LAYOUTS, TEMAS E HTML

Prof. Walmes Zeviani

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

1.
Layouts

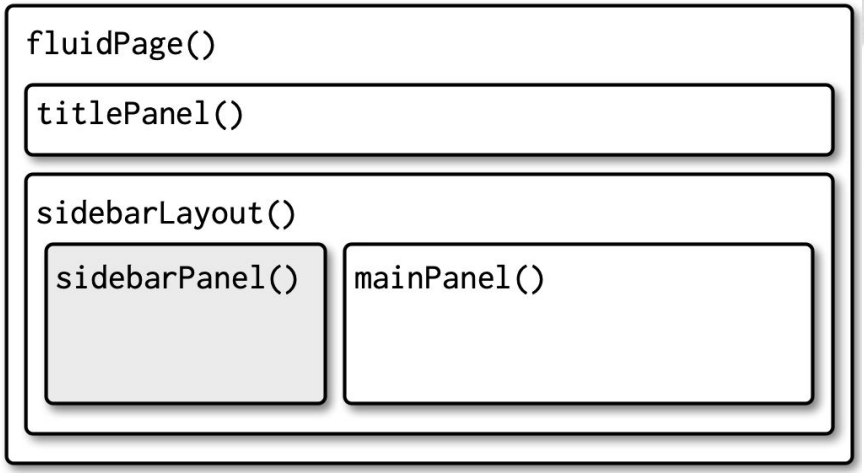
Diferentes formas de apresentar conteúdo na aplicação

Layouts

Layouts é sobre como apresentar o conteúdo na aplicação para o usuário.

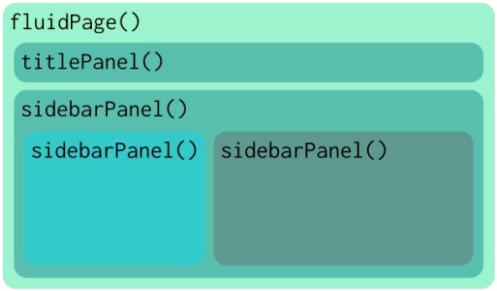
O {shiny} tem construtores que podem ser combinados para isso.

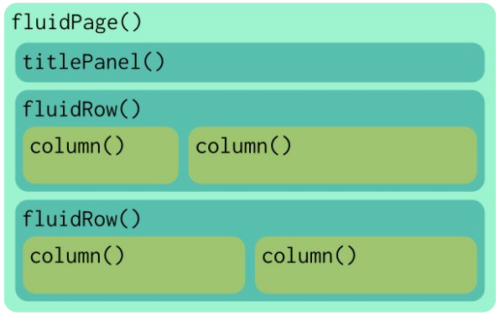
- ***Page()**: `fluidPage()`, `etc.`
- ***Layout()**: `sidebarLayout()`, `etc.`
- ***Row()**: `fluidRow()`, `etc.`
- ***Panel()**: `tabsetPanel()`, `etc.`

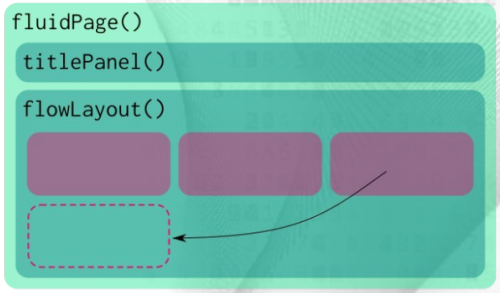


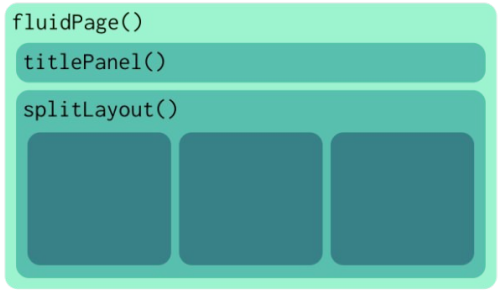
https://mastering-shiny.org/action-layout.html

