INPUTS, OUTPUTS E REATIVIDADE

Prof. Walmes Zeviani

JUSTIÇA 4.0: INOVAÇÃO E EFETIVIDADE NA REALIZAÇÃO DA JUSTIÇA PARA TODOS
PROJETO DE EXECUÇÃO NACIONAL BRA/20/015

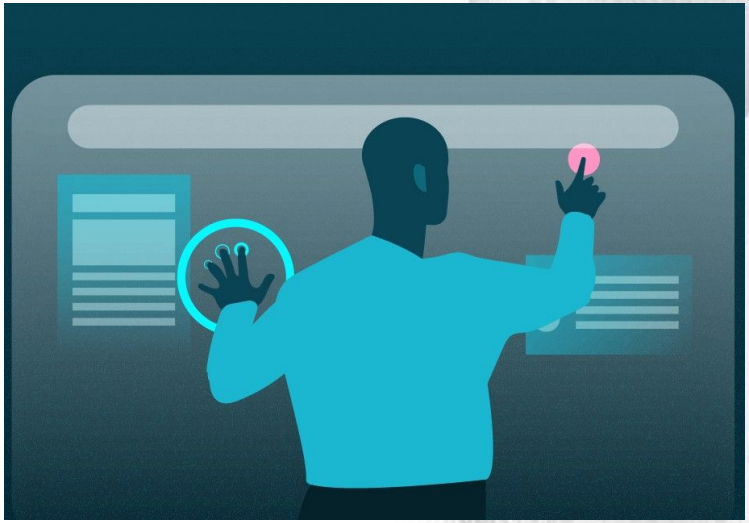
1.
Inputs

Widgets que o usuário interage

Inputs

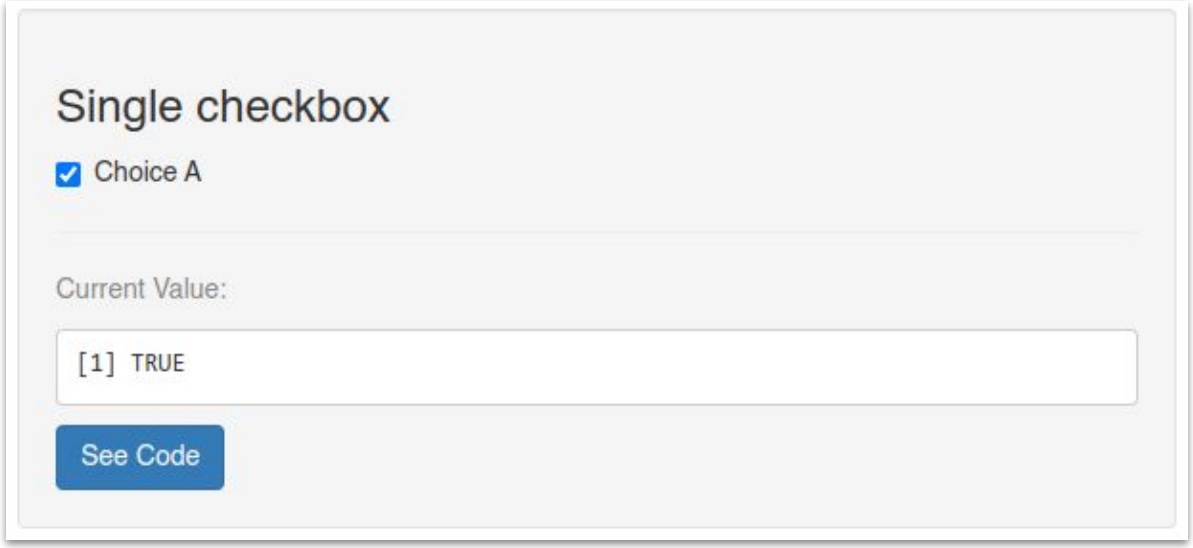
São elementos com os quais o usuário interage, controle e define variáveis na interface:

- Texto.
- Números.
- Datas.
- Filtros.
- Arquivos.



https://rockcontent.com/blog/increase-user-interaction/

Caixa de seleção

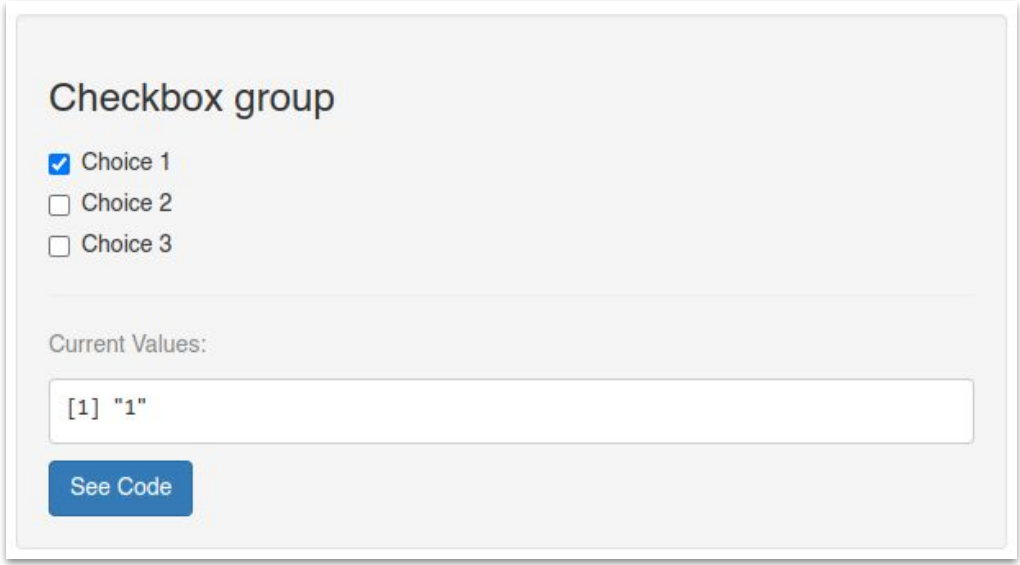


checkboxInput()
Retornam TRUE ou FALSE.

