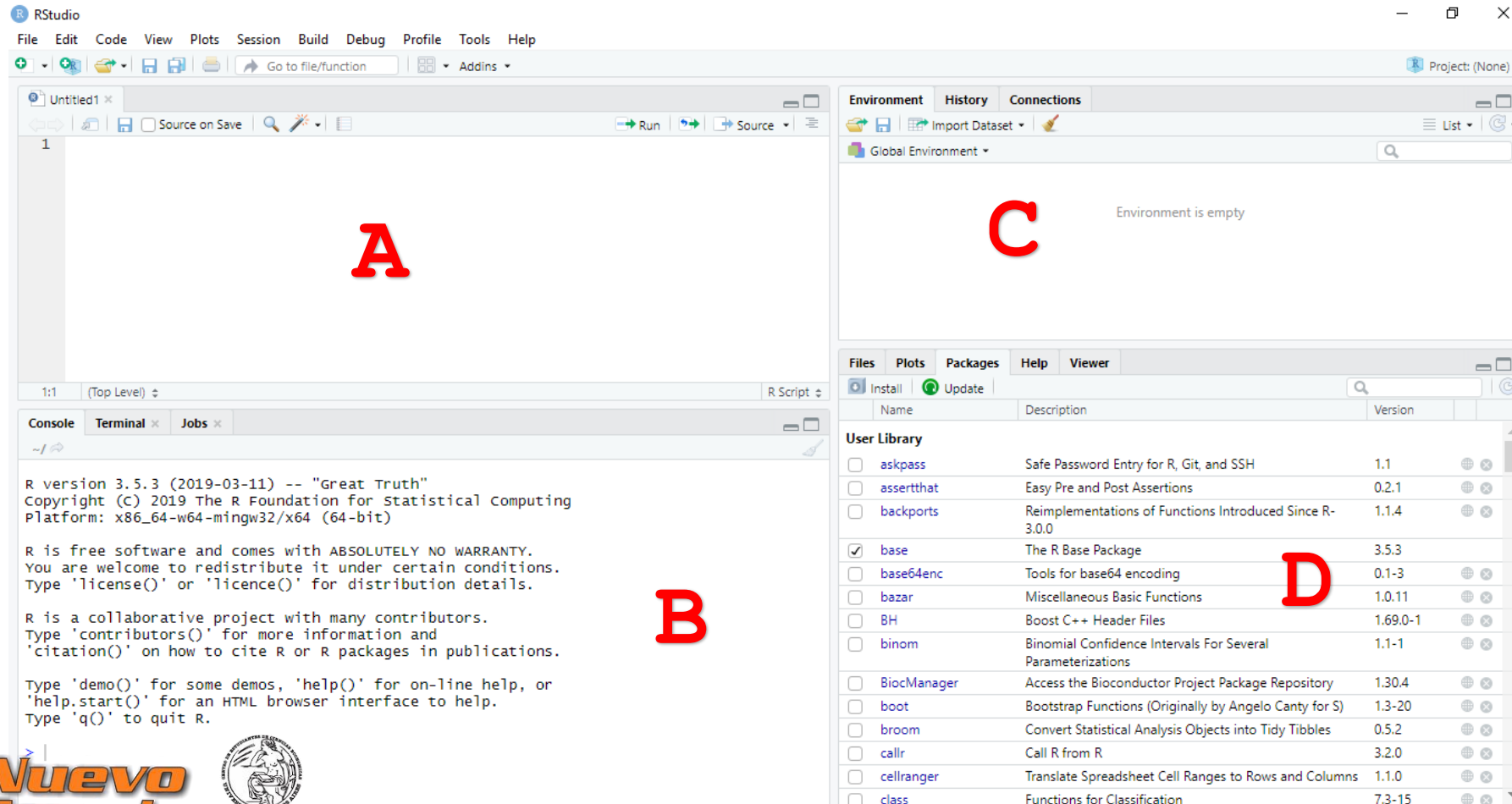
4. Luego de realizar la ejecución del archivo descargado y seguir las instrucciones para su instalación, podrán comenzar a utilizar dicha GUI. La pantalla que aparecerá al abrir el programa es la siguiente:



Instalación



4. Luego de realizar la ejecución del archivo descargado y seguir las instrucciones para su instalación, podrán comenzar a utilizar dicha GUI. La pantalla que aparecerá al abrir el programa es la siguiente:



- A. Editor de código
- B. Consola (muestra resultados)
- C. Visualización de objetos en memoria
- D. Browser (files, gráficos, helps, paquetes)



Introducción a WEB SCRAPING





Scraping



El concepto de web scraping se refiere a una técnica de recolección de datos presentes en páginas web a través de un protocolo HTML o PHP. En general esto se logra a través de un programa automatizado (*bot*) que efectúa consultas (*queries*) a un servidor web, descarga información (generalmente en formato html) y re-expresa esa información (*parsing*) en el formato necesario para su uso. ¿Por qué puede ser deseable?

