

Acessando Dados da Web em R

Raspagem de Dados em Webpages

Tiago Ventura | venturat@umd.edu

University of Maryland, College Park

Raspagem de Dados em Webpages

Uma quantidade crescente de dados está disponível na web:

- Discursos, frases, programas de governo, imagens,
- Dados de mídia social, artigos de jornal, press releases
- Leis, arquivos históricos, informação geográfica.

Esses conjuntos de dados geralmente são fornecidos em um formato **não estruturado**.

Raspagem de Dados: Extrair dados da internet e organizá-los para análise.

Há duas formas principais de acessar dados na internet.

1. **Raspar dados em websites:** coleta informação diretamente do site, da parte que qualquer pessoa visualiza. É como se você pudesse se multiplicar por mil, e coletar manualmente.
 - Em R: Pacote `rvest`
2. **Acessar APIs** (Interface de Programação de Aplicações): acessar um canal por trás da webpage por onde dados são gerados e compartilhados.
 - Em R: `httr` ou pacotes para APIs específicas

Acessar dados via APIs é mais seguro, prático, e rápido. Sempre opte pela segunda.

Quando e porquê raspar dados?

- Cópia e Cola consome tempo e aumenta as chances de R
- Raspagem pode ser amplificado para diferentes aplicações, é reproduzível, e facilita detectar erros.
- Construir um código consome tempo hoje, porém seu eu futuro lhe será sempre grato.

Rotina de Raspagem

1. Carregar o nome das páginas da internet
2. Fazer o download dos sites em formato HTML ou XML
3. Encontrar as partes do site que são do seu interesse (aqui dá bastante trabalho)
4. Limpar e processar os dados

Generalizar e Automatizar

5. Tentar com apenas um site todos os passos acima
6. Escrever um função em R para você repetir de forma automática a operação
7. Aplicar a função a sua lista de sites.

Ética em Raspagem de Dados

- Não atinja servidores com muita frequência
- Retarda o serviço para o que humanos fariam manualmente
- Encontre sites de origem confiáveis
- Não raspe durante o horário de pico
- Melhora a velocidade do seu código
- Use dados com responsabilidade (Como geralmente sendo acadêmicos)

Raspando Sites Estáticos

O que é um site? HTML e Javascript.

HTML É uma linguagem de texto estruturada por marcações (tags). O segredo para raspagem é basicamente identificar quais marcações você pretende coletar informação.

```
<html>
<head>
  <title> Michael Cohen's Email </title>
</head>
<body>
  <div id="payments">
    <h2>Second heading</h2>
    <p>Just <a href="http://www.google.com">google it!</a></p>
</body>
```


Municípios de Fronteira no Brasil.

W Lista de municípios fronteiriços x +

pt.wikipedia.org/wiki/Lista_de_municipios_frenteiricos_do_Brasil

Ver artigos principais: [Lista de municípios fronteiriços do Brasil por área](#), [Lista de municípios fronteiriços do Brasil por população](#), [Lista de municípios fronteiriços do Brasil por densidade demográfica](#), [Lista de municípios fronteiriços do Brasil por PIB](#) e [Lista de municípios fronteiriços do Brasil por PIB per capita](#)

Município	Estado	Área territorial	População (IBGE/2007)	Densidade demográfica (hab/km²)	PIB (IBGE/2005)	PIB per capita (R\$)	IDH/2000
1 – Açegua	Rio Grande do Sul	1.550	4.138	2,66	71.638.000	17.266	ni
2 – Acrelândia	Acre	1.575	11.520	7,31	114.350.000	9.986	0,680
3 – Alecrim	Rio Grande do Sul	315	7.357	23,35	44.373.000	5.944	0,743
4 – Almeirim	Pará	72.960	30.903	0,42	462.258.000	13.485	0,745
5 – Alta Floresta d'Oeste	Rondônia	7.067	23.857	3,37	186.812.000	6.525	0,715
6 – Alto Alegre	Roraima	25.567	14.386	0,56	115.786.000	5.239	0,662
7 – Alto Alegre dos Parecis	Rondônia	3.959	11.615	2,93	90.226.000	6.001	ni
8 – Amajari	Roraima	28.472	7.586	0,26	31.897	5.240.000	0,654
9 – Antônio João	Mato Grosso do Sul	1.144	8.350	7,29	39.989.000	5.067	0,702
10 – Aral Moreira	Mato Grosso do Sul	1.656	9.236	5,57	105.697.000	13.132	0,723
11 – Assis Brasil	Acre	2.876	5.351	1,86	30.298.000	5.984	0,670
12 – Atalaia do Norte	Amazonas	76.355	13.682	0,17	29.028.000	2.570	ni
13 – Bagé	Rio Grande do Sul	4.096	112.550	27,47	906.488.000	7.473	0,802