DICA

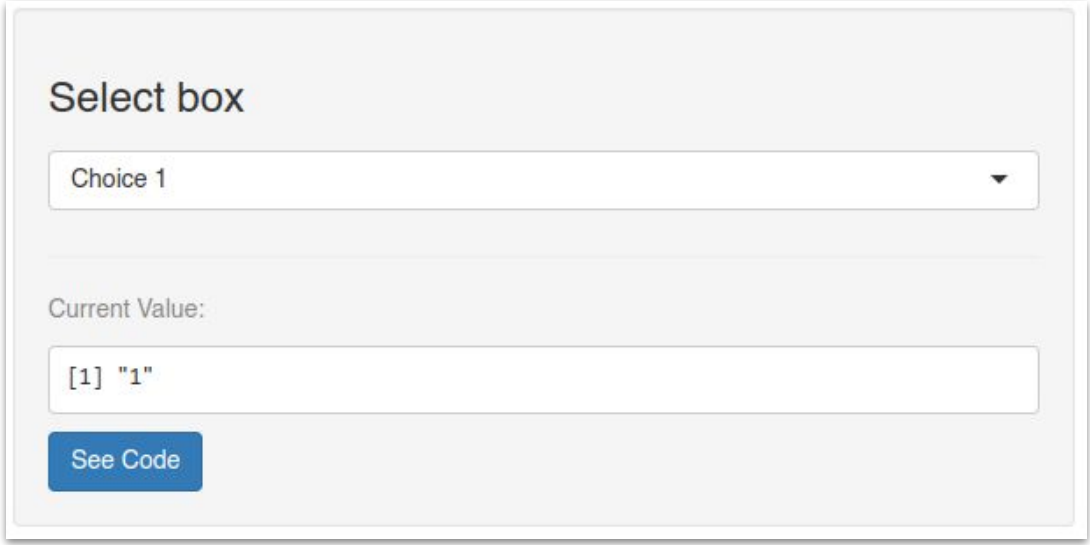
```
ls("package:shiny") |>
    grep(pattern = "checkbox",
         value = TRUE)
```

Grupo de caixas de seleção



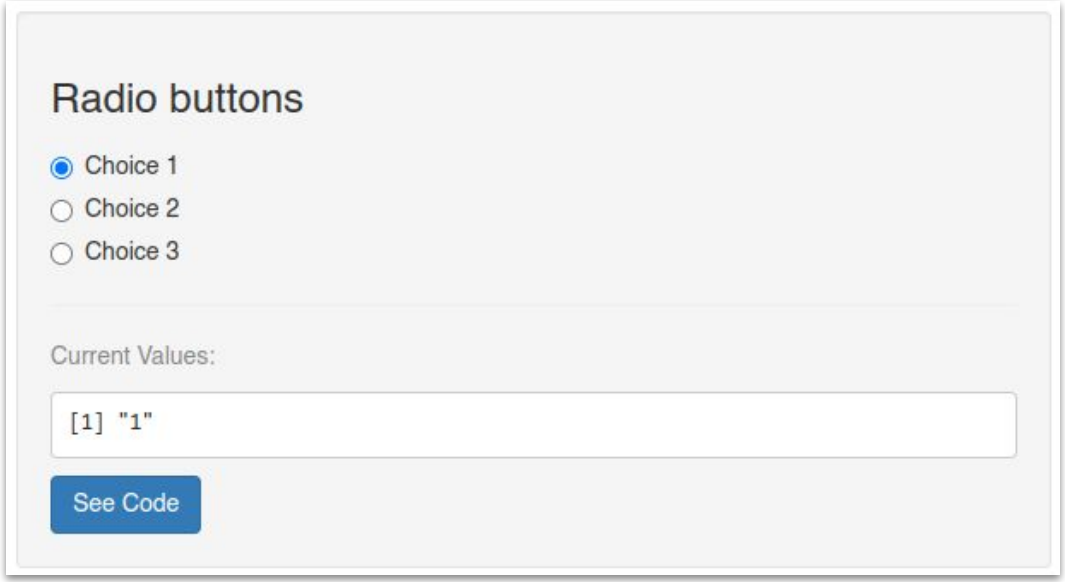
checkboxGroupInput()
Retornam as posições do TRUE.

Lista de seleção



selectInput() e **seletizeInput()**
Retornam o(s) elemento(s) escolhidos.

Botões de seleção



radioButtons()
Retorna o elemento selecionado.

Texto livre

textInput()
Retorna um character.

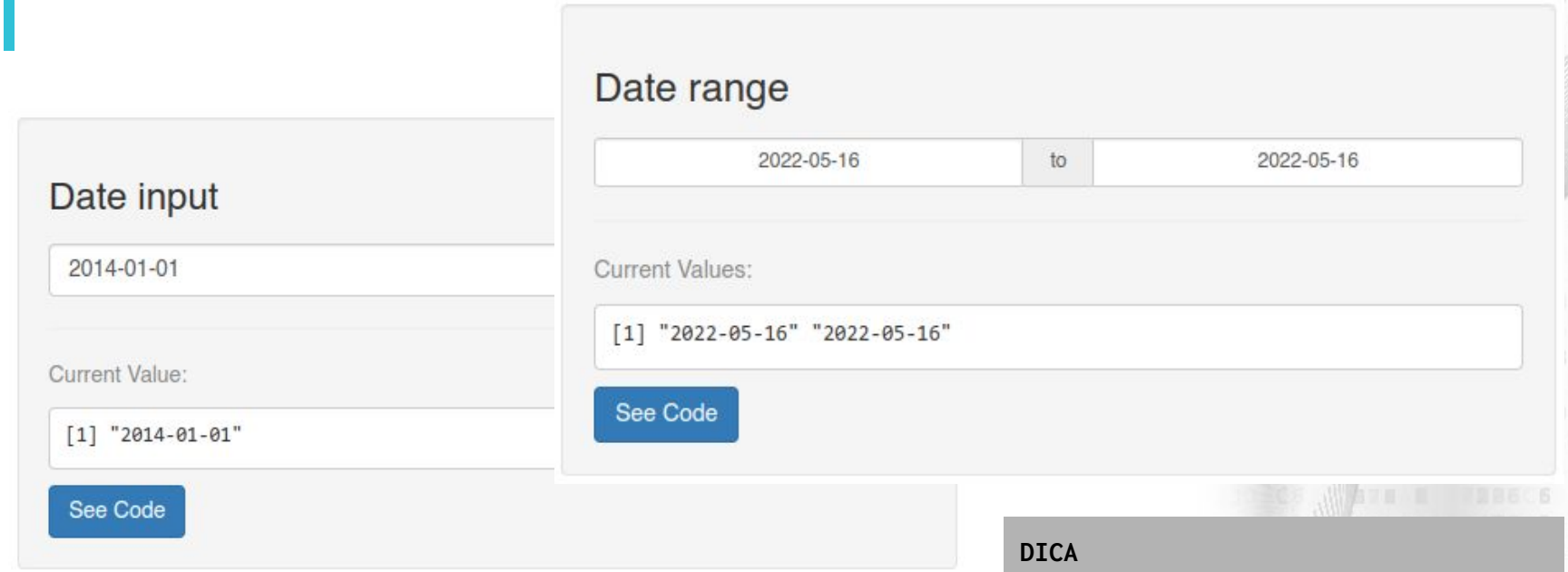
Entrada de valor numérico

numericInput()
Retorna valores numéricos.