- Los datos presentes en una página web están formateados para ser legibles por humanos, no para su procesamiento y análisis.
- Puede no haber otra forma de acceder a los datos.
- Datos generados periódicamente son candidatos naturales de estas técnicas, ya que hacerlo a mano es prohibitivamente costoso para períodos largos.
- Es escalable, incluso si interesan datos en un momento del tiempo, su volumen también puede ser prohibitivo para su descarga por un usuario físico, especialmente si las fuentes no provienen de un mismo servidor.



Scraping



Web Scraping es una técnica para obtener datos no estructurado (etiquetas HTML) de una página web, a un formato estructurado de columnas y renglones, en los cuales se puede acceder y usar más fácilmente. Esta técnica también es conocida como,

- Rastreo web
- Data Scraping
- Extracción de datos

Existen diversos programas para scrapear una página, e incluso sitios web completos.

Estos programas son llamados, con frecuencia, bot, spider o crawler.

Siendo software para diversas aplicaciones, están limitados, por lo mismo, siempre es mejor desarrollar código a la medida.



Scraping



Algunas cuestiones a tener en cuenta:

- **No todas las páginas, ni todas las secciones de una página son accesibles legalmente.** Qué información se puede acceder para cada página viene dada por los Términos de servicio (TOS), y un archivo llamado robots.txt.
- Los TOS son legibles por humanos y nos dicen cuales son las reglas para el acceso automatizado, que tipo de información recolecta una página y que se hace con ella, además de varios *disclaimers* legales.
- Podemos pensar en un robots.txt como un TOS para *bots*.

Por convención este archivo se encuentra en

<http://website.com/robots.txt>

Donde website se sustituye por la página a acceder.

Ejemplo, entrar a <https://www.clarin.com/robots.txt>

Que dice que todas las paginas son accesibles, salvo las que dicen **Disallow**





Scraping



Algunas cuestiones a tener en cuenta:

- **El web scraping es más efectivo cuando se tienen páginas web estáticas y bien estructuradas.** Esto se debe a que la mayoría de las metodologías dependen de bajar el código html subyacente a la página, el cuál es mutable a través del tiempo.
- En algunos casos existe una metodología alternativa relacionada con las ***Application Programming Interface (APIs)***. En términos generales las APIs son preferibles al web scraping. Tener acceso a una API puede ser costoso, imposible, o no accesible para muchos problemas. **La fortaleza del web scraping es su flexibilidad.**
- Es importante limitar el llamado a las páginas html ya que existe riesgo de bloqueo de IP en estos casos.



Formas de Scraping



De forma funcional, podemos scrapear datos, es decir, obtenerlos de diferentes maneras, algunas muy arcaicas,

- Copiar y pegar, obviamente realizado por algún humano: esta es una forma lenta, nada eficiente de obtener datos de la web.
- Coincidencia de patrones de texto: otro enfoque simple y poderoso para extraer información de la web, es mediante el uso de expresiones regulares a través de lenguajes de programación.
- Interfaz API: sitios web como Facebook, Twitter, LinkedIn, entre otros, proporcionan una API pública y / o privada, a las que se tiene acceso usando programación, para recuperar los datos en el formato deseado.
- Análisis de DOM: a través de algunos programas es posible recuperar el contenido dinámico, o partes de una pagina de sitios web, generado por scripts desde el cliente.



Breve introducción al HTML



Las siglas HTML significan Hyper Text Markup Language, un lenguaje que describe la estructura de las páginas web. Algunas características:

- Hay una jerarquía. Página de HTML esta compuesta de elementos.
- Los elementos tienen *tags*.
- Tags son nombres para el contenido (título, tabla, etc.)
- `<algo>` abre una sección (algo es el tag de la sección).
- `</algo>` cierra esa sección.
- Entre apertura y cierre se encuentra el contenido.



