Custom themes in Shiny & R Markdown with {bslib} & {thematic}

Carson Sievert
Software Engineer, RStudio

Slides bit.ly/rstudio-global-theming

@cpsievert

RStudio®

New R 📦s for custom themes

- `{bslib}`: customize Bootstrap CSS from R
- `{thematic}`: translate CSS to static R plots
- Learn more:
  - rstudio.github.io/bslib
  - rstudio.github.io/thematic

Start using {bslib} with {shiny}

```r
library(shiny)

ui <- fluidPage(
 theme = bslib::bs_theme(),
 ...
)
```

Start using {bslib} with {shiny}

```r
library(shiny)

ui <- fluidPage(
 theme = bslib::bs_theme(),
 ...
)
```

Use with any `fluidPage()`, `navbarPage()`, or `bootstrapPage()`!

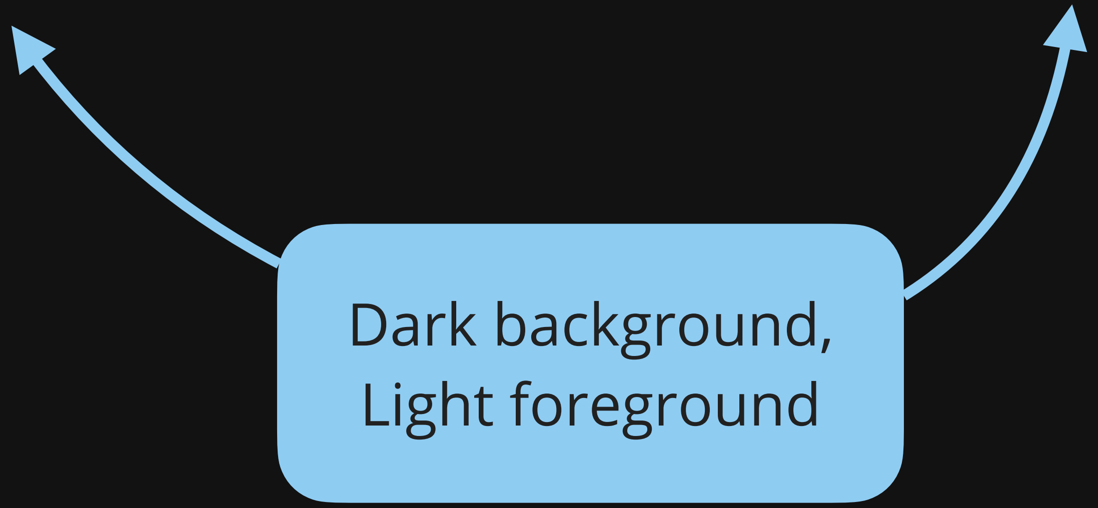
Hello Bootstrap 4

```r
library(shiny)

ui <- fluidPage(
  theme = bslib::bs_theme(version = 4),
  ...
)
```

Also works with 3
(and eventually 5)!

Custom base colors

```r
library(shiny)

ui <- fluidPage(
  theme = bslib::bs_theme(
    bg = "#121212", fg = "#E4E4E4"
  ),
  ...
)
```



Dark background,
Light foreground

Custom accent colors

```r
library(shiny)

ui <- fluidPage(
  theme = bslib::bs_theme(
    bg = "#121212", fg = "#E4E4E4",
    primary = "#BB86FC",
    secondary = "#48DAC6"
  ),
  ...
)
```

The most important accent colors

Custom fonts

```
library(shiny)

ui ← fluidPage(
 theme = bslib::bs_theme(
  bg = "#121212", fg = "#E4E4E4",
  primary = "#BB86FC",
  secondary = "#48DAC6",
  base_font = font_google("Open Sans")
 ),
 ...
)
```

