Dashboards com R

htmlwidgets

curso-r

março de 2022

htmlwidgets

`htmlwidgets` são bibliotecas de visualização JavaScript encapsuladas em pacotes de R. Elas nos permitem usar diversas ferramentas JavaScript diretamente do R, adicionando algumas poucas linhas de código em nosso script.

Usando `htmlwidgets`, conseguimos construir tabelas, gráficos, mapas e muito outras visualizações interativas e naturalmente bonitas.

Vamos falar aqui dos seguintes pacotes:

- `{reactable}`, para tabelas
- `{plotly}`, para gráficos
- `{leaflet}`, para mapas

Clique aqui para acessar uma lista completa de todos os `htmlwidgets` disponíveis.

Tabelas com reactable

O pacote `reactable` nos permite criar tabelas interativas baseadas na biblioteca React Table. Para criar uma reactable no nosso app, precisaremos das funções `reactable()`, `reactableOutput()` e `renderReactable()`.

Clique aqui para acessar o tutorial completo do pacote `{reactable}`.

Exemplo de construção:

```
# ui
reactable::reactableOutput("tabela")

# server
ouput$tabela <- reactable::renderReactable({
  reactable::reactable(imdb)
})
```

Tabelas com reactable

A interatividade dos `htmlwidgets` não depende de uma sessão R rodando por trás. Você pode utilizá-los em qualquer documento `.html`.

```
reactable::reactable(
  mtcars, compact = TRUE, defaultPageSize = 4,
  striped = TRUE
)
```

	mpg	cyl	disp	hp	drat	wt
Mazda RX4	21	6	160	110	3.9	2.62
Mazda RX4 Wag	21	6	160	110	3.9	2.875
Datsun 710	22.8	4	108	93	3.85	2.32
Hornet 4 Drive	21.4	6	258	110	3.08	3.215

1–4 of 32 rows Previous **1** 2 3 4 5 … 8 Next

Pacotes alternativos para tabelas

A seguir, uma lista de pacotes/funções alternativos que trazem soluções para visualização de tabelas.

- `knitr::kable()`: não é um htmlwidget (não possui interatividade), mas é uma solução para formatar tabelas quando não precisamos que elas sejam interativas. Funciona em conjunto com o pacote `{kableExtra}`.

- `DT::datatable()`: outro htmlwidget para criar tabelas interativas. Funciona tal como o `reactable()`, mas um pouco mais burocrático para formatar as tabelas. Baseado na biblioteca JavaScript DataTables.

Tutoriais

- Tutorial kable e kableExtra

- Tutorial DT

Gráficos com plotly

O pacote `plotly` nos permite criar gráficos interativos baseados na biblioteca Plotly (construída em D3). Para criar um plotly no nosso app, precisaremos criar um plotly e das funções `plotlyOutput()` e `renderPlotly()`.

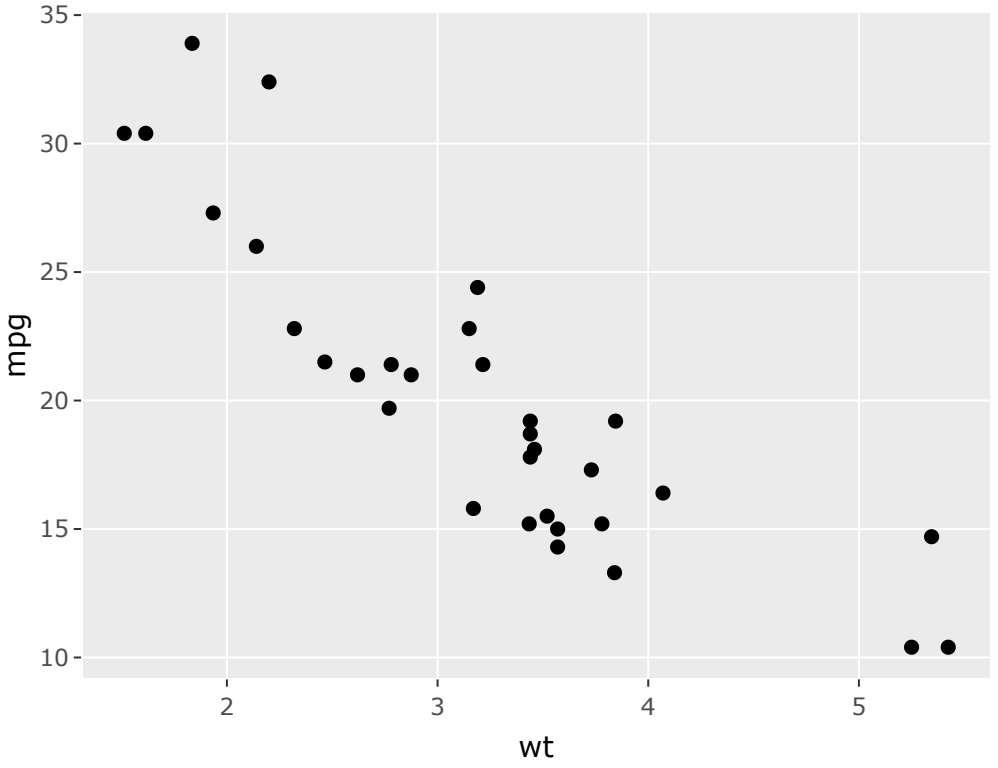
Embora seja possível criar um plotly do zero usando a função `plot_ly()`, um jeito muito eficiente de utilizar essa biblioteca é criar um ggplot e então utilizar a função `ggplotly()`. Veja o exemplo a seguir.