Deslizadores numéricos

sliderInput()
Retorna valores numéricos.

Seletores de data



dateInput() e **dateRangeInput()**
Retorna characters no formato de data.

DICA

```
ls("package:shiny") |>
    grep(pattern = "date.*Input",
         value = TRUE)
```

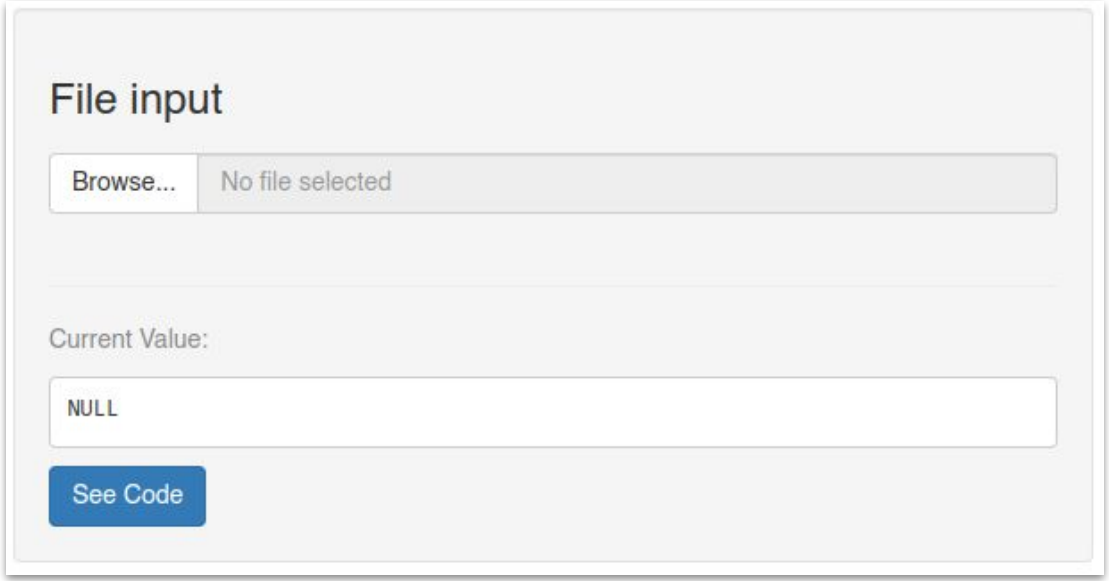
Botões de ação

actionButton(), **submitButton()**, etc
Retornam inteiros.

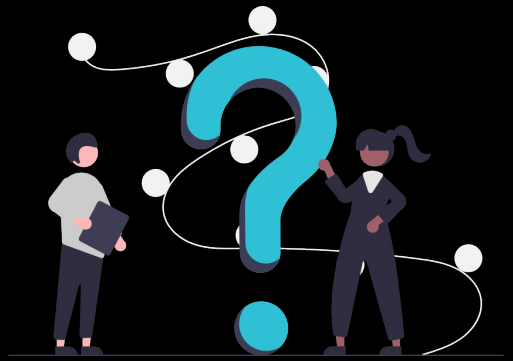
DICA

```
ls("package:shiny") |>
    grep(pattern = "button",
         value = TRUE)
```

Upload de arquivo



fileInput()
Retorna informações para o upload.



Questions?

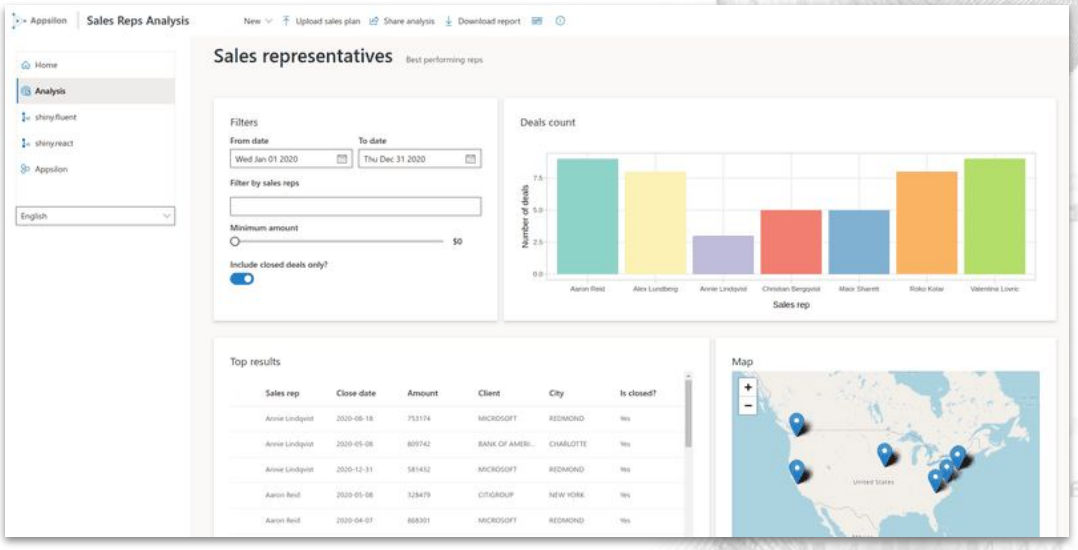
2.
Outputs

Widgets que o usuário examina

Outputs

São elementos que o seu usuário examina:

- Gráficos.
- Tabelas.
- Saídas de texto.
- Imagens.
- Arquivos.



https://appsilon.com/

Gráficos estáticos

`renderPlot()` → `plotOutput()`

Tabelas estáticas

`renderTable()` → `tableOutput()`

Category1	Info	Category2	Size	MoreStuff
a	Text info 1	o	12	More Stuff 1
a	Text info 2	o	19	More Stuff 2
a	Text info 3	p	29	More Stuff 3
a	Text info 4	o	53	More Stuff 4
a	Text info 5	s	73	More Stuff 5
a	Text info 6	o	17	More Stuff 6
a	Text info 7	r	81	More Stuff 7
a	Text info 8	s	3	More Stuff 8
a	Text info 9	r	21	More Stuff 9
a	Text info 10	s	5	More Stuff 10