Breve introducción al HTML



Una sección puede y en general tiene más secciones adentro. De manera que podemos tener algo parecido a la figura, que se lee:

- `<html>`: abro sección html (página)
- `<head>`: abro encabezado
- `<title>`: abro título
- Page Title: el título de la página (**Esto es contenido**)
- `</head>`: cierro encabezado
- `<body>`: abro cuerpo
- `<h1>`: abro heading
- My First Heading: el heading (**Esto es contenido**)
- `</h1>`: cierro heading
- `<p>`: abro párrafo
- My First Paragraph: el párrafo (**Esto es contenido**)
- `</p>`: cierro párrafo
- `</body>`: cierro cuerpo
- `</html>`: cierro la página

```
<html>
<head>
<title>Page Title</title>
</head>
<body>

<h1>My First Heading</h1>
<p>My first paragraph.</p>

</body>
</html>
```



Breve introducción al HTML



Todos los elementos de HTML pueden tener atributos que nos dan información adicional para ese elemento (por ejemplo, su formato).

Algunos atributos vienen preespecificados:

- **href** significa que lo que sigue es un link
- **src** significa que lo que sigue es el nombre de archivo de una imagen
- **lang** permite especificar lenguaje
- **style** permite especificar estilo (color, tamaño de fuente, etc)
- **width** permite especificar el ancho de una imagen
- **height** permite especificar el alto de una imagen

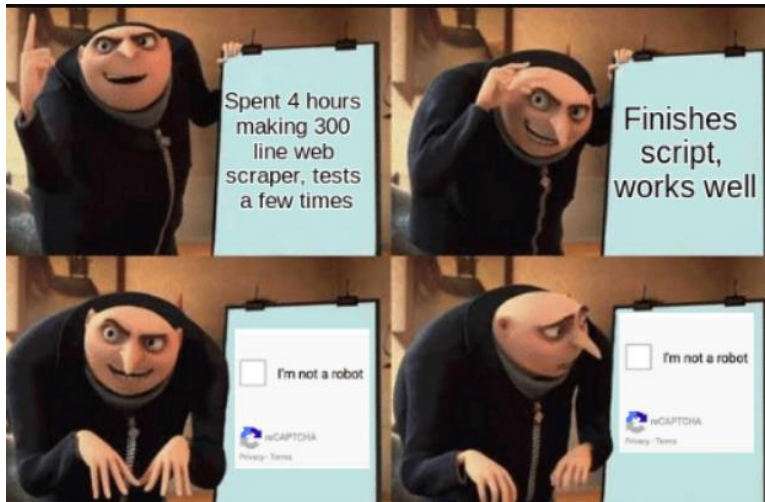


Breve introducción al HTML



Un atributo especificado por el usuario es una clase:

- Acá todas las ciudades que pertenecen a la Clase de "cities" van a tener fondo negro y color Blanco



```
<!DOCTYPE html>
<html>
<head>
<style>
.cities {
  background-color: black;
  color: white;
  margin: 20px;
  padding: 20px;
}
</style>
</head>
<body>

<div class="cities">
  <h2>London</h2>
  <p>London is the capital of England.</p>
</div>

<div class="cities">
  <h2>Paris</h2>
  <p>Paris is the capital of France.</p>
</div>

<div class="cities">
  <h2>Tokyo</h2>
  <p>Tokyo is the capital of Japan.</p>
</div>

</body>
</html>
```



Scraping



Obtendremos descripciones para cada producto a scrapear en Amazon,

- Títulos (nombre del producto)
- Precio, entre otros datos.

Además, veremos los problemas más comunes que surgen al obtener datos de Internet debido a la falta de uniformidad en el código del sitio web.

Daremos una de las soluciones, la que ocupamos para completar la tarea a realizar.

¿Para qué sirve Web Scraping?

¿En qué podríamos usar web Scraping?



Aplicaciones de Scraping



- Clasificar productos o servicios para crear motores de recomendación
- Obtener datos de texto, por ejemplo de Wikipedia, entre otras fuentes para hacer sistemas basados en Procesamientos de Lenguajes Naturales (Deep Learning)
- Generar datos de las etiquetas de imágenes, de sitios web como Google, Flickr, etc. para entrenar modelos de clasificación de imágenes
- Consolidar datos de redes sociales: Facebook y Twitter, para realizar Análisis de Sentimiento u opinión.
- Extraer comentarios de usuarios y de sitios de comercio electrónico como Amazon, Walmart, etc.
- Conocer la reputación online de funcionarios, marcas, artistas, productos, etc.
- Optimizar precios de tiendas online, a través del análisis histórico de la competencia.
- Conocer los resultados de búsqueda en Google de diversas palabras clave.
- Identificar las posiciones en dichos resultados, tipo de contenidos, y mucho más.