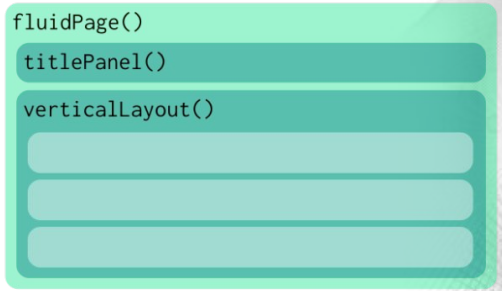
Alguns layouts





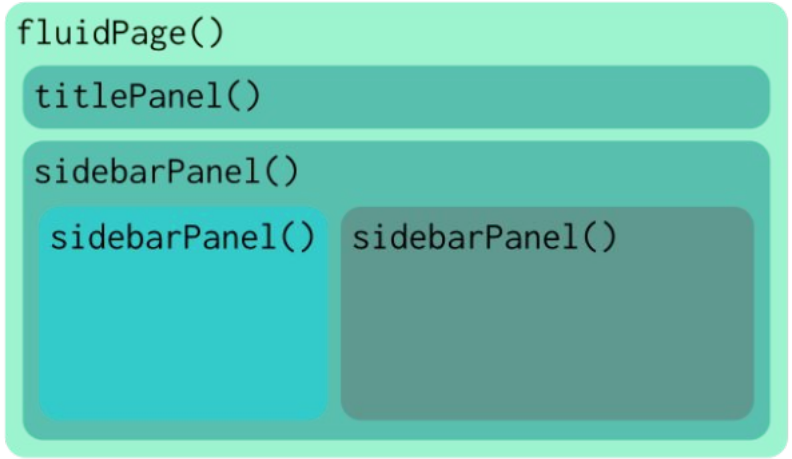






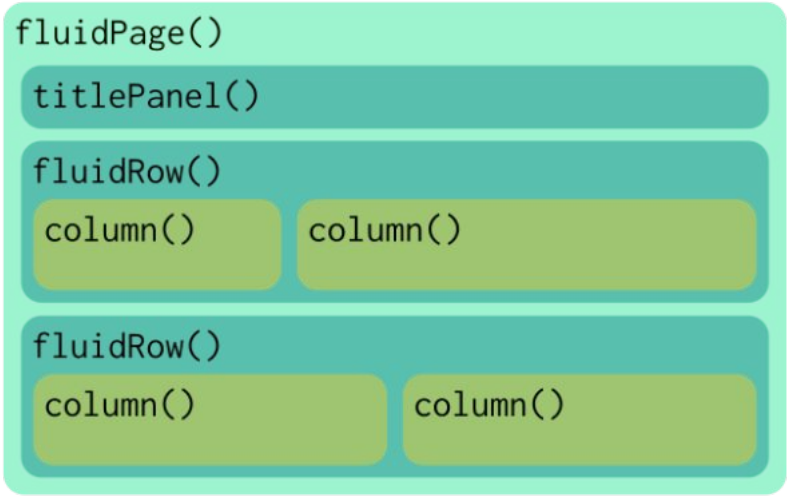
Menu lateral e área principal

- O que mais estamos acostumados.
- Na lateral os widgets (inputs).
- Na área principal, os gráficos, tabelas, etc.



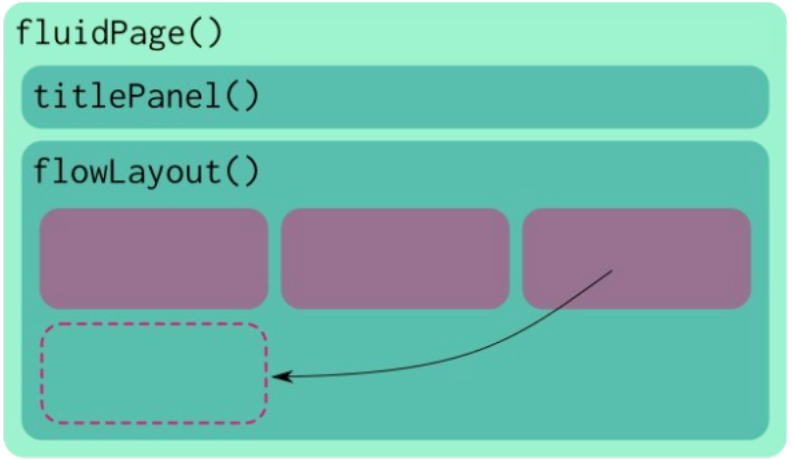
Linhas reativas

- Permite disposição na forma de grid com conteúdo responsivo.
- Colunas podem ter largura diferente.
- Adapta bem para mobile.



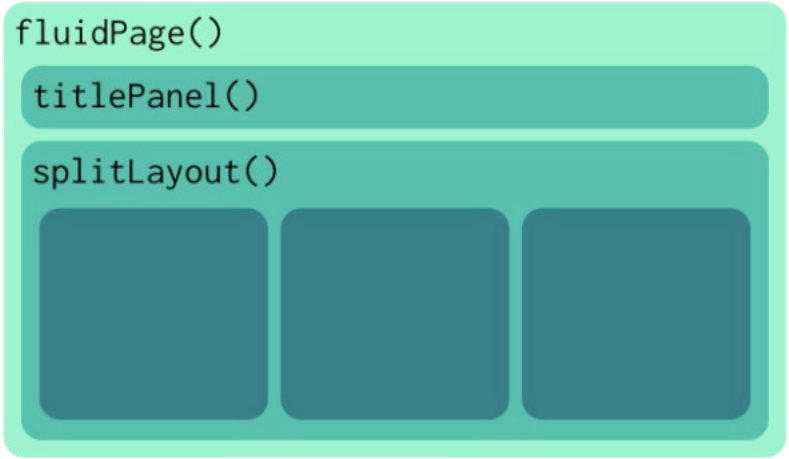
Linhas reativas

- Permite disposição ainda mais responsiva.
- Itens são empilhados conforme a largura diminui.
- Interessante para *infoboxes*.



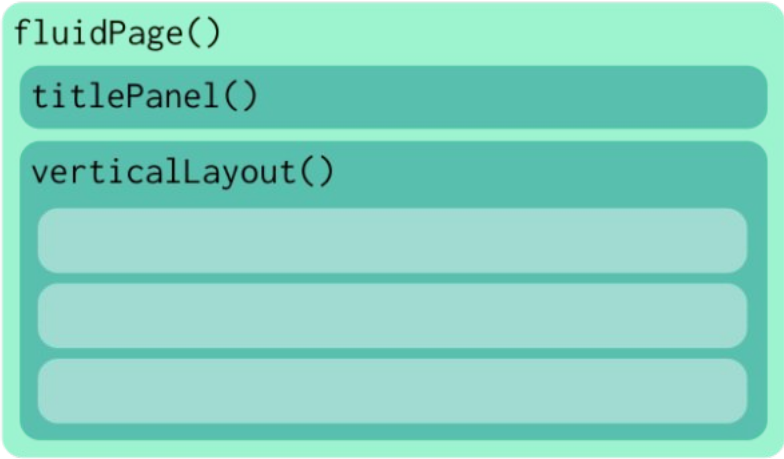
Layout fixo em colunas

- Permite disposição em colunas.
- Ele reduz a largura mas não empilha os elementos.
- Para mobile pode não ser interessante.



