

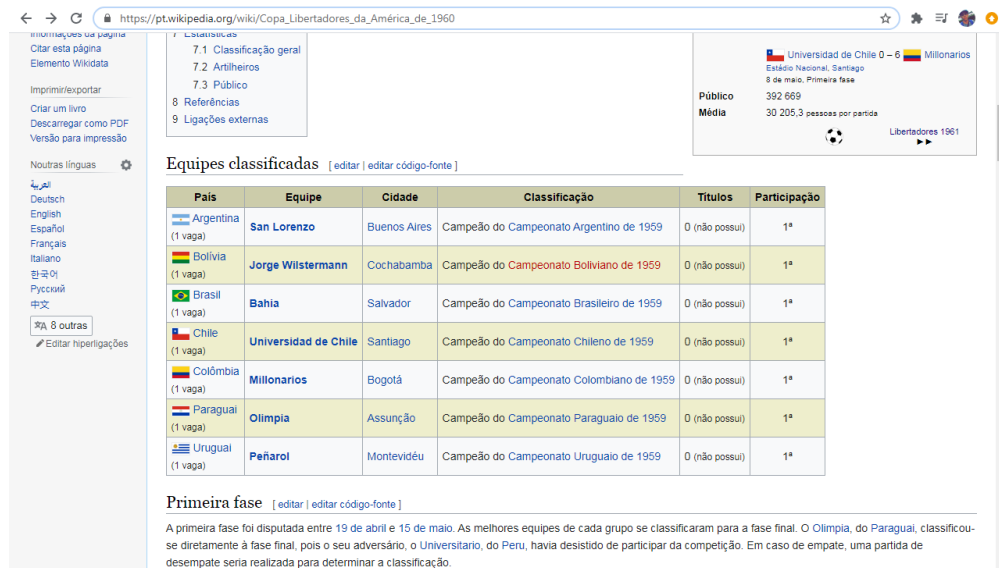
# Análise de dados sobre a Copa Libertadores da América

Para começar, são necessários três pacotes: `rvest` e `httr` para extrair os dados da web, e `tidyverse` para manipular os dados, uma vez que eles já estão em nosso workspace.

```
library(tidyverse)
library(rvest)
library(httr)
```

- Extraindo os dados para a primeira edição do torneio

Os dados sobre os times classificados para a Libertadores no ano de 1960 estão na url [https://pt.wikipedia.org/wiki/Copa\\_Libertadores\\_da\\_Am%C3%A9rica\\_de\\_1960](https://pt.wikipedia.org/wiki/Copa_Libertadores_da_Am%C3%A9rica_de_1960).



Pais	Equipe	Cidade	Classificação	Títulos	Participação
Argentina (1 vaga)	San Lorenzo	Buenos Aires	Campeão do Campeonato Argentino de 1959	0 (não possui)	1ª
Bolívia (1 vaga)	Jorge Wilstermann	Cochabamba	Campeão do Campeonato Boliviano de 1959	0 (não possui)	1ª
Brasil (1 vaga)	Bahia	Salvador	Campeão do Campeonato Brasileiro de 1959	0 (não possui)	1ª
Chile (1 vaga)	Universidad de Chile	Santiago	Campeão do Campeonato Chileno de 1959	0 (não possui)	1ª
Colômbia (1 vaga)	Millonarios	Bogotá	Campeão do Campeonato Colombiano de 1959	0 (não possui)	1ª
Paraguai (1 vaga)	Olimpia	Assunção	Campeão do Campeonato Paraguai de 1959	0 (não possui)	1ª
Uruguai (1 vaga)	Peñarol	Montevideu	Campeão do Campeonato Uruguaio de 1959	0 (não possui)	1ª

Figure 1: Página da competição no ano de 1960, onde a tabela com o título “Equipes Classificadas” é a que importamos para as análises.

Ao clicar com o botão direito do mouse e selecionar a opção Inspecionar, o XPath da tabela é dado por `//*[@id="mw-content-text"]/div[1]/table[2]`

Com essas informações, importamos a tabela com o comando abaixo.

```
url = ('https://pt.wikipedia.org/wiki/Copa_Libertadores_da_Am%C3%A9rica_de_1960')

resultado = url %>%
  read_html() %>%
  html_nodes(xpath = '//*[@id="mw-content-text"]/div[1]/table[2]') %>%
  html_table()
```