Type here to search

POR 5:35 PM 12/5/2019

Marcador Manualmente

W Lista de municípios fronteiriços

pt.wikipedia.org/wiki/Lista_de_municipios_frontereiros_do_Brasil

Apps CMSC470 Introdu... PDF Word Count | F... Machine Learning C... 28 Jupyter Notebo... Document Embedd... Joel Watson Academic Writing... Other bookmarks

Brasiléia Assis Brasil Cruzeiro do Sul Porto Xavier

Tipos de fronteiras [\[editar | editar código-fonte \]](#)

Bifronteiriços [\[editar | editar código-fonte \]](#)

Município	Estado	Países fronteiriços
1 – Atalaia do Norte	Amazonas	Peru
2 – Barra do Quaraí	Rio Grande do Sul	Uruguai e Argentina
3 – Assis Brasil	Acre	Bolívia e Peru
4 – Corumbá	Mato Grosso do Sul	Paraguai e Bolívia
5 – Foz do Iguaçu	Paraná	Argentina e Paraguai
6 – Laranjal do Jari	Amapá	Suriname e Guiana Francesa
7 – Oriximiná	Pará	Guiana e Suriname
8 – São Gabriel da Cachoeira	Amazonas	Colômbia e Venezuela
9 – Uiramutã	Roraima	Venezuela e Guiana
10 – Uruguaiana	Rio Grande do Sul	Uruguai e Argentina

Back Alt+Left Arrow
Forward Alt+Right Arrow
Reload Ctrl+R
Save as... Ctrl+S
Print... Ctrl+P
Cast...
Translate to English
View page source Ctrl+U
Inspect Ctrl+Shift+I

Type here to search

POR 10:29 PM
PTB 12/5/2019

W Lista de municípios fronteiriços x +

pt.wikipedia.org/wiki/Lista_de_municipios_frenteiricos_do_Brasil

Brasília Assis Brasil Cruzeiro do Sul Porto Xavier

Tipos de fronteiras [editar | editar código-fonte]

File Edit View History Bookmarks Tools Help

W Lista de municípios fronteiriços x https://pt.wikipedia.org/wiki/Lista_de_municipios_frenteiricos_do_Brasil

view-source:https://pt.wikipedia.org/wiki/Lista_de_municipios_frenteiricos_do_Brasil 133%

```

209 <td><a href="/wiki/Roraima" title="Roraima">Roraima</a></td>
210 <td><a href="/wiki/Venezuela" title="Venezuela">Venezuela</a> e <a href="/wiki/Guiana" title="Guiana">Guiana</a>
211 </td></tr>
212 <tr>
213 <td>10 - <a href="/wiki/Uruguai" title="Uruguai">Uruguai</a></td>
214 <td><a href="/wiki/Rio_Grande_do_Sul" title="Rio Grande do Sul">Rio Grande do Sul</a></td>
215 <td><a href="/wiki/Uruguai" title="Uruguai">Uruguai</a> e <a href="/wiki/Argentina" title="Argentina">Argentina</a>
216 </td></tr></tbody></table>
217 <h3><span class="mw-headline" id="Fronteira_simples">Fronteira simples</span><span class="mw-editsection"><span cla
218 <table class="wikitable sortable">
219
220 <tbody><tr bgcolor="#e6e6e6">
221 <th>Município
222 </th>
223 <th>Estado
224 </th>
225 <th>País fronteiriço
226 </th></tr>

```

Começando nosso percurso em R

```
# Instalar pacotes
install.packages("tidyverse")
install.packages("purrr")
install.packages("rvest")
install.packages("stringr")
install.packages("kableExtra")
install.packages("Rcurl")
```

```
# Ativar os pacotes
library("tidyverse")
library("purrr")
library("rvest")
library("stringr")
library("kableExtra")
```

```
## -- Attaching packages ----- tidyverse 1.2

## v ggplot2 3.2.1      v purrr 0.3.2
## v tibble 2.1.3       v dplyr 0.8.1
## v tidyr 0.8.3        v stringr 1.4.0
## v readr 1.3.1        v forcats 0.4.0

## -- Conflicts ----- tidyverse_conflicts
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()      masks stats::lag()

## Loading required package: xml2

##

## Attaching package: 'rvest'
```

Resumo Sobre Raspagem de HTML

Primeiro passo: Leia a página em R, e depois selecione a tag com sua informação.