Tabelas interativas

`renderDataTable()` → `dataTableOutput()`

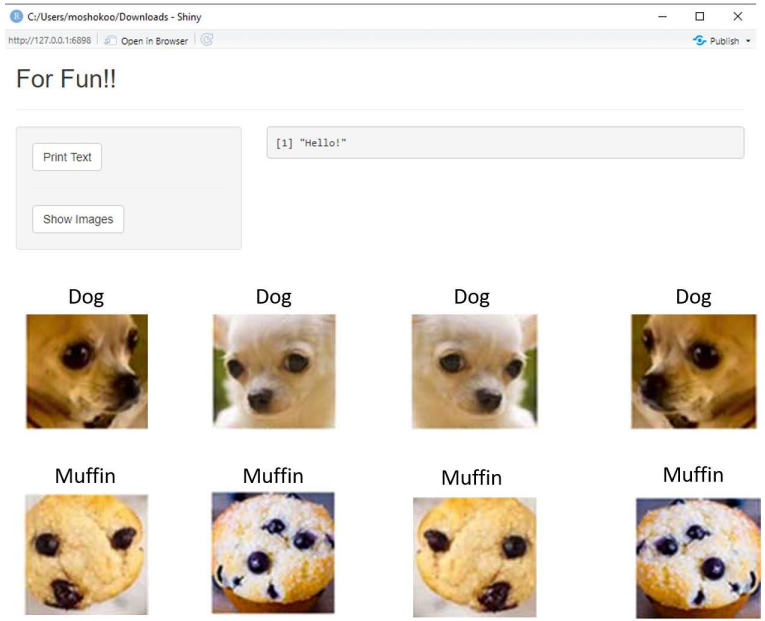
Show 10 entries Search:

	id	title	first_name	last_name	email	gender	ip_address	ssn
			A	Al	All		All	A
1	1	Mrs	Kyla	Doddemeede	kdoddemeede0@comsenz.com	Female	167.87.233.15	531-8
2	2	Mrs	Kermy	Ramage	kramage1@skyrock.com	Male	17.5.202.255	565-1
3	3	Mr	Vanessa	Gwinn	vgwinn2@imdb.com	Female	81.15.14.40	721-0
4	4	Mr	Mitzi	Crutchley	mcrutchley3@vinaora.com	Female	88.69.92.162	758-9
5	5	Rev	Correy	Folder	cfolder4@webs.com	Male	40.72.10.128	372-4
6	6	Ms	Brianna	Bes	bbes5@bluehost.com	Female	168.4.139.117	542-2
7	7	Dr	Dex	Retallick	dretallick6@mashable.com	Male	197.114.108.207	555-3
8	8	Honorable	Jennilee	Foxcroft		Female	117.62.196.148	765-2
9	9	Dr	Helena	Sagrott	hsagrott8@soundcloud.com	Female	7.51.0.104	108-6
10	10	Rev	Clemence	Ansill	cansill9@ebay.co.uk	Female	122.126.60.141	572-8

Showing 1 to 10 of 1,000 entries Previous 1 2 3 4 5 … 100 Next

Imagens

`renderImage()` → `imageOutput()`



Interessante para usar QR code.

Saída de texto

`renderPrint()` → `textOutput()`
`verbatimTextOutput()`
`htmlOutput()`

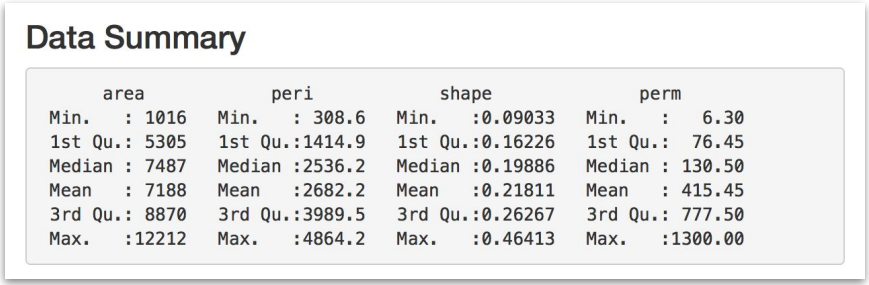
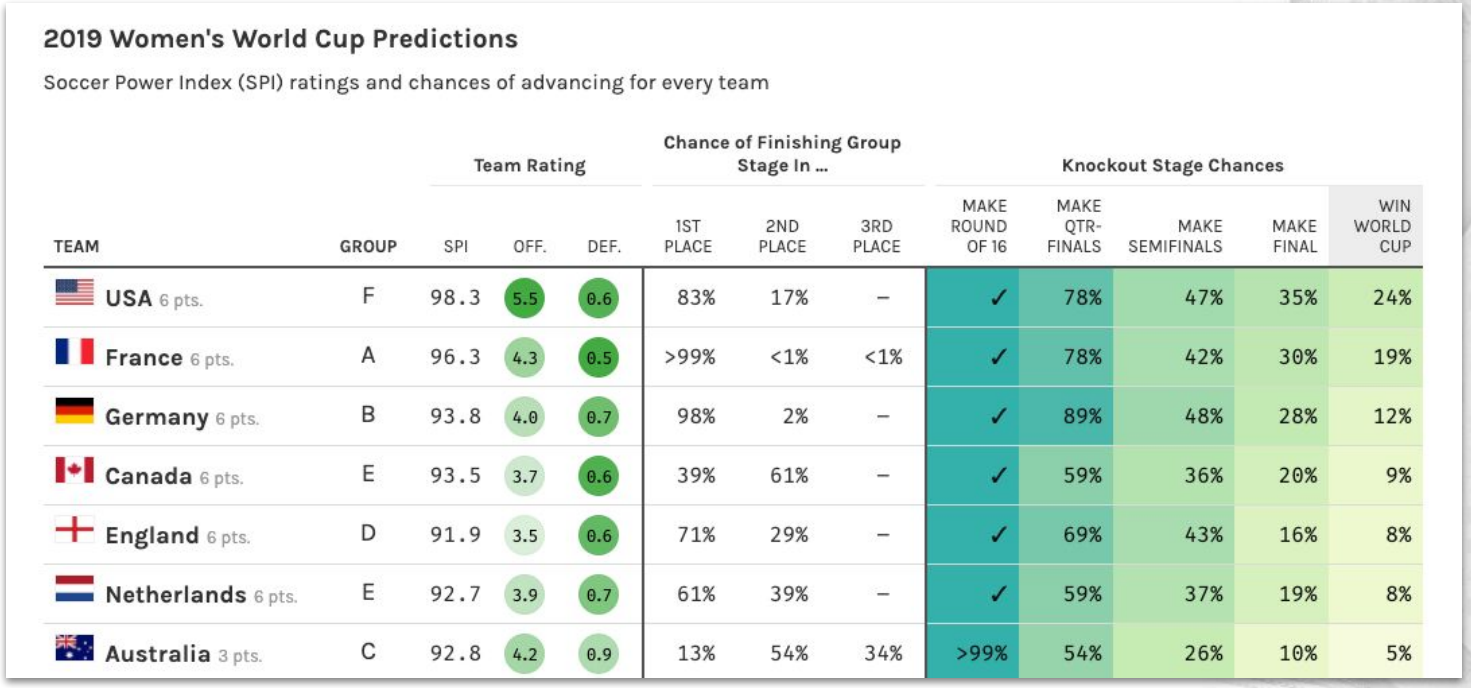