Scraping con R



Los requisitos previos para hacer **web scraping con R**, básicamente son dos:

- Tener instalado R, descargarlo desde su página [CRAN R - Project](https://cran.r-project.org/).
- Contar con conocimiento práctico del lenguaje R. Aunque, para esta tarea no es tan necesario...
- Utilizar el paquete **rvest** de R, escrito por [Hadley Wickham](https://www.hadley.co.uk/).
- Suponer que tienen conocimiento básico de HTML y CSS, es una ventaja.
- Usar un complemento de código abierto llamado [SelectorGadget](#). Será suficiente para que cualquiera pueda hacer web scraping. instala o descargar la extensión del gadget. Asegurarse tener esta extensión instalada.
- Usar Google Chrome, dado que se encuentra disponible esta extensión en su web store.

Como en todo existen alternativas diversas para obtener la clase CSS, sea a través de otros complementos o directamente observando el código fuente de la página.



¿Qué es RVEST?

Esencialmente es una librería (o paquete) de R.

Esencialmente permite extraer y manipular datos de una página web, usando html y xml.

Se distribuye bajo la licencia GPL-3 (General Public Licence).

El autor se inspiró en las librerías Robobrowser y beatiful soup escritas en Python.

¿Cómo instalarlo?

Para instalar la librería revest requiere iniciar la consola de R.



Scraping con R



Se requiere conocer las instrucciones en código, a las que llamaremos funciones, para para hacer las tareas más comunes en la **extracción y manipulación de datos web**. A continuación, en negrita, se listan las funciones más importantes. Entre comillas se describirán los parámetros más usados.

- **read_html**(«url») con esta función se crea un objeto que contiene todo el código o etiquetas HTML.
- **html_nodes**(«objeto html», «etiqueta css») es usada para seleccionar partes del objeto que contiene todo el código html. El segundo parámetros es la clase CSS que está relacionada con la sección que deseamos extraer.
- **html_name**() obtiene los atributos html
- **html_text**() extrae el texto html
- **html_attr**() regresa los atributos específicos html
- **html_attrs**() obtiene los atributos html
- **html_table**() convierte una tabla html en una estructura de datos en R



Caso de aplicación

Veamos todo lo anterior en la práctica...





Caso de estudio - Amazon



En general, lo que tenemos que hacer es ir a una página, en este caso [Amazon México](#) y obtener la URL.

- En la página principal, elegir una categoría, en nuestro elegiremos la cocina, seguido la categoría de café y té.
- Para finalizar en cafeteras.
- De la página anterior obtenemos la URL.
- Luego cargamos la librería en R y leemos la página web.



Caso de estudio - Amazon



Si bien las necesidades de obtener datos, depende de cada proyecto.

Para hacer más sencilla la explicación se recopilará los siguientes dos datos.

- Producto, es el título de los productos.
- Precio, obviamente hace referencia al precio del producto.

Identificar la clase CSS para scrapear


Identifiquemos la clase CSS que está relacionada con el título o nombre del producto.

Para ello seleccionamos el complemento y después el nombre del producto. Con esto obtendremos la clase.




Caso de estudio - Amazon






Patrocinado ⓘ
Hamilton Beach 46299 Cafetera para 12 Tazas Programable, Estándar, color Negro, Paquete de 1
de Hamilton Beach
\$649.00 \$959.00
✓prime
Envío GRATIS en pedidos elegibles
★★★★☆ < 235

Más vendido



Oster Cafetera, 12 Tazas, Bvstcdw12B-013, Negra, de Oster
\$469.00 \$738.00
✓prime
Recíbelo **Mañana, abr 15**
Más opciones de compra
\$436.17 de caja abierta (1 oferta)
Envío GRATIS en pedidos mayores a \$499
★★★★☆ < 251



Taurus Cafetera, 650W, Negra, de Taurus
\$298.00 \$449.00
✓prime
Recíbelo **Mañana, abr 15**
Envío GRATIS
★★★★★ < 100