- Como raspar? `rvest` em R:
- `read_html`: webpage -> HTML -> R
- `html_text`: Converte HTML -> texto
- `html_table`: Converte HTML -> tabelas
- `html_nodes`: Seleciona CSSS no HTML
- `html_attrs`: Seleciona atributos no HTML(links, imagens)
- Como encontrar as tags de seu interesse?
 - `selectorGadget`: extensão no Google Chrome e Firefox.

Passo 1: Encontre o Site.

```
# Crie o nome da sua url

minha_url <- "https://pt.wikipedia.org/wiki/
Lista_de_munic%C3%ADpios_frenteiri%C3%A7os_do_Brasil"

# Somente o nome
print(minha_url)
```


Passo 2: Raspe os Dados. Simples assim:

```
source <- read_html(minha_url)
```

```
# O que é esse objeto?
```

```
class(source) # XML=HTML
```

```
## [1] "xml_document" "xml_node"
```

Passo 3: Extrair Dados

```
# Como extrair a tabela?  
  
tabelas <- source %>%  
  html_table()
```

[[1]]

##

Município

Estado

Países front

1 1 - Atalaia do Norte

Amazonas

2 2 - Barra do Quaraí Rio Grande do Sul

Uruguai e Ar

3 3 - Assis Brasil

Acre

Bolívia

4 4 - Corumbá Mato Grosso do Sul

Paraguai e

5 5 - Foz do Iguaçu

Paraná

Argentina e P

6 6 - Laranjal do Jari

Amapá Suriname e Guiana F

##

[[2]]

##

Município

Estado País fronteiriço

1 1 - Aceguá Rio Grande do Sul

Uruguai

2 2 - Acrelândia Acre

Bolívi

3 3 - Alecrim Rio Grande do Sul

Argentin

4 4 - Almeirim Pará

Surinam

5 5 - Alta Floresta d'Oeste Rondônia

Bolívi

6 6 - Alto Alegre Roraima

Venezuel

##

[[3]]

##

Município

Estado Área territorial

Passo 4: Limpar e Salvar Nossos Primeiros Dados

```
tabela_limpa <- tabelas[[3]] %>%  
# Converter para um banco de dados mais bonito  
  as.tibble() %>%  
# Cria Duas novas Colunas  
  mutate(city = Município,  
          uf_name = Estado) %>%  
  select(city, uf_name) %>%  
# consertar o encoding  
  mutate(city = str_sub(city,5),  
         city = str_replace(city, pattern="- ", ""),  
         city = str_trim(city),  
         city_key = stringi::stri_trans_general(city,  
          "Latin-ASCII"),  
         city_key= str_replace_all(city_key, " ", ""),  
         city_key=str_to_lower(city_key))
```

```
## Warning: `as.tibble()` is deprecated, use `as_tibble()` (but
```

```
## This warning is displayed once per session.
```

```
## # A tibble: 4 x 3
##   city          uf_name          city_key
##   <chr>        <chr>          <chr>
## 1 Aceguá      Rio Grande do Sul acegua
## 2 Acrelândia Acre              acrelandia
## 3 Alecrim     Rio Grande do Sul alecrim
## 4 Almeirim    Pará              almeirim
```

Municípios com Eleições em 1985

```
# Selecione a URL

minha_url <- "https://pt.wikipedia.org/wiki/
              Elei%C3%A7%C3%B5es_municipais_no_Brasil_em_1985"

# Pega a página
page <- read_html(minha_url)
```



```
# Pegue as tabelas
```

```
out <- page %>%  
  html_nodes(".wikitable") %>%  
  html_table()
```

```
# Combinando as tabelas
```

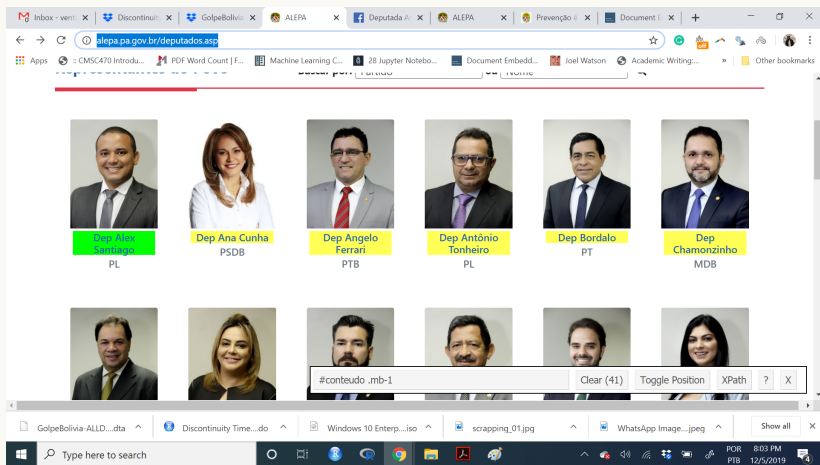
```
out_cap <- out[1]  
out_mun <- out[2:19]  
ot_municipios <- out_mun %>% bind_rows()
```