Layout fixo em linhas

- Permite disposição em linhas.
- É interessante para exibir conteúdo com muita largura, como tabelas.

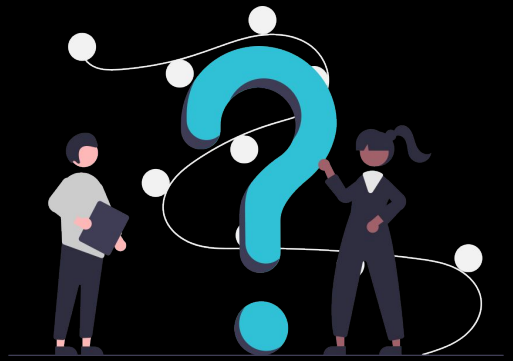


Importante sobre layouts

- É relativamente fácil mudar a disposição.
- No entanto, planeje a disposição **no começo** da construção do dashboard para evitar retrabalho.
- Você só não terá sucesso no desenvolvimento de aplicações shiny se não planejar o seu desenvolvimento.
  - Rascunhe e interface e construa com placeholders.
  - Desenvolva todo o backend: dados, funções, exibições de gráficos, tabelas, etc.
  - Faça a conexão dos elementos para gerar a reatividade.

Fluxo de desenvolvimento

1. ui.R
2. global.R
3. server.R



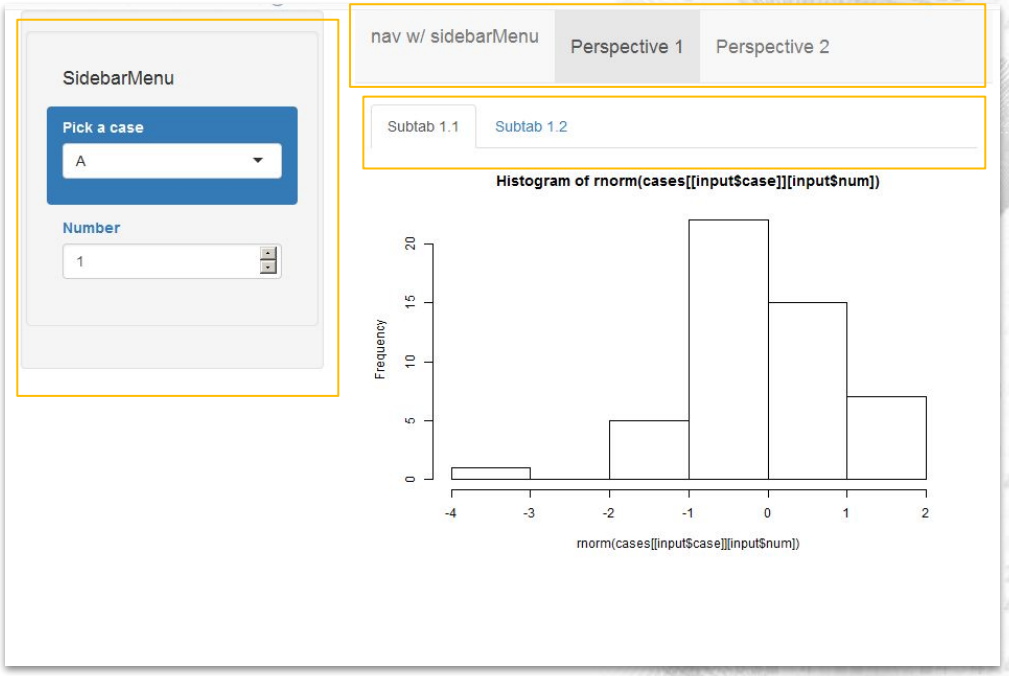
Questions?