Elements

Console

Sources

Network

Performance

Memory

2

×

```
...erore
  ▶ ka class="a-link-normal s-access-detail-page s-color-twister-
    title-link a-text-normal" title="Hamilton Beach 46299 Cafetera para
    12 Tazas Programable, Estándar, color Negro, Paquete de 1" href="/
    gp/sredirect/picassoRedirect.html/
    ref=pa_sp_atf_browse_kitchen_sr_pg1_
    26psc%3D1&qualifier=1586882041&id=117510696740006&widgetName=sp_atf
    browse">...</a>
    ::after
  </div>
  ▶ <div class="a-row a-spacing-none">...</div>
    ::after
  </div>
```

html body div#a-page header.nav-opt-sprite.nav-locale-mx.nav-lang-es.nav-ssl.nav-unrec

a.a-link-normal.s-access-detail-page.s-color-twister-title-link.a-text-normal 1 of 27 Cancel

Styles Event Listeners DOM Breakpoints Properties Accessibility

Filter :hov .cls +

element.style {

article, aside, details, 51tax7M48-L.../AmazonUI:3

figcaption, figure, footer, header, hgroup, nav, section {

display: block;

margin -

border -

padding -

628.667 x 132

Console What's New

Highlights from the Chrome 80 update

Support for let and class redeclarations



Caso de estudio - Amazon



Patrocinado ⓘ
Hamilton Beach 46299 Cafetera para 12 Tazas Programable, Estándar, color Negro, Paquete de 1
de Hamilton Beach
\$649.00 \$959.00
✓prime
Envío GRATIS en pedidos elegibles
★★★★☆ ~ 235

Más vendido



Oster Cafetera, 12 Tazas, Bvstddcw12B-013, Negra, de Oster
\$469.00 \$738.00
✓prime
Recíbelo Mañana, abr 15
Más opciones de compra
\$436.17 de caja abierta (1 oferta)
Envío GRATIS en pedidos mayores a \$499
★★★★☆ ~ 251



Taurus Cafetera, 650W, Negra, de Taurus
\$298.00 \$449.00
✓prime
Recíbelo Mañana
Envío GRATIS
★★★★★



```
<span class="a-size-small a-color-secondary"></span>  
<span class="a-size-base a-color-price s-price a-text-bold">  
$649.00</span>  
</a>  
<span class="a-letter-space"></span>  
<span aria-label="Suggested Retail Price: $959.00" class="a-size-small a-color-secondary a-text-strike">$959.00</span>  
::after  
</div>
```

html body div#a-page header.nav-opt-sprite.nav-locale-mx.nav-lang-es.nav-ssl.nav-unrec

span.a-size-base.a-color-price.s-price.a-text-bold 1 of 25 Cancel

Styles Event Listeners DOM Breakpoints Properties Accessibility

Filter :hov .cls +

```
element.style {  
}  
  
article, aside, details, 51tax7M48-L.../AmazonUI:3  
figcaption, figure, footer, header, hgroup, nav, section {  
  display: block;  
}
```

Console What's New x

Highlights from the Chrome 80 update

Support for let and class redeclarations



Internet Movie Database

¡OTRO caso de aplicación!





Caso de estudio - IMDB



IMDB es una base de datos de películas y series con información muy variada, títulos, cast, ratings, recaudación, año de lanzamiento, y genero entre otros.

Vamos a generar una base de datos con información de las 50 películas más valoradas por los usuarios a partir de:

- <https://www.imdb.com/robots.txt> - Chequear acceso
- https://www.imdb.com/search/title?groups=top_250&sort=user_rating - datos

Queremos armar un dataframe con:

- Título
- Año de estreno
- Duración en minutos
- Genero
- Rating promedio
- Cantidad de valoraciones
- Recaudación (en millones de \$)



Caso de estudio - IMDB



Vamos a usar las siguientes **librerías** (paquetes de R):

- **robotstxt** - nos va a permitir chequear si tenemos acceso
- **rvest** - descargar html
- **selectr** - ayuda con el parsing
- **xml2** - más parsing
- **dplyr** - manipulación de datos
- **stringr** - manipulación de strings
- **forcats** - dependencias
- **magrittr** - dependencias
- **tidyr** - nos permite cambiar la sintaxis de r y usar el operador %>%
- **ggplot2** - graficos
- **lubridate** - dependencias
- **tibble** - formateo de dataframe
- **purrr** - dependencias





Caso de estudio - IMDB



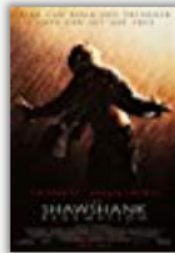
Para ver el código html ponemos **inspect** (o inspeccionar elemento):

IMDb "Top 250" (Sorted by IMDb Rating Descending)

1-50 of 250 titles. | [Next »](#)