##	Bandeira	Município	Prefeito eleito	Partido
## 1	NA	Assis Brasil	José Vieira da Silva	PMDB
## 2	NA	Brasileia	Messias Ribeiro	PMDB
## 3	NA	Cruzeiro do Sul	João Barbosa	PMDB
## 4	NA	Feijó	Lívio Severiano	PMDB
## 5	NA	Mâncio Lima	Paulo Dene	PMDB

Usando o CSS Selector

0. Ative o Selector Gadget.
1. Clique no texto da pagina (Ficará em Amarelo).
2. Clique nas marcações equivocadas (Ficarão vermelho).
3. Faça isso até isolar a informação que você busca.

Exemplo do CSS Selector



Processando os nomes

```
# Coleta de todos os nomes
```

```
minha_url <- "https://www.alepa.pa.gov.br/deputados.asp"
```

```
nomes <- read_html(minha_url) %>%  
  html_nodes(css="#conteudo .mb-1") %>%  
  html_text()
```

```
## [1] "Dep Alex Santiago "      "Dep Ana Cunha "  
## [3] "Dep Angelo Ferrari "    "Dep Antônio Tonheiro "  
## [5] "Dep Bordalo "           "Dep Chamonzinho "  
## [7] "Dep Chicão "            "Dep Cilene Couto "  
## [9] "Dep Delegado Caveira "  "Dep Delegado Nilton Neves"
```

```
# Limpar os nomes
```

```
nomes_limpos <- nomes %>%  
  str_remove_all(., "Dep") %>%  
  str_to_lower() %>%  
  str_trim() %>%  
  str_replace_all(" ", "") %>%  
  str_replace("\\\\.", "") %>%  
  stringi::stri_trans_general("Latin-ASCII")
```

```
## [1] "https://www.alepa.pa.gov.br/alexsantiago"  
## [2] "https://www.alepa.pa.gov.br/anacunha"  
## [3] "https://www.alepa.pa.gov.br/angeloferrari"  
## [4] "https://www.alepa.pa.gov.br/antoniotonheiro"  
## [5] "https://www.alepa.pa.gov.br/bordalo"  
## [6] "https://www.alepa.pa.gov.br/chamonzinho"  
  
## [1] "https://www.alepa.pa.gov.br/alexsantiago"  
## [2] "https://www.alepa.pa.gov.br/anacunha"  
## [3] "https://www.alepa.pa.gov.br/angeloferrari"  
## [4] "https://www.alepa.pa.gov.br/antoniotonheiro"
```

Raspando um Caso

```
# url base
base_url <- "https://www.alepa.pa.gov.br/"

# Combina com o nome do deputado
url_dep <- paste0(base_url, nomes_limpos)
```

```
## [1] "https://www.alepa.pa.gov.br/alexsantiago"  
## [2] "https://www.alepa.pa.gov.br/anacunha"  
## [3] "https://www.alepa.pa.gov.br/angeloferrari"  
## [4] "https://www.alepa.pa.gov.br/antoniotonheiro"  
## [5] "https://www.alepa.pa.gov.br/carlosbordalo"  
## [6] "https://www.alepa.pa.gov.br/chamonzinho"
```


File Edit View History Bookmarks Tools Help

Home / Twitter x Inbox - venturati@umd.edu x ECPR-SC104/slides at mas... x Introduction to R workshop not... x Slack | Keng-Chi Chang, Ke... x ALEPA x +


https://www.alepa.pa.gov.br/wanderlan

Getting Started SelectorGadget How To Use Linux Scr... Joel Watson Pre-doctoral Fellowshi... Research on Elicitation... Exclusivo: as milícias a... API de dados - Portal ... Ipeadata

INÍCIO NOTÍCIAS MULTIMÍDIA ▾ PROPOSIÇÕES IMPRENSA ATIVIDADE PARLAMENTAR

Sua busca aqui... **BUSCAR**

Dep Wanderlan



dep.drwanderlan@alepa.pa.gov.br

Deputado Estadual - MDB

Entre em contato

Nome:

Assunto:

Email:

Mensagem:

Type here to search

POR 12:22 AM
PTB 12/14/2019

```
# url

url_dep1 <- url_dep[[1]]

#source
source <- url_dep1 %>% read_html()
```

