Curso Webscrapping

Michel Alves

18/03/2022

Unidade 2

Extraindo dados

A primeira etapa ao realizar o webscrapping é obter o HTML. Considere a seguinte URL:

```
library("xml2")
library("rvest")

url <- "https://www.imdb.com/search/title/?count=10&release_date=2021,2021&title_type=feature"

pagina <- read_html(url)
```

Existem várias formas de se extrair as informações a partir do HTML. O primeiro passo é localizar os nós desejados usando a função `html_element()` ou `html_elements()`.

```
pagina |> html_element("h3")
pagina |> html_elements("h3")
```

Texto

Para extrair texto, usa-se a função `html_text()` ou `html_text2()`. Para mostrar o uso dessas funções, iremos obter os números relativos ao rank do filme.

```
pagina |> html_elements(xpath = "//h3") |> html_text2()
```

Para uma descrição mais completa sobre a sintaxe do xpath, clique aqui.

Uma ferramenta que nos ajuda a analisar o seletor xpath que estamos desenvolvendo está disponível aqui

As duas barras no xpath `"//h3"` seleciona todos os elementos `h3` no HTML, não importando sua posição no HTML.

Para obter somente o número correspondente ao rank de cada filme, temos que especificar isso no xpath.

```
pagina |> html_elements(xpath = "//span") |> html_text2()
pagina |> html_elements(xpath = "//h3/span") |> html_text2()
pagina |> html_elements(xpath = "//h3/span[1]") |> html_text2()
rank <- as.numeric(pagina |> html_elements(xpath = "//h3/span[1]") |> html_text2())
```

Outra opção para os ranks é mostrada a seguir:

```
pagina |> html_elements(xpath = "//span[@class='lister-item-index unbold text-primary']") |> html_text2()
```

Agora vamos obter os títulos dos filmes e seus respectivos anos de lançamento.

```
pagina |> html_elements(xpath = "//h3/a") |> html_text2()
nomes <- pagina |> html_elements(xpath = "//h3/a") |> html_text2()

ano <- pagina |> html_elements(xpath = "//h3/span[2]") |> html_text2()
ano <- as.numeric(unlist(regmatches(ano, gregexpr("[[:digit:]]+", ano))))
```

Agora vamos obter o(s) gênero(s) e duração dos filmes.

```
pagina |> html_elements(xpath = "//span[@class='genre']") |> html_text2()
pagina |> html_elements(xpath = "//*[@class='genre']") |> html_text2()
generos <- pagina |> html_elements(xpath = "//*[@class='genre']") |> html_text2()
generos <- strsplit(generos, ", ")

pagina |> html_elements(xpath = "//*[@class='runtime']") |> html_text2()
duracao <- pagina |> html_elements(xpath = "//*[@class='runtime']") |> html_text2()
duracao <- as.numeric(unlist(regmatches(duracao, gregexpr("[[:digit:]]+", duracao))))
```

E se quisermos obter a duração de um filme específico? Suponha que desajamos obter a duração do filme Encanto.

```
pagina |> html_element(xpath = "//h3/a[contains(text(), 'Encanto')]") |> html_text2()
pagina |> html_element(xpath = "//h3/a[contains(text(), 'Encanto')]/..") |> html_text2()
pagina |> html_element(xpath = "//h3/a[contains(text(), 'Encanto')]/../..") |> html_text2()
pagina |> html_element(xpath = "//h3/a[contains(text(), 'Encanto')]/../../*/*[@class='runtime']") |> html_text2(
```

Atributos

Agora vamos obter o link para cada um dos filmes

```
pagina |> html_elements(xpath = "//h3/a") |> html_attr('href')
links <- pagina |> html_elements(xpath = "//h3/a") |> html_attr('href')
```

Finalmente vamos juntar as informações em uma tabela.

```
filmes <- cbind(rank, ano, nomes, duracao, generos, links)
```

Desafio

Acesse a seguinte URL:

```
url <- "https://www.imdb.com/search/title/?count=20&title_type=feature&release_date=2015-12-31,2021-01-01&sort=u

pagina <- read_html(url)
```

Coloque em uma tabela as notas, os nomes e as durações de cada filme. Observe que nem todos os filmes apresentam a duração.

Tabelas

Para extrair uma tabela de um HTML, basta usar a função `html_table()`.

```
url <- "https://pt.wikipedia.org/wiki/Lista_de_prefeitos_de_Belo_Horizonte"
pagina <- read_html(url)

pagina |> html_table()
pagina |> html_element(xpath = "//table[@class='wikitable']") |> html_table(header = FALSE, trim = FALSE)
```

Arquivos

Para importar um arquivo, primeiro deve-se obter a URL onde o mesmo está disponível. Para isso, pode-se usar a função `html_attr()`.

```
library("rio")
url <- "http://fjp.mg.gov.br/produto-interno-bruto-pib-de-minas-gerais/"
pagina <- read_html(url)

pagina |> html_elements(xpath = "//*[contains(text(), 'PIB anual')]/../../*/*/*/strong") |> html_text2()

link <- pagina |> html_elements(xpath = "//*[contains(text(), 'Bases de dados')]/../../*/*/li[2]/a") |> html_att
arquivo <- rio::import(link)
```

Exercício

Acesse a seguinte página do DATASUS e obter as opções disponíveis para

- Linha
- Coluna
- Conteúdo
- Períodos disponíveis
- Idade da mãe
- Sexo

```
url <- "http://tabnet.datasus.gov.br/cgi/deftohtm.exe?sinasc/cnv/nvmg.def"
pagina <- read_html(url)
```

Em seguida, obtenha os valores para as seguintes seleções:

- Linha: Município
- Coluna: Ano do nascimento
- Conteúdo: Nascim p/ ocorrência
- Períodos disponíveis: 2019
- Idade da mãe: 20 a 24 anos
- Sexo: Masc

Referências

https://www.w3schools.com/xml/xpath_syntax.asp https://www.analyticsvidhya.com/blog/2017/03/beginners-guide-on-web-scraping-in-r-using-rvest-with-hands-on-knowledge/ https://www.scrapingbee.com/blog/web-scraping-r/ http://www.macoratti.net/vb_xpath.htm