2.
Menus de navegação

Adicionando mais navegabilidade

Menus de navegação

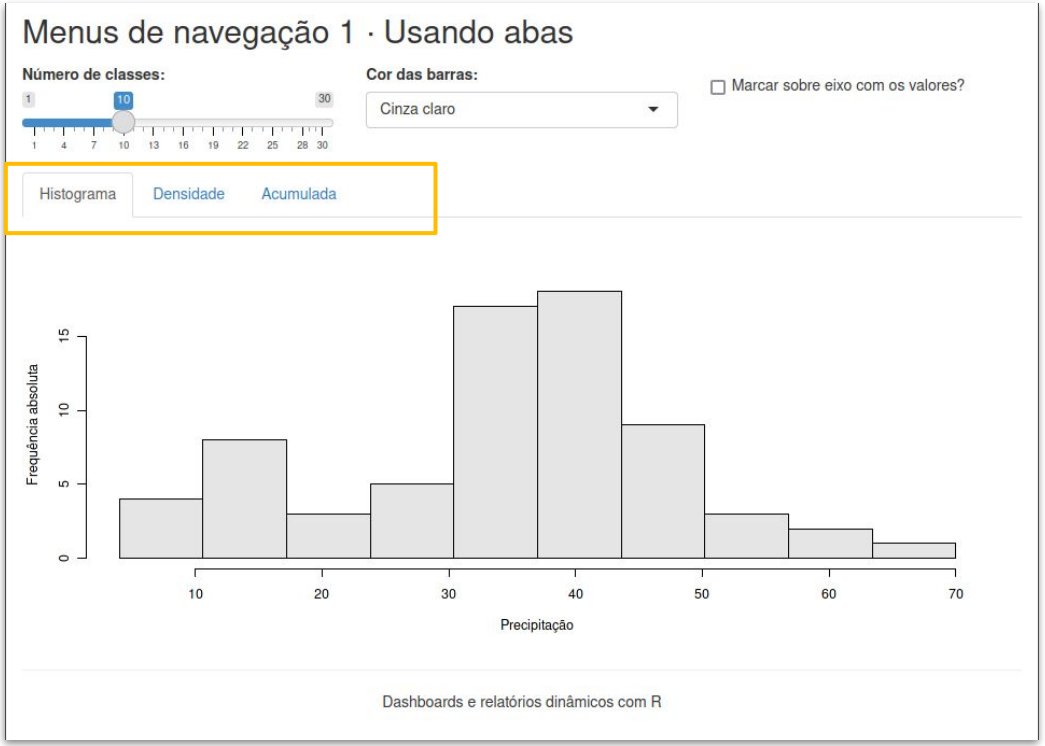
O shiny permite arranjos incríveis. Com menus você pode criar aplicações realmente complexas.



https://stackoverflow.com/questions/46145608/how-to-combine-top-navigation-navbarpage-and-a-sidebar-menu-sidebarmenu-in-s

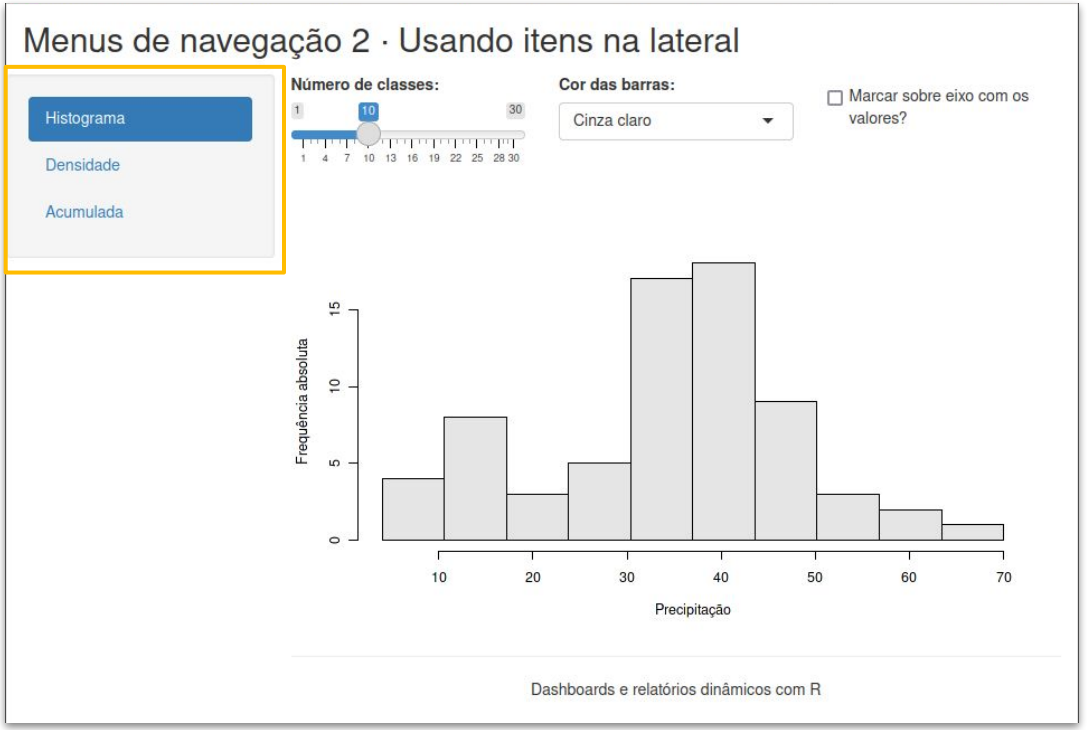
Disposição em abas

- Abas permitem acomodar muitos elementos.
- No entanto, apenas um é visível por vez.



Disposição em itens de menu

- Menus laterais também acomodam muitos elementos.
- No entanto, apenas um é visível por vez.