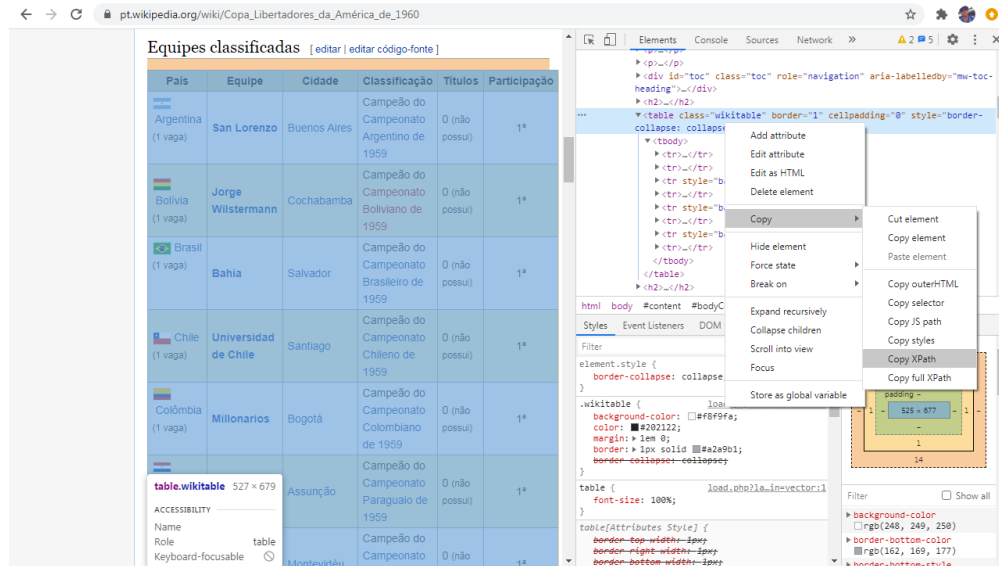


Figure 2: Copiano o XPath da tabela.

O valor que é armazenado na variável resultado é uma lista com um elemento, onde esse elemento é a tabela.

```
resultado[[1]][,1:3] # exibindo as 3 primeiras colunas
```

```
##          País          Equipe      Cidade
## 1 Argentina(1 vaga) San Lorenzo Buenos Aires
## 2 Bolívia(1 vaga)  Jorge Wilstermann Cochabamba
## 3 Brasil(1 vaga)   Bahia          Salvador
## 4 Chile(1 vaga)   Universidad de Chile Santiago
## 5 Colômbia(1 vaga) Millonarios    Bogotá
## 6 Paraguai(1 vaga) Olimpia        Assunção
## 7 Uruguai(1 vaga) Peñarol       Montevidéu
```

Para o objetivo desse projeto, só as colunas País e Equipe são necessárias. Logo, selecionamos essas duas colunas e armazenamos no objeto tabela. Por simplicidade, os nomes das colunas ficarão com todas as letras em minúsculo e sem acentuação. Além disso, adicionamos uma coluna referente ao ano em que ocorreu o torneio. Isso é importante, pois quando unirmos as tabelas de todas as edições do torneio, queremos saber qual foi o ano em que determinado clube se classificou para a libertadores.

```
tabela = resultado[[1]] %>% select(País, Equipe)
colnames(tabela) = c("pais", "equipe")
no_equipes = nrow(tabela) # número de equipes classificadas
edicao = rep(1960, no_equipes)
tabela = cbind(edicao, tabela) # adicionando a coluna edicao
```

Ao contruir os gráficos, é conveniente que os valores na coluna pais estejam sem os parenteses e a informação contida nos parenteses, ou seja, só o nome do país. Uma solução, retirada do stackoverflow, é a seguinte:

```
tabela$pais = gsub("\\s*\\([^\s\\)]+\\)", "", as.character(tabela$pais))
tabela
```

##	edicao	pais	equipe
## 1	1960	Argentina	San Lorenzo
## 2	1960	Bolívia	Jorge Wilstermann
## 3	1960	Brasil	Bahia
## 4	1960	Chile	Universidad de Chile
## 5	1960	Colômbia	Millonarios
## 6	1960	Paraguai	Olimpia
## 7	1960	Uruguai	Peñarol

A tarefa agora é extrair as tabelas para todas as edições do torneio e deixar no formato acima.

Seria interessante podermos generalizar o algoritmo acima para extrair a tabela dos outros anos. No entanto, não seria produtivo analisar as páginas do Wikipedia para cada ano manualmente a fim de encontrar um padrão entre elas, pois temos mais de 40 páginas. Temos que encontrar uma maneira sistemática de fazer isso. Como usamos apenas duas informações para importar a tabela, url da página e XPath da tabela, precisamos encontrar uma forma de generalizar nosso código a partir desses dois elementos.

- Análise da url

Vamos usar a página do torneio no ano de 2020 para comparar com a página do ano de 1960.