Capturando Dados I

```
nome <- url_dep1 %>% str_remove("https://www.alepa.pa.gov.br/")

posicao <- source %>%
  html_nodes(css=".col-lg-8 .col-lg-8 .text-primary")
  html_text()

biografia <- source %>%
  html_nodes(css=".col-lg-8 .col-lg-8 p") %>%
  html_text() %>%
  paste0(., collapse = " ")

noticias <- source %>%
  html_nodes(css=".font-weight-bold a") %>%
  html_attr("href")
```

Capturando Dados II

```
twitter <- source %>%  
  html_nodes(css=".p-2 a") %>%  
    html_attr("href") %>%  
    str_subset("twitter")  
  
email <- source %>%  
  html_nodes(css=".mt-0 a") %>%  
    html_attr("href") %>%  
    str_subset("@") %>%  
    str_remove("mailto:")  
  
# Combina tudo como um banco de dados  
  
deputados <- data_frame(url_dep1, nome,  
                        posicao,  
                        biografia, noticias,  
                        twitter, email)
```

```
deputados %>% slice(1:5)
```

```
## # A tibble: 1 x 7
```

```
##   url_dep1      nome   posicao   biografia      noticias twit
```

```
##   <chr>        <chr> <chr>    <chr>        <chr>    <chr>
```

```
## 1 https://www~ alexs~ Deputado~ " Alex José de~ /notici~ http
```

Expandir escrevendo nossa própria função

- Finalizamos agora nossa primeira etapa.
- Precisamos expandir isso para todos os 41 deputados.
 - Escrever uma função em R, com base no nosso caso.
 - Aplicar a função a todos os nossos deputados.

O objetivo da função é evitar que você repita o seu código muitas vezes.

- O nome da função.
- Os inputs da função.
- O que a função faz.

Funções em R.

```
nome_da_funcao <- function(arg1,arg2){  
  
  # O que ela faz  
  
  out <- what the function does.  
  
  # Output  
  return(out) # output  
  
}
```


Exemplo

```
add_me <- function( argument1, argument2 ){  
  value <- argument1 + argument2  
  return(value)  
}
```

```
add_me(2,3)
```

```
## [1] 5
```

Nossa função para dados da Alepa

```
raspar_alepa <- function(url){  
  
  source <- url %>% read_html() # unica modificacao  
  nome <- url %>% str_remove("https://www.alepa.pa.gov.br/")  
  (...)  
  deputados <- tibble(url, nome,  
                        posicao,  
                        biografia,  
                        noticias, twitter, email)  
  
  # Output  
  return(deputados)  
  Sys.sleep(sample(5:10, 1))  
}
```

Testando a Função

```
raspar_alepa(url_dep[[20]])
```

```
## # A tibble: 1 x 7
```

```
##   url_dep1      nome   posicao   biografia      noticias tw
```

```
##   <chr>        <chr> <chr>    <chr>        <chr>    <c
```

```
## 1 https://www~ erald~ Deputado~ " Eraldo Jorge ~ /noticia~ <N
```

Aplicando a Função à uma Lista de Sites.

Há diversas formas de aplicar uma função à múltiplos objetos. Esse processo é tecnicamente chamado **programação funcional**.

1. Escrever um Loop: Ineficiente
2. Funções lapply: Inconsistentes
3. purrr : pacote do tidyverse.

```
# Aplicando nossa lista de links a uma função.
```

```
dados <- map(url_dep, raspar_alepa)
```

```
# Combine tudo
```

```
dados <- bind_rows(dados)
```

Dados Alepa

```
## # A tibble: 5 x 7
##   url_dep1      nome      posicao      biografia      noticias      twit
##   <chr>        <chr>    <chr>      <chr>          <chr>        <chr>
## 1 https://ww~ alexs~ Deputado~ " Alex José de~ /noticia~ http
## 2 https://ww~ anacu~ Deputada~ " Médica Ginec~ /noticia~ <NA>
## 3 https://ww~ angel~ Deputado~ " \n          ~ /noticia~ <NA>
## 4 https://ww~ anton~ Deputado~ " Antônio Tonh~ /noticia~ <NA>
## 5 https://ww~ carlo~ Deputado~ " Deputado est~ /noticia~ http
```

```
write.csv(dados, "deputados_para.csv")
```

O exercício de hoje vai ser o seguinte. É simples:

1. Abra o banco de dados que acabamos de criar
2. Veja os links para as notícias dos deputados.
3. Crie um código para acessar as cinco ultimas notícias destes deputados.
4. Colete somente o título das notícias. Se você quiser coletar o link, pode ir este passo além.