Tabela interativa -> {reactable}

`reactable::renderReactable()` ⟶ `reactable::reactableOutput()`



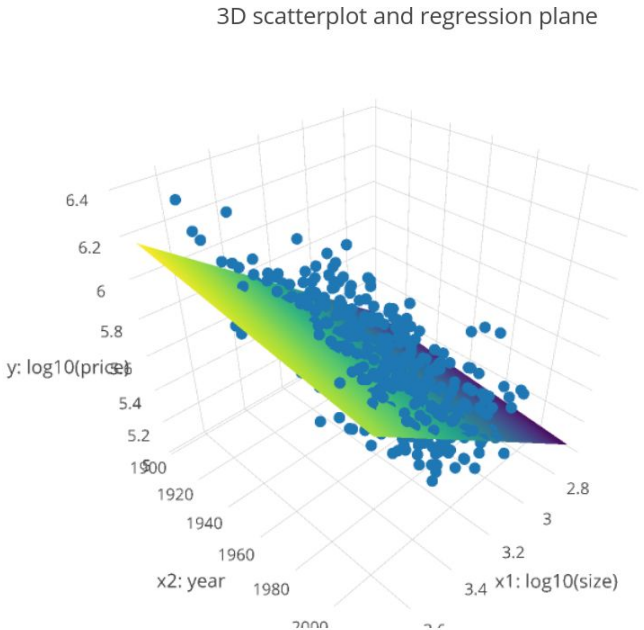
2019 Women's World Cup Predictions

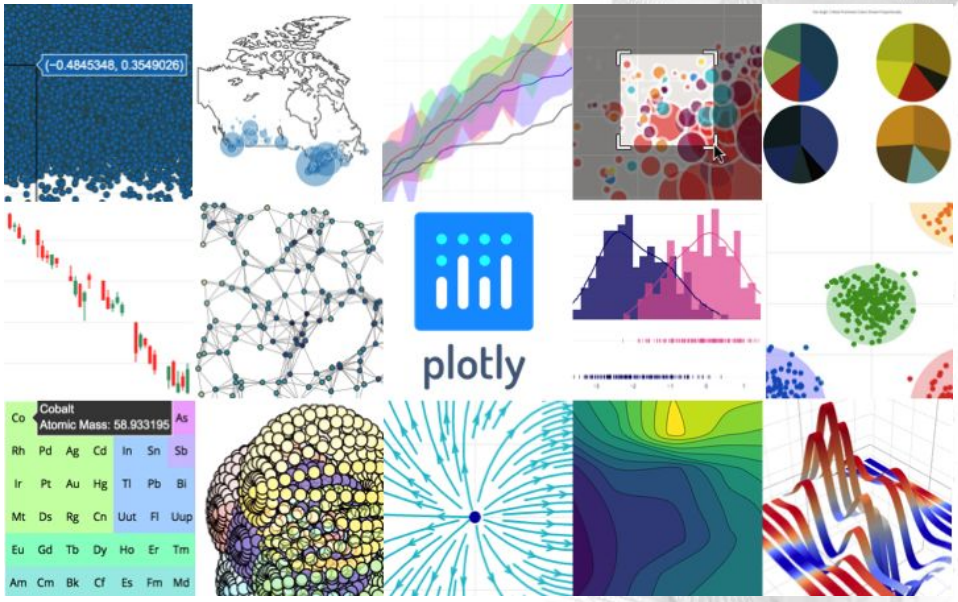
Soccer Power Index (SPI) ratings and chances of advancing for every team

		Team Rating			Chance of Finishing Group Stage In ...			Knockout Stage Chances				
TEAM	GROUP	SPI	OFF.	DEF.	1ST PLACE	2ND PLACE	3RD PLACE	MAKE ROUND OF 16	MAKE QTR-FINALS	MAKE SEMIFINALS	MAKE FINAL	WIN WORLD CUP
USA 6 pts.	F	98.3	5.5	0.6	83%	17%	–	✓	78%	47%	35%	24%
France 6 pts.	A	96.3	4.3	0.5	>99%	<1%	<1%	✓	78%	42%	30%	19%
Germany 6 pts.	B	93.8	4.0	0.7	98%	2%	–	✓	89%	48%	28%	12%
Canada 6 pts.	E	93.5	3.7	0.6	39%	61%	–	✓	59%	36%	20%	9%
England 6 pts.	D	91.9	3.5	0.6	71%	29%	–	✓	69%	43%	16%	8%
Netherlands 6 pts.	E	92.7	3.9	0.7	61%	39%	–	✓	59%	37%	19%	8%
Australia 3 pts.	C	92.8	4.2	0.9	13%	54%	34%	>99%	54%	26%	10%	5%