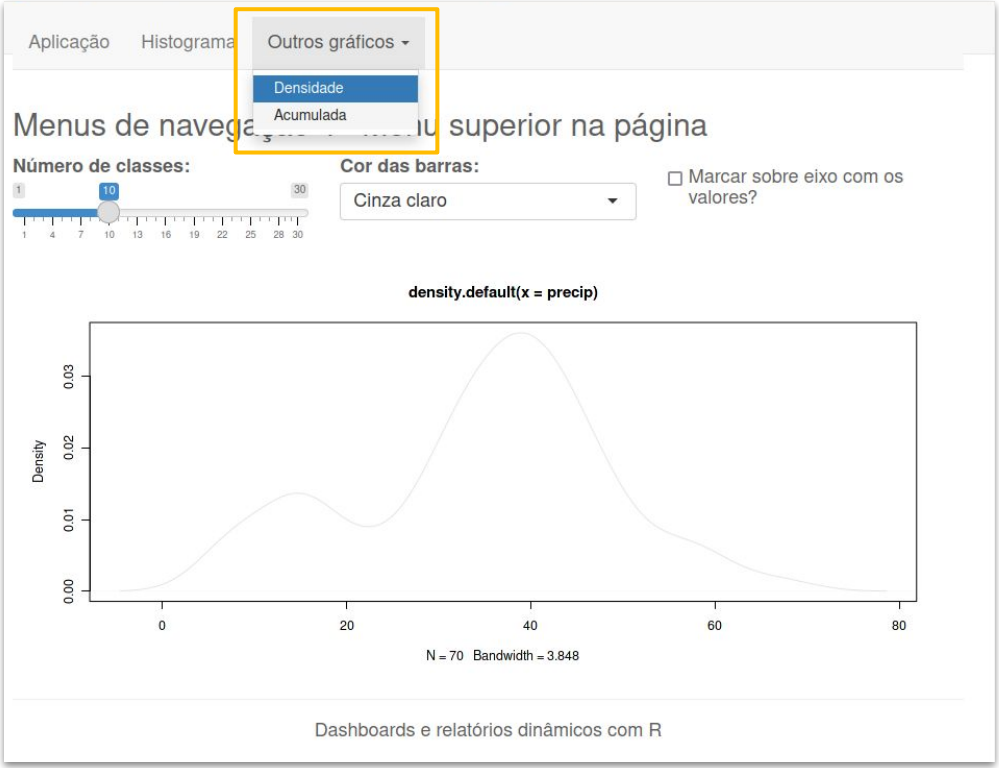


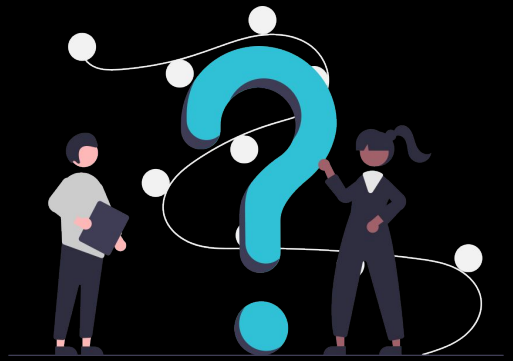
Menu de navegação superior

- Funciona como abas mas são usados no topo da página.

Subitens no menu

- O menu superior permite entrada de menu.
- Dá pra fazer uma hierarquia bem profunda.





Questions?