View Mode: [Compact](#) | **[Detailed](#)**

Sort by: [Popularity](#) | [A-Z](#) | **[User Rating ▼](#)** | [Number of Votes](#) | [US Box Office](#) | [Runtime](#) | [Year](#) | [Release Date](#) | [Date of Your Rating](#) | [Your Rating](#)



1. **Cadena perpetua** (1994)

13 | 142 min | Drama

★ **9,3**

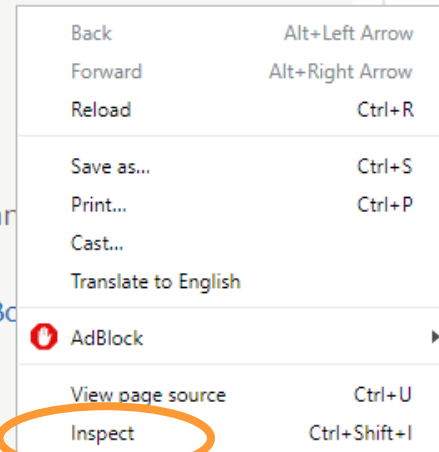
☆ [Rate this](#)

80 Metascore

Two imprisoned men bond over a number of years, finding solace and redemption through acts of common decency.

Director: [Frank Darabont](#) | Stars: [Tim Robbins](#), [Morgan Freeman](#), [Bo Diddley](#), [Samuel L. Jackson](#), [James Whitmore](#), [Cliff Gorman](#), [John Cazale](#), [John Cazale](#), [John Cazale](#), [John Cazale](#)

Votes: 2.093.324 | Gross: \$28.34M

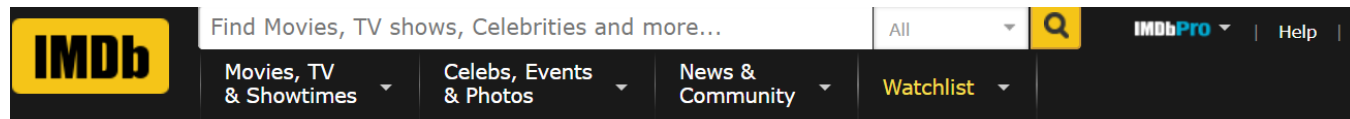




Caso de estudio - IMDb



Por encima del título nos marca el elemento html con su tag. Como el título es, además, un hipervínculo tenemos un atributo *href*, y una clase que le da el mismo formato a cada película de la lista:



IMDb "Top 250" (Sorted by IMDb Rating Descending)

1-50 of 250 titles. | [Next »](#)

View Mode: [Compact](#) | [Detailed](#)

Sort by: [Popularity](#) | [A-Z](#) | [User Rating ▼](#) | [Number of Votes](#) | [US Box Office](#) | [Runtime](#) | [Year](#) | [Release Date](#) | [Date of Your Rating](#) | [Your Rating](#)



1. [Cadena perpetua](#) (1994)

13 | 142 min | Drama

★ 9,3

☆ [Rate this](#)

80 Metascore

Two imprisoned men bond over a number of years, finding solace and eventual redemption through acts of common decency.

Director: [Frank Darabont](#) | Stars: [Tim Robbins](#), [Morgan Freeman](#), [Bob Gunton](#), [William Sadler](#)

Votes: 2.093.324 | Gross: \$28.34M



2. [El padrino](#) (1972)