Tutoriais

- Tutorial plotly
- Interactive web-based data visualization with R, plotly, and shiny

Gráficos com plotly

```
library(ggplot2)
p <- ggplot(mtcars) +
  geom_point(aes(x = wt, y = mpg))
plotly::ggplotly(p, height = 400)
```



Gráficos com plotly

Para colocar um plotly no nosso app, basta utilizar as funções `plotlyOutput()` e `renderPlotly()`.

```
# ui
plotly::plotlyOutput("grafico")

# server
ouput$grafico <- plotly::renderPlotly({

  p <- ggplot(mtcars) +
    geom_point(aes(x = wt, y = mpg))

  plotly::ggplotly(p, height = 400)

})
```

Pacotes alternativos

A seguir, uma lista de pacotes/funções alternativos que trazem soluções para visualização de gráficos.

- `highcharter`: pacote gráfico baseado na biblioteca JavaScript Highcharts. A biblioteca Highcharts é gratuita apenas para fins educacionais e não lucrativos (exceto órgãos governamentais). Para outros usos, você pode precisar de uma licença.

- `echarts4r`: pacote gráfico baseado na biblioteca JavaScript `echarts`. Essa biblioteca é código aberto, podendo ser utilizada gratuitamente para qualquer propósito.

- Procure por pacotes para tipos específicos de gráficos na galeria de htmlwidgets.

Tutoriais

- Tutorial highcharter

- Tutorial echarts4r

- Documentação Highcharts

Mapas com leaflet

O pacote `{leaflet}` nos permite criar mapas interativos baseados na biblioteca JavaScript open-source Leaflet. Para colocar um mapa leaflet no nosso app, precisaremos criar um leaflet e das funções `leafletOutput()` e `renderLeaflet()`.

Para criar um mapa leaflet, utilizamos a função `leaflet::leaflet()` e diversas funções auxiliares para caracterizar nosso mapa. Um tutorial de como utilizar o `leaflet` se encontra aqui.

A seguir, mostramos um exemplo simples de como criar um mapa leaflet.

Mapas com leaflet

```
library(leaflet)

leaflet(height = 300) %>%
  addTiles() %>%
  addMarkers(
    lng = -46.6623969, lat = -23.5581664,
    popup = "A Curso-R morava aqui. Agora ela mora na internet
  )
```

Mapas com leaflet

Para colocarmos um leaflet no nosso app, basta utilizarmos as funções `leafletOutput()` e `renderLeaflet()`.

```
# ui
leaflet::leafletOutput("mapa")

# server
ouput$mapa <- leaflet::renderLeaflet({

  leaflet(height = 300) %>%
    addTiles() %>%  # Adiciona a camada gráfica do OpenStreetM
    addMarkers(
      lng = -46.6623969, lat = -23.5581664,
      popup = "A Curso-R mora aqui :)"
    )

})
```

Pacotes alternativos

- `ggplot2::geom_sf`: não é um htmlwidget (não possui interatividade), mas é uma boa solução para construção de mapas utilizando o framework do pacote `ggplot2()` em conjunto do pacote `sf`.
- `highcharter::hcmap()`: variação do `highcharter` para mapas, baseada na biblioteca JavaScript Highcharts.
- `{tmap}`: Pacote focado em mapas temáticos: Thematic Maps.

Tutoriais

- Documentação geom_sf()
- Construindo mapas com o highcharter
- Documentação Highmaps

Atividade

Vamos construir htmlwidgets no nosso Shinydashboard.

Ao RStudio:

Referências e material extra

htmlwidgets

- Galeria htmlwidgets

reactable

- A biblioteca React Table
- Tutorial reactable

DT

- Tutorial DT

plotly

- Tutorial plotly
- Interactive web-based data visualization with R, plotly, and shiny

Referências e material extra

highcharter/highcharts

- Tutorial highcharter
- Biblioteca Highcharts
- Galeria Highcharts
- Documentação Highcharts

leaflet

- Biblioteca Leaflet
- Tutorial Leaflet

highmaps

- Galeria Highmaps
- Documentação Highmaps

Referências e material extra

tmap

- Thematic Maps

Miscelânea

- Documentação geom_sf()
- Tutorial kable e kableExtra
- Biblioteca D3