Gráficos interativos -> {plotly}

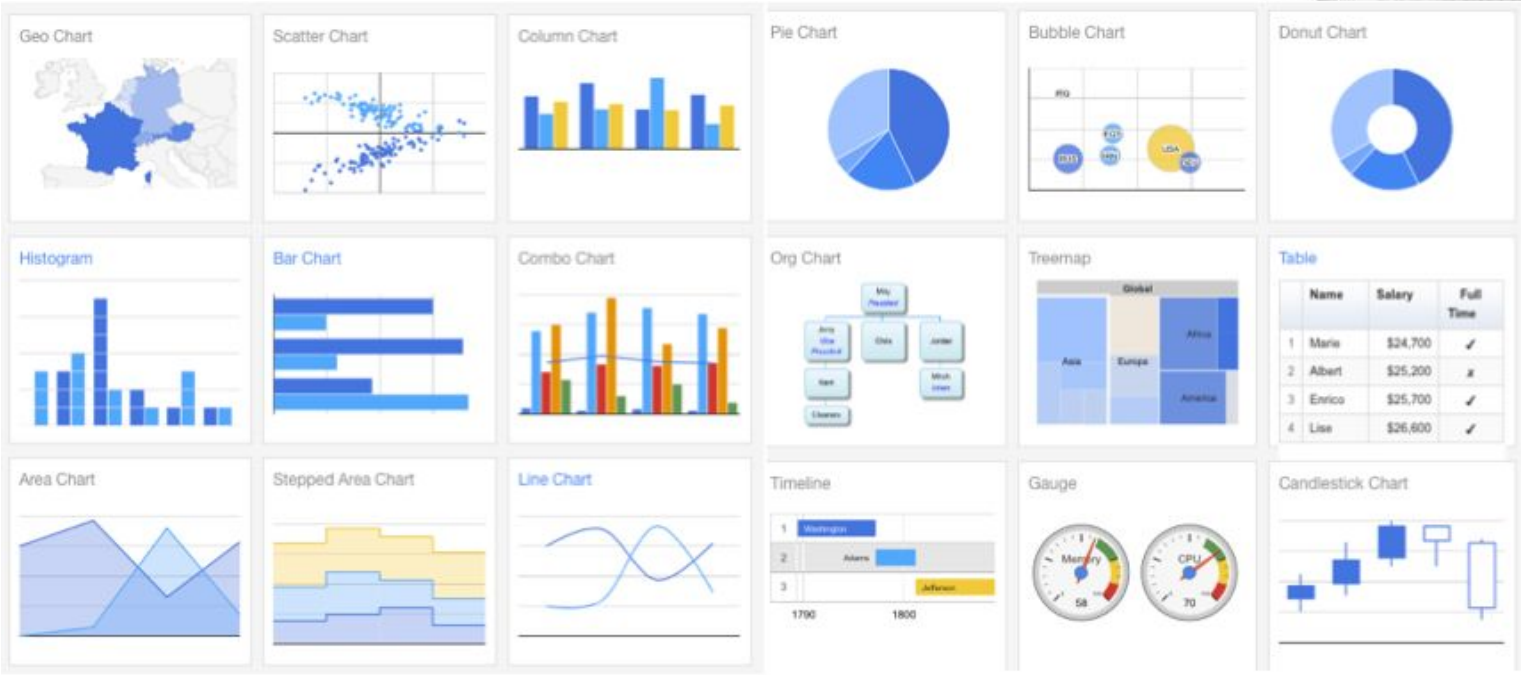
`plotly::renderPlotly()` ⟶ `plotly::plotlyOutput()`





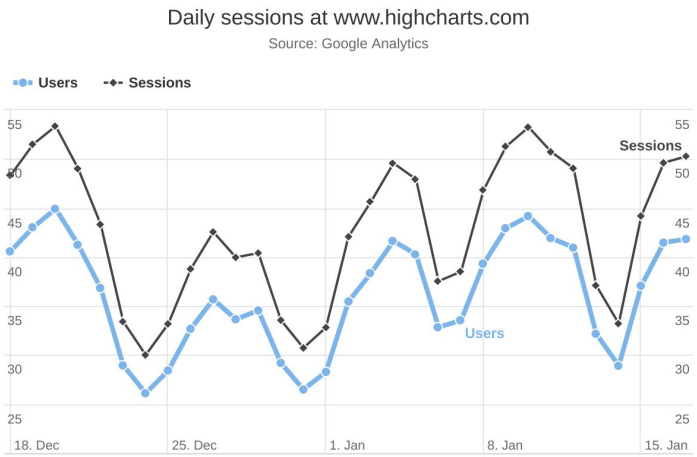
Gráficos interativos -> {googleVis}

`googleVis::renderGvis()` ⟶ `htmlOutput()`



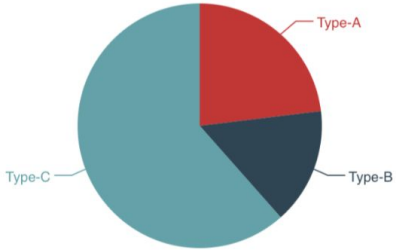
Gráficos interativos -> {highcharter}

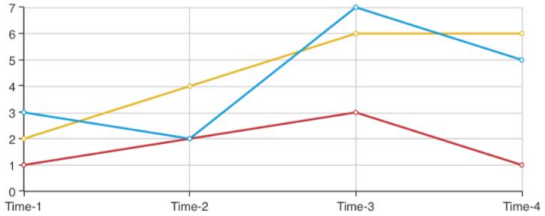
`highcharter::renderHighchart()` ⟶ `highcharter::highchartOutput()`

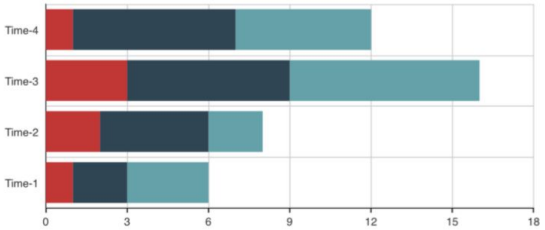


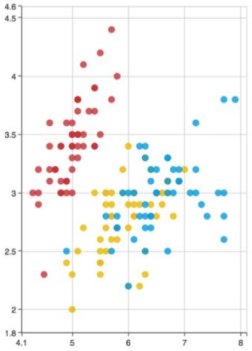
Gráficos interativos -> {echarts4r}

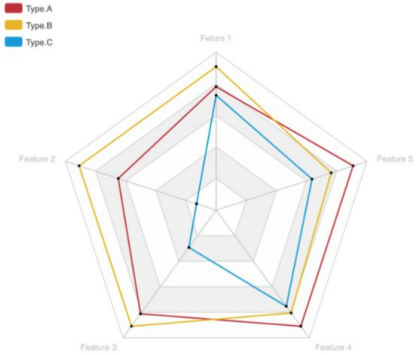
`echarts4r::renderEcharts4r()` ⟶ `echarts4r::echarts4rOutput()`