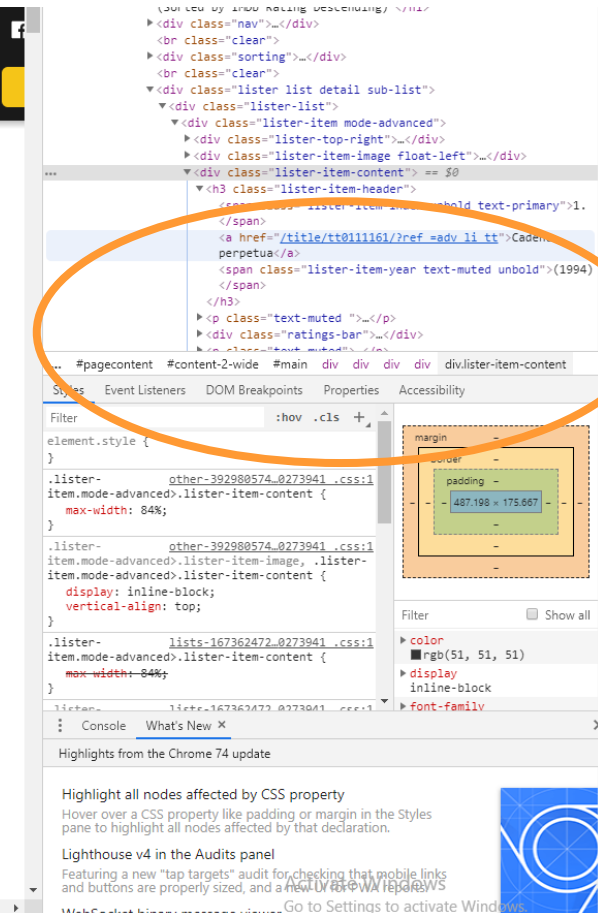
13 | 175 min | Crime, Drama

★ 9,2

☆ [Rate this](#)

100 Metascore

The aging patriarch of an organized crime dynasty transfers control of his clandestine empire to his reluctant son.





Zoom in.

```
<div class="list-item-image float-left" /.../div/
▼<div class="list-item-content"> == $0
  ▼<h3 class="list-item-header">
    <span class="list-item-index unbold text-primary">1.
    </span>
    <a href="/title/tt0111161/?ref=adv_li_tt">Cadena
    perpetua</a>
```

Para confeccionar la primera columna de nuestra base tenemos que levantar los primeros 50 títulos.

Como cada película tiene el mismo formato podemos llamar a **<h3 class=list-item-content>** que se repite para cada elemento (porque es lo que le da formato).

También sabemos que el contenido que buscamos esta entre **<a> **.



Caso de estudio - IMDB



Para armar el script primero nos fijamos si tenemos todas las librerías instaladas con:

```
1 install.packages("robotstxt")
2 install.packages("rvest")
3 install.packages("selectr")
4 install.packages("xml2")
5 install.packages("dplyr")
6 install.packages("stringr")
7 install.packages("forcats")
8 install.packages("magrittr")
9 install.packages("tidyr")
10 install.packages("ggplot2")
11 install.packages("lubridate")
12 install.packages("tibble")
13 install.packages("purrr")
```

-> Luego las cargamos:

```
15 library(robotstxt)
16 library(rvest)
17 library(selectr)
18 library(xml2)
19 library(dplyr)
20 library(stringr)
21 library(forcats)
22 library(magrittr)
23 library(tidyr)
24 library(ggplot2)
25 library(lubridate)
26 library(tibble)
27 library(purrr)
28
```




Caso de estudio - IMDB



Siempre el primer paso es pedir acceso a robots.txt:

```
30 paths_allowed(  
31     paths = c("https://www.imdb.com/search/title?groups=top_250&sort=user_rating")  
32 )
```

Esta es una función de la librería **robotstxt** que devuelve un booleano. TRUE si tenemos acceso a la información del link. FALSE caso contrario.

```
35 imdb <- read_html("https://www.imdb.com/search/title?groups=top_250&sort=user_rating")  
36
```

Leemos el html y lo guardamos en una variable. Acá bajamos toda la información que necesitamos. Luego, lo que se hace es *parsing* y procesamiento.

Queremos minimizar la cantidad de llamadas para evitar bloqueo de IP.

Si corremos esta variable vamos a tener una copia fiel del html que vemos con el explorador, **lamentablemente ilegible para humanos.**



Caso de estudio - IMDB



La variable `imdb` representa el resultado de una función `read_html()`. Podemos pasarle argumentos con el operador `%>%` (opera como un decorador de Python, nos permite componer funciones).

```
45  imdb %>%  
46    html_nodes(".lister-item-content h3 a") %>%  
47    html_text() -> movie_title
```

Hmtl_nodes es una función de *parsing*, le estamos pasando la clase: `lister-item-content`, el número de header: `h3`, y la tag del contenido: `a` (la función las espera en este orden).

Luego aplicamos el **operador** `%>%` de nuevo para decirle que solo queremos el texto del contenido, eso se lo asignamos (`->`) a una variable `movie_title`.



Caso de estudio - IMDB



Revisando la siguiente columna a llenar encontramos pocas diferencias.

```
▼<p class="text-muted ">  
  <span class="certificate">13</span>  
  <span class="ghost">|</span>  
  <span class="runtime">142 min</span> == $0  
  <span class="ghost">|</span>  
  <span class="genre">  
    Drama          </span>  
</p>
```

Ahora el separador es p.

La clase principal es la misma *lister-item-content*.

Pero tiene una clase adicional, *runtime*.