3.
Temas

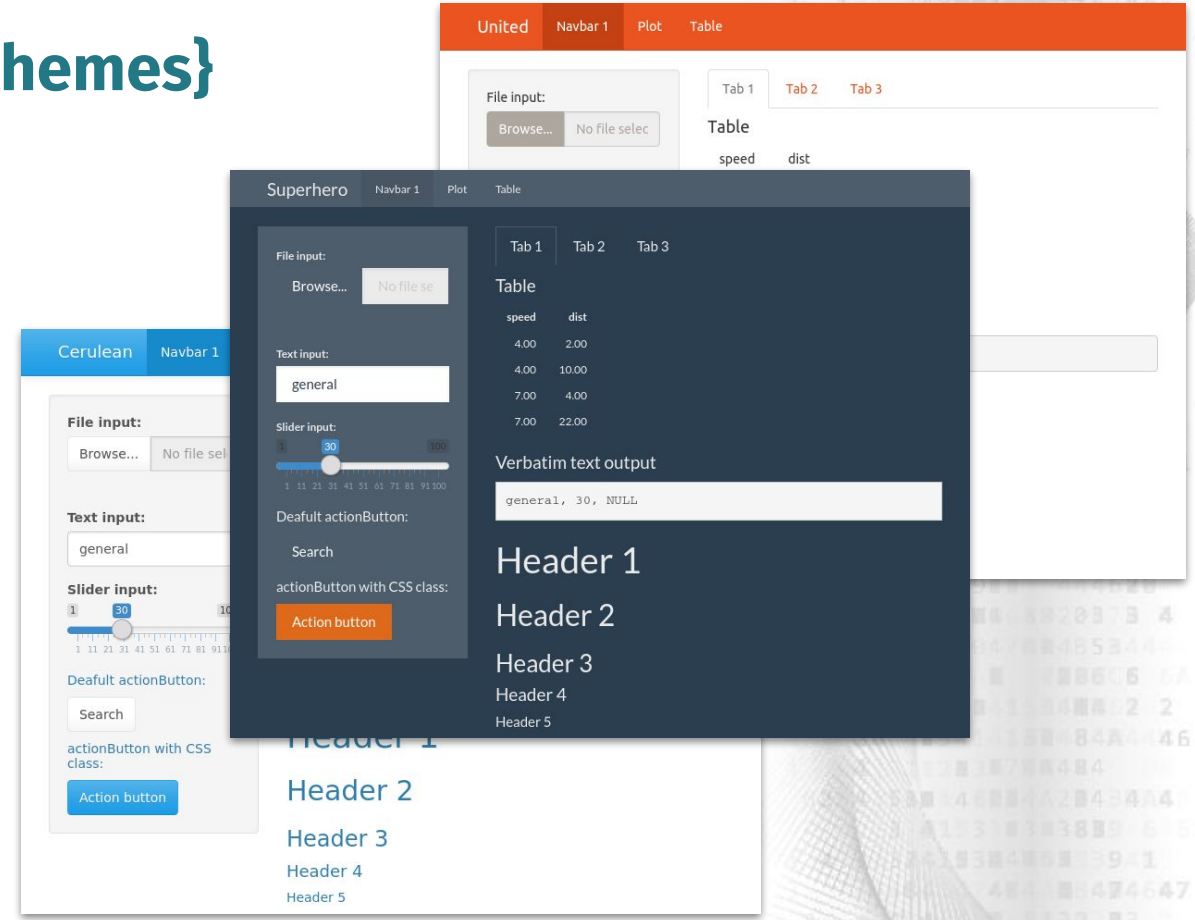
Como dar um colorido

Temas no {shinythemes}

O uso de temas é o jeito mais fácil de mudar a **identidade visual** da aplicação.

O {shinythemes} tráz 16 opções.

Ajustes finos podem ser feitos com CSS.

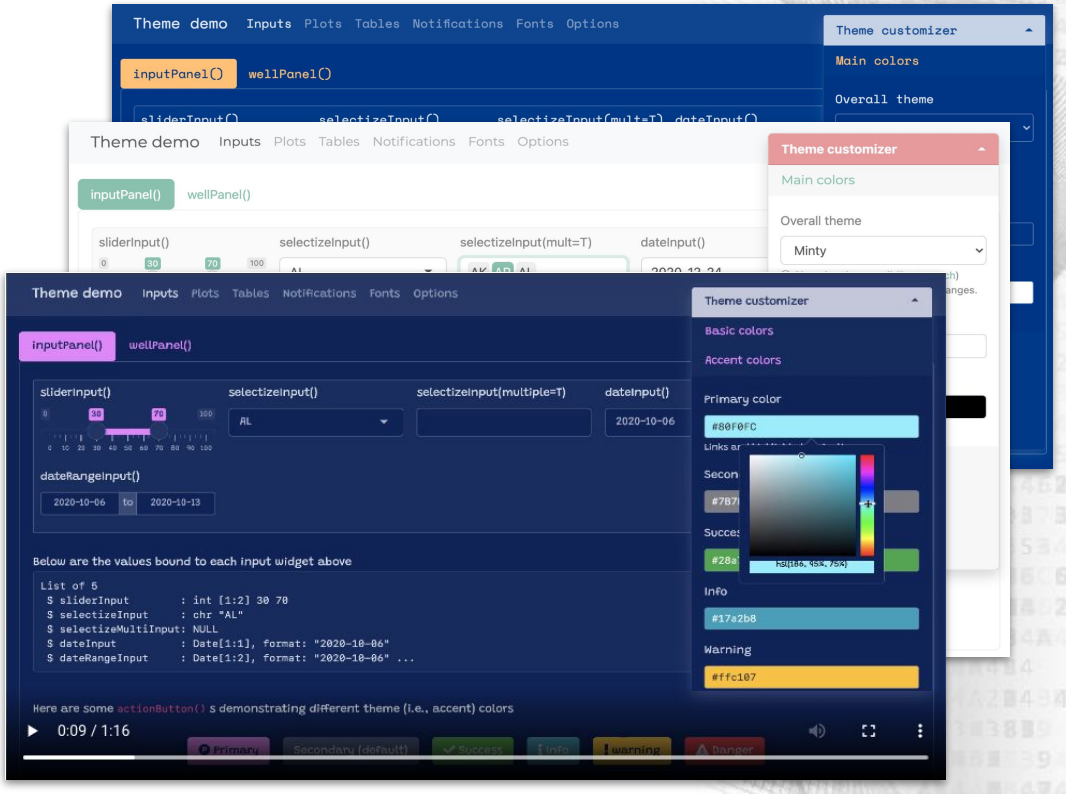


https://rstudio.github.io/shinythemes/

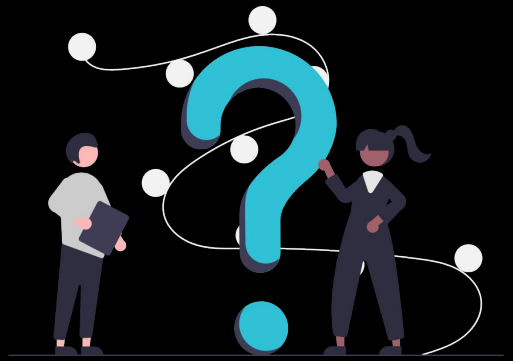
Temas no {bslib}

Mais opções de temas estão no {bslib}.

O {bslib} tem recursos para customizar de forma bem ágil.



https://shiny.rstudio.com/articles/themes.html



Questions?