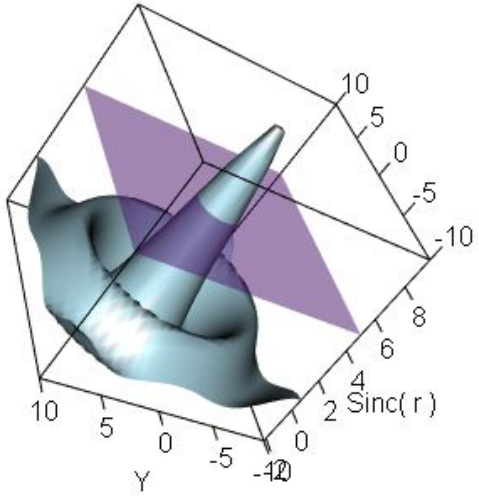




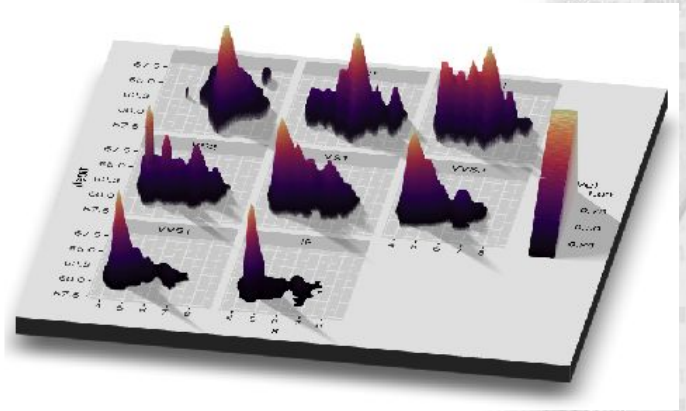


Gráficos interativos -> {rlg}

`rgl::renderRglwidget()` ⟶ `rgl::rglwidgetOutput()`



`{rgl}`



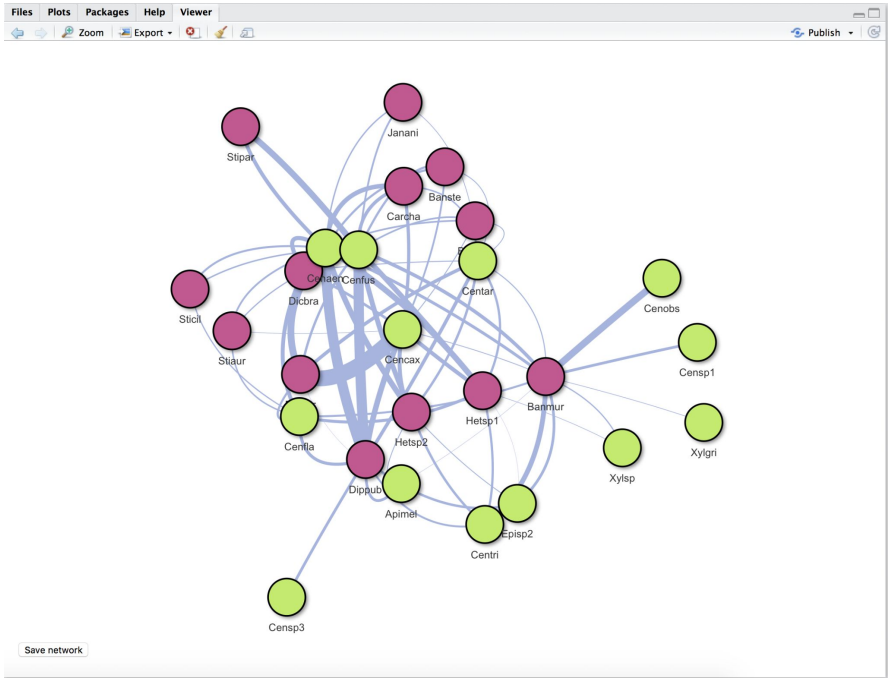
`{rayshader}`

Mapas interativos -> {leaflet}

`leaflet::renderLeaflet()` ⟶ `leaflet::leafletOutput()`

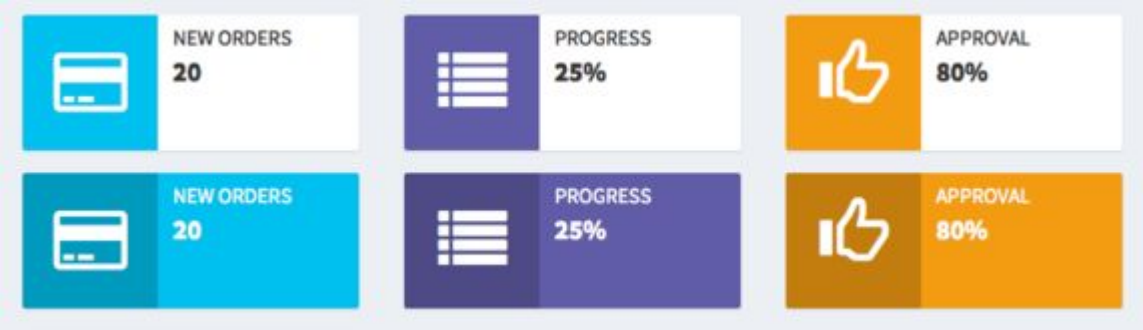
Redes de relacionamento -> {visNetwork}

`visNetwork::renderVisNetwork()` ⟶ `visNetwork::visNetworkOutput()`



Caixas de informação

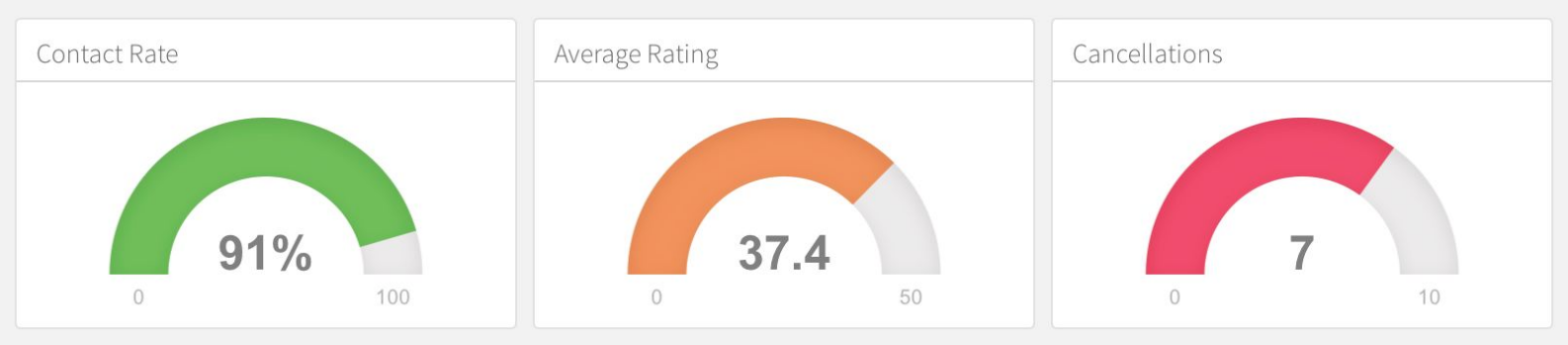
`shinydashboard::renderInfoBox()` → `shinydashboard::infoBoxOutput()`

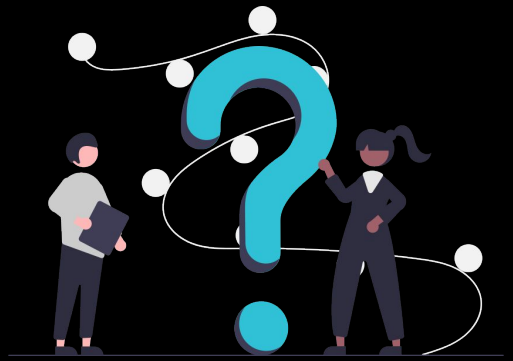


`shinydashboard::renderValueBox()` → `shinydashboard::valueBoxOutput()`

Indicadores ou medidores

`flexdashboard::renderGauge()` → `flexdashboard::gaugeOutput()`





Questions?