Además el contenido viene en *string* y yo lo quiero en *numeric*.



Caso de estudio - IMDB



El código es:

```
imdb %>%  
  html_nodes(".lister-item-content p .runtime") %>%  
  html_text() %>%  
  str_split(" ") %>%  
  map_chr(1) %>%  
  as.numeric() -> movie_runtime
```

Componemos dos funciones adicionales. La primera es *str_split()* convierte el texto en un vector de *strings*, *map_chr(1)* toma este vector y devuelve los elementos de este vector que se pueden convertir en el tipo del argumento pasado, *as.numeric* convierte al vector en números sin problemas. Luego se asigna a una variable.

Año, rating, y genero funcionan de la misma manera (**ver script**).



Caso de estudio - IMDB



La cantidad de votos es un atributo escondido (en el sentido de que no esta visible a simple vista en la página, y por lo tanto no tiene formateo).

Pero sabemos que si votamos el conteo debe cambiar. De hecho se posicionamos el cursor por encima de la opción de valorar una película vemos que la clase que le da formato tiene varias subclases.

Una pequeña búsqueda y la encontramos:

```
<meta itemprop="ratingValue" content="8.9">
<meta itemprop="bestRating" content="10">
<meta itemprop="ratingCount" content="1636445">
<span class="rating-bg">&nbsp;</span>
<span class="rating-imdb " style="width: 124.6px">
&nbsp;</span>
```



Caso de estudio - IMDB



Aquí la sintaxis cambia un poco, como hay varias subclases anidadas podemos pedir directamente la palabra clave `"ratingcount"`, y que el nodo la busque a través de un `xpath`.

La sintaxis para esto es `xpath = '//meta[@itemprop="ratingCount"]'` según la librería `xml2`.

```
<meta itemprop="ratingValue" content="8.9">
<meta itemprop="bestRating" content="10">
<meta itemprop="ratingCount" content="1636445">
<span class="rating-bg">&nbsp;</span>
<span class="rating-imdb" style="width: 124.6px">
&nbsp;</span>
```

Tenemos un código similar al anterior

```
imdb %>%
  html_nodes(xpath = '//meta[@itemprop="ratingCount"]') %>%
  html_attr('content') %>%
  as.numeric() -> movie_votes
```




Caso de estudio - IMDB



Revenue es similar a *rating count* en que necesita un *xpath*, pero además necesita formateo.

```
104
105 imdb %>%
106   html_nodes(xpath = '//span[@name="nv"]') %>%
107   html_text() %>%
108   str_extract(pattern = "^\\$.*") %>%
109   na.omit() %>%
110   as.character() %>%
111   append(values = NA, after = 30) %>%
112   append(values = NA, after = 46) %>%
113   str_sub(start = 2, end = nchar(.) - 1) %>%
114   as.numeric() -> movie_revenue
115
116 movie_revenue
117
```



Caso de estudio - IMDB



Ponemos todo en un dataframe y terminamos:

```
118 # junto todo
119 top_50 <- tibble(title = movie_title, release = movie_year,
120                  `runtime (mins)` = movie_runtime, genre = movie_genre, rating = movie_rating,
121                  votes = movie_votes, `revenue ($ millions)` = movie_revenue)
122
123 top_50
```





Conclusión



La intención de este tutorial fue mostrar de manera sencilla cómo hacer web scraping. Se consideró un caso simple, y podríamos decir que ideal.

No se mostraron todos los problemas o retos que se pueden presentar al hacer esta tarea.

Las **ventajas** que observo son: R es **gratuito**, fácil de usar y tiene bastante documentación. Toda generada por la comunidad. La **flexibilidad** de extraer virtualmente todo lo que se desee.

Las **desventajas**, esencialmente una: programación avanzada.

Si se dese hacer la extracción de datos, se requiere enfrentar a diversas situaciones, por lo que el nivel de programación se especializa:

- Paginación
- Valores nulos
- Clases CSS condicionadas
- Scripts
- Obtención de todas las urls del sitio
- Clasificación por categoría
- Limpieza e integración de datos
- Exportación de datos



Ejercicio propuesto



1. Armar un df con los premios nobel histórico a partir de: https://en.wikipedia.org/wiki/List_of_Nobel_laureates

¿Se animan?



**Nuevo
Espacio**



¿Dudas?

Gracias



**“Lo importante es no dejar de hacerse
preguntas”**

Albert Einstein

martinmasci@economicas.uba.ar

rdelrosso@economicas.uba.ar