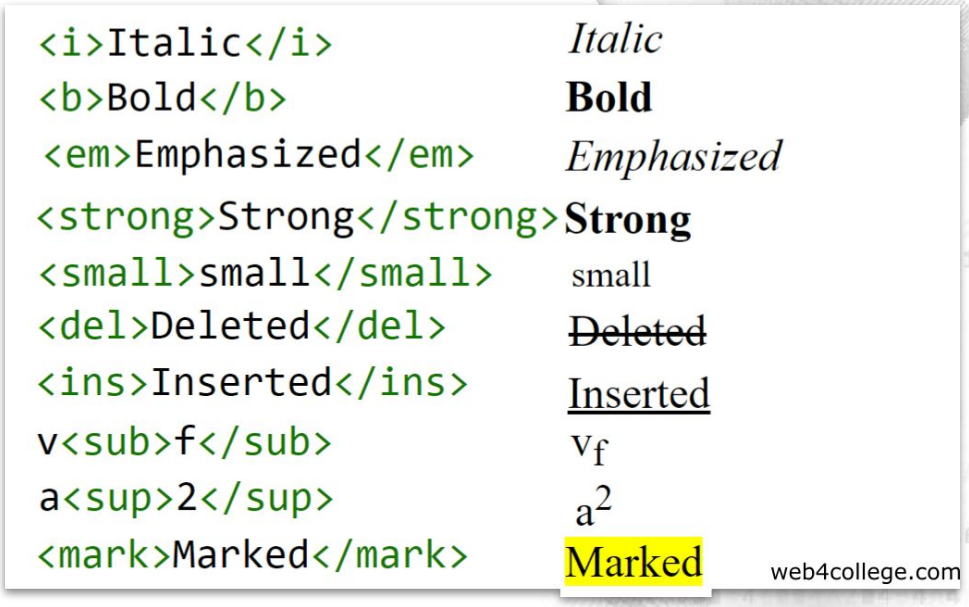
3.
HTML e CSS

Fazendo mais ajustes

HTML

Com HTML você pode:

- Criar elementos do zero.
- Decorar a exibição do texto.

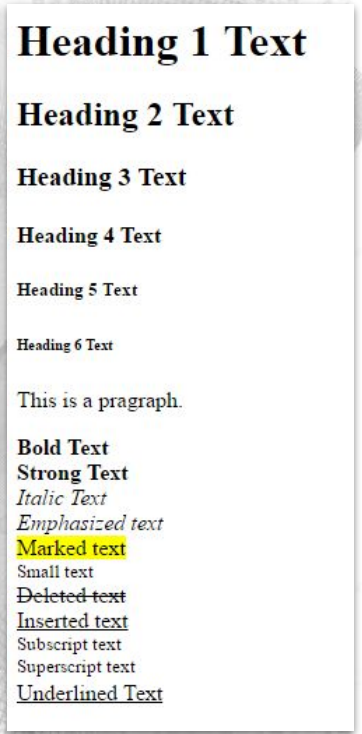


http://webdesigningbydkc.blogspot.com/2015/07/html-text-formatting.html

Criação de elementos HTML

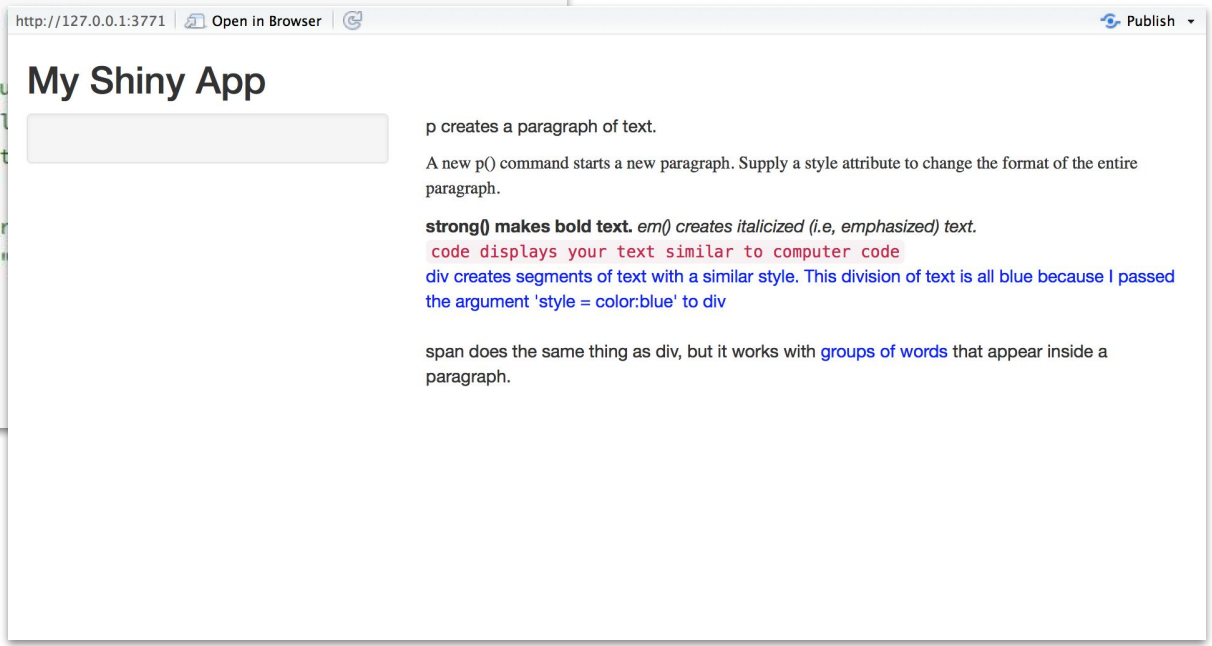
Existem várias funções para construção dos elementos em HTML.