1960: [https://pt.wikipedia.org/wiki/Copa\\_Libertadores\\_da\\_America\\_de\\_1960](https://pt.wikipedia.org/wiki/Copa_Libertadores_da_America_de_1960)

2020: [https://pt.wikipedia.org/wiki/Copa\\_Libertadores\\_da\\_America\\_de\\_2020](https://pt.wikipedia.org/wiki/Copa_Libertadores_da_America_de_2020)

Comparando as duas urls, o que difere são os 4 últimos dígitos, ao qual se referem ao ano em questão. Logo, supomos que a url para acessar a página do torneio de qualquer ano é [https://pt.wikipedia.org/wiki/Copa\\_Libertadores\\_da\\_America\\_de\\_xxxx](https://pt.wikipedia.org/wiki/Copa_Libertadores_da_America_de_xxxx), onde xxxx é substituído pelo ano que queremos analisar.

Com isso, vamos verificar se conseguimos conectar com as url's de cada ano. O código abaixo faz essa tarefa.

```
for (ano in 1960:2020) {
  url = paste0('https://pt.wikipedia.org/wiki/Copa_Libertadores_da_America_de_', ano)
  url %>% read_html()
}
```

O código não retornou nenhum erro, o que indica que todas as url's são válidas.

- Análise do XPath

A tabela dos times classificados em 1960 tem XPath `//*[@id="mw-content-text"]/div[1]/table[2]`. Para o ano de 1961, o mesmo Xpath. Com isso, a estratégia é verificar se a tabela para todos os anos tem o mesmo Xpath. Criamos um vetor vazio anos, onde armazenamos no vetor cada ano antes de importar a tabela.

```
anos = numeric(0)
for (ano in 1960:2020) {
  anos = c(anos, ano) # adicionando o ano no vetor anos
  url = paste0('https://pt.wikipedia.org/wiki/Copa_Libertadores_da_America_de_', ano)
  resultado = url %>% read_html() %>%
    html_nodes(xpath = '//*[@id="mw-content-text"]/div[1]/table[2]') %>%
    html_table()
}
```

O algoritmo retornou erro, o que significa que, para algum ano, o Xpath da tabela não é o mesmo que em 1960. Analisando o algoritmo, esse ano é o último elemento do vetor anos.

```
last(anos) # extraíndo último elemento
```

O algoritmo não conseguiu extrair a tabela para o ano de 1976. Antes de avaliar separadamente esse caso, vamos verificar os anos restantes.

```
for (ano in 1960:2020) {  
  url = paste0('https://pt.wikipedia.org/wiki/Copa_Libertadores_da_Am%C3%A9rica_de_', ano)  
  resultado = try(url %>% read_html() %>%  
    html_nodes(xpath = '//*[@id="mw-content-text"]/div[1]/table[2]') %>%  
    html_table())  
  if(inherits(resultado, "try-error")) {  
    anos = c(anos, ano) # adicionando o ano no vetor anos  
  }  
  # anos = c(anos, ano) # adicionando o ano no vetor anos  
}  
anos
```

O Xpath da tabela para o ano de 1976 é `//*[@id="mw-content-text"]/div[1]/table[3]`. Seguindo o mesmo raciocínio acima, vamos ver quais anos que também tem esse valor de Xpath.

```
aux = numeric(0)  
for (ano in anos) {  
  url = paste0('https://pt.wikipedia.org/wiki/Copa_Libertadores_da_Am%C3%A9rica_de_', ano)  
  resultado = try(url %>% read_html() %>%  
    html_nodes(xpath = '//*[@id="mw-content-text"]/div[1]/table[3]') %>%  
    html_table())  
  if(inherits(resultado, "try-error")) {  
    aux = c(aux, ano) # adicionando o ano no vetor anos  
  }  
  # anos = c(anos, ano) # adicionando o ano no vetor anos  
}  
aux
```

Sobraram dois anos: 1980 e 1993. Além disso, a mensagem de erro é diferente que os casos anteriores. Para entender melhor, analisamos a tabela do ano de 1980. Acontece que a tabela tem um formato diferente. Com isso, temos que adotar outra estratégia para importar essa tabela.

Com a tabela nesse formato, não conseguimos importar de uma vez para todos os países. Logo, importamos para cada país separadamente e agrupamos em uma única tabela. Além disso, importamos cada tabela no formato texto, pois nosso algoritmo não importa como tabela.

```
url = ('https://pt.wikipedia.org/wiki/Copa_Libertadores_da_Am%C3%A9rica_de_1980#Primeira_fase')  
  
resultado = url %>%  
  read_html() %>%  
  html_nodes(xpath = '//*[@id="mw-content-text"]/div[1]/table[3]') %>%  
  html_text()  
  
resultado[[1]]  
  
start <- gsub("\\n\\n", "~~", as.character(resultado[[1]]))  
  
dat <- map(start, function(x) {
```

```
tibble(text = unlist(str_split(x, pattern = "\\n"))) %>%  
  rowid_to_column(var = "line")  
})  
  
tabela_aux = as.data.frame(dat[[1]] %>% select(text))  
pais = rep(tabela_aux[1,1], nrow(tabela_aux))
```