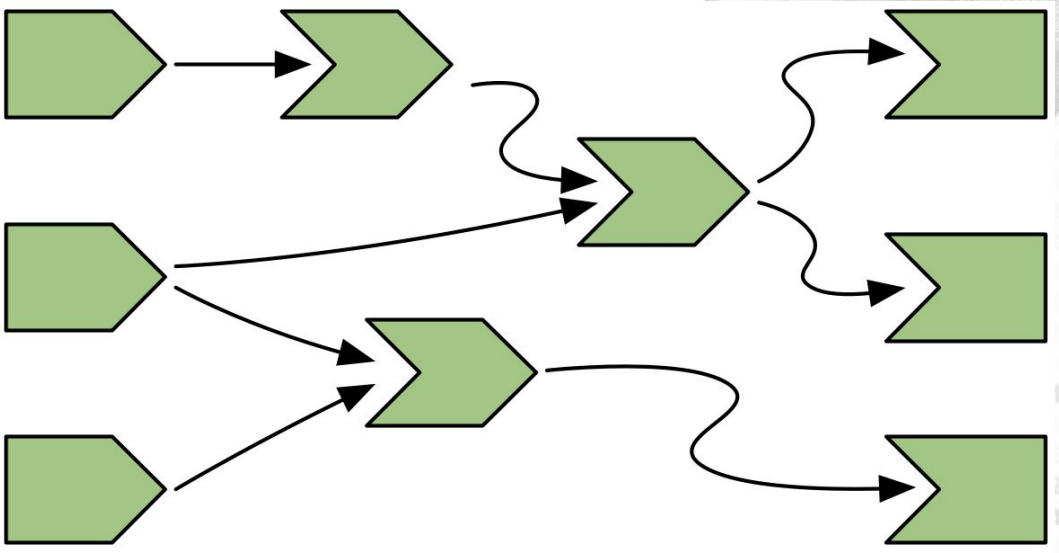
3.
Elementos para a reatividade

Como conectar "produtores" com "consumidores"

Reatividade

É o que conecta alterações em elementos de input com execução de código e outputs.

- Expressões reativas.
- Observadores.



https://mastering-shiny.org/reactive-graph.html

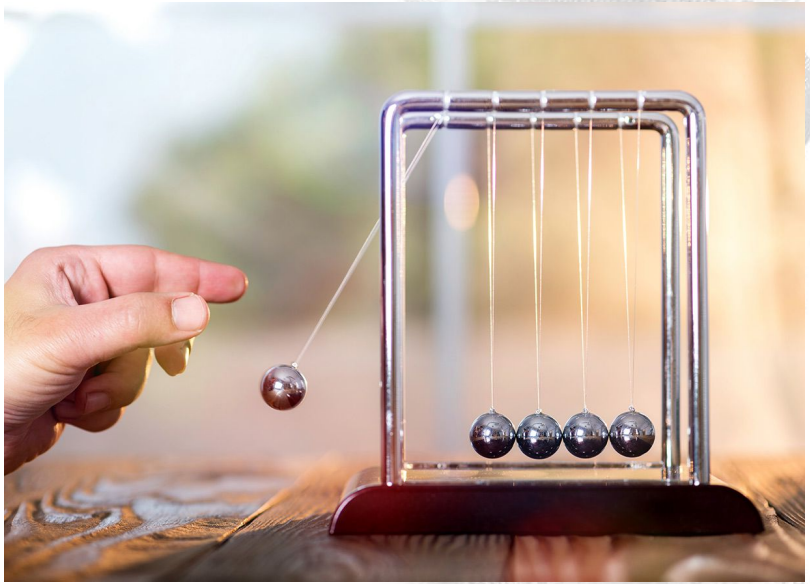
Expressões reativas

`reactive()`

Reage a todas inputs ou expressões reativas (IER) presentes em sua definição.

`eventReactive()`

Reage aos IER indicados. Ambos retornam conteúdo.



https://www.institutoclaro.org.br/educacao/para-ensinar/planos-de-aula/3a-lei-de-newton/

Expressões reativas

```
result <- reactive({
    input$X + input$Y
})

output$RESULT <- renderPrint({
    result()
})
```


```
result <- reactive({
    input$X + input$Y
})

output$RESULT <- renderPrint({
    result()
})
```

Outros elementos reativos

`reactiveVal()`

Um vetor com um valor reativo.

`reactiveValues()`

Um vetor com vários valores reativos.

Outros elementos reativos

`reactiveTime()`

Executa a cada intervalo de tempo definido.

`reactiveFileReader()`

Executa a cada intervalo de tempo e se o arquivo mudar, então é lido.

`reactivePool()`

Idem acima mas comunmente usado com bancos de dados.

Observadores

`observe()`

Reage a todas inputs ou expressões reativas (IER) presentes em sua definição mas sem retornar conteúdo.

`observeEvent()`

Reage aos IER indicados, sem retornar conteúdo também.

http://operacoesmilitaresguia.blogspot.com/2020/11/observacao-do-tiro-de-artilharia-por.html

Observadores

```
observe({
    updateTextInput(inputId = "TEXT",
                    value = input$X)
})
```

```
observeEvent(eventExpr = input$CLICK,
             handlerExpr = {
                 updateTextInput(inputId = "TEXT",
                                 value = input$X)
             })
```

Isolador

`isolate()`

Previne que determinado IER manifeste reatividade dentro de escopos sensíveis.

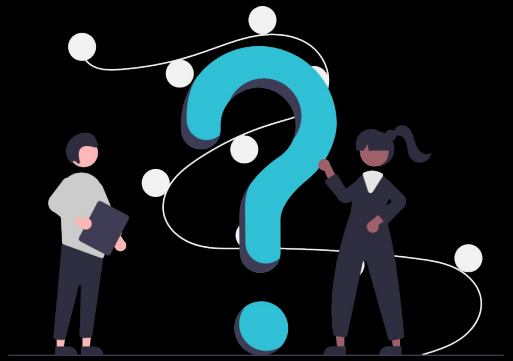
https://www.ndtv.com/offbeat/thief-falls-asleep-covered-in-doritos-wakes-up-in-handcuffs-1779059

Isolador

```
output$RESULT <- renderPrint({
    result()
})

result <- reactive({
    input$X + isolate(input$Y)
})
```

```
observe({
    input$saveButton
    data <- get(isolate(input$dataset))
    writeToDatabase(data)
})
```



Questions?

VAMOS PRATICAR