p	<p>	A paragraph of text
h1	<h1>	A first level header
h2	<h2>	A second level header
h3	<h3>	A third level header
h4	<h4>	A fourth level header
h5	<h5>	A fifth level header
h6	<h6>	A sixth level header
a	<a>	A hyper link
br	 	A line break (e.g. a blank line)
div		A division of text with a uniform style
span		An in-line division of text with a uniform style
pre	<pre>	Text 'as is' in a fixed width font
code	<code>	A formatted block of code
img	<img>	An image
strong		Bold text
em		Italicized text
HTML		Directly passes a character string as HTML code



https://shiny.rstudio.com/tutorial/written-tutorial/lesson2/

```
ui <- fluidPage(
  titlePanel("My Shiny App"),
  sidebarLayout(
    sidebarPanel(),
    mainPanel(
      p("p creates a paragraph of text."),
      p("A new p() command starts a new paragraph. Supply a style attribute to change the format
of the entire paragraph.", style = "font-family: 'times'; font-si16pt"),
      strong("strong() makes bold text."),
      em("em() creates italicized (i.e, emphasized)
      br(),
      code("code displays your text similar to compu
      div("div creates segments of text with a simil
because I passed the argument 'style = color:blue' t
      br(),
      p("span does the same thing as div, but it wor
        span("groups of words", style = "color:blue"
        "that appear inside a paragraph.")
    )
  )
)
```

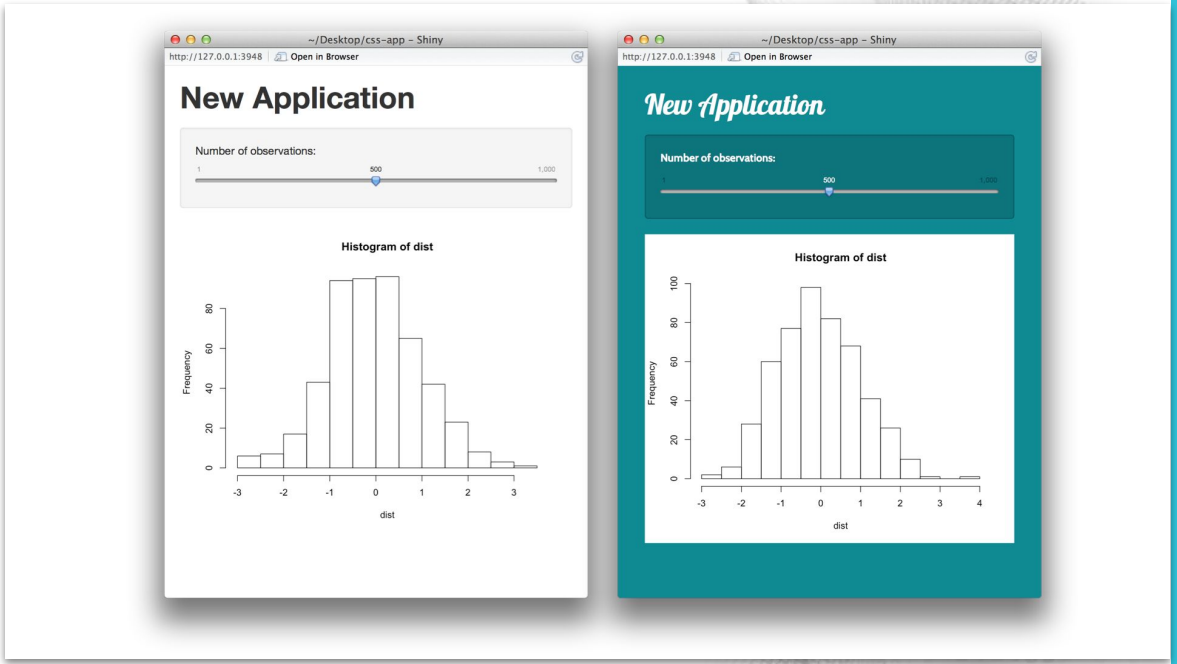


https://shiny.rstudio.com/tutorial/written-tutorial/lesson2/

CSS

Com CSS você muda inúmeros detalhes da aparência:

- Fontes.
- Cores.
- Linhas e contornos.
- Margens.
- Alinhamento.
- Sombras.



http://shiny.rstudio-staging.com/articles/css.html

CSS em linha

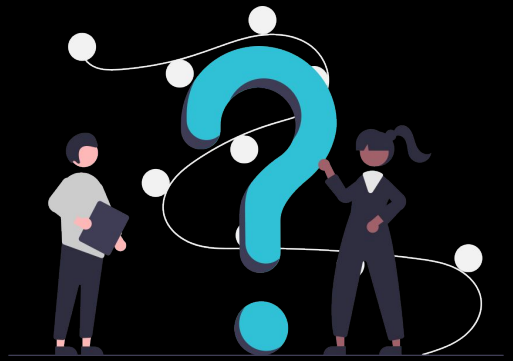
```
shinyUI(
    fluidPage(
        tags$head(
            tags$style(HTML("DECLARAÇÕES CSS AQUI"))
        ),
        ...
    )
)
```

CSS em arquivo

```
shinyUI(
    fluidPage(
        includeCSS("styles.css"),
        ...
    )
)
```

Importante sobre temas e CSS

- Se você já **possui** identidade visual, aplique logo **no começo**.
- Se você ainda **não possui** uma identidade visual, deixe para aplicar/testar **no final** quando sua aplicação tiver conteúdo.



Questions?

VAMOS PRATICAR