Intelligently download, cache, and import font files

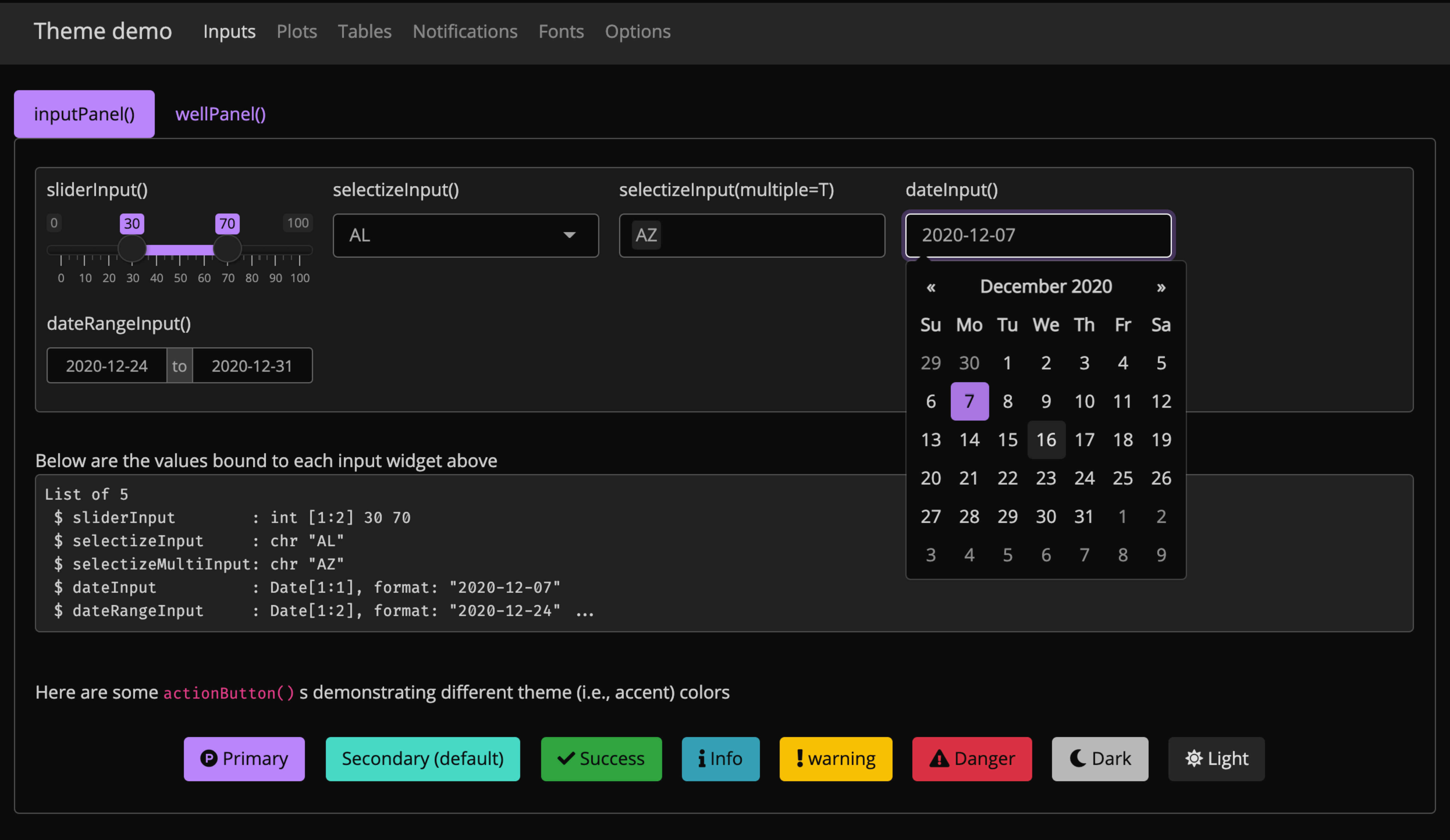
Detailed customization

```
library(shiny)

ui ← fluidPage(
 theme = bslib::bs_theme(
  bg = "#121212", fg = "#E4E4E4",
  primary = "#BB86FC",
  secondary = "#48DAC6",
  base_font = font_google("Open Sans”"),
  "progress-bar-bg" = "orange"
 ),
  ...
)
```

100s of more specific styles available

Preview a theme

```
bslib::bs_theme_preview(theme)
```



sliderInput(): 30 – 70

selectizeInput(): AL

selectizeInput(multiple=T): AZ

dateInput(): 2020-12-07

dateRangeInput(): 2020-12-24 to 2020-12-31

« December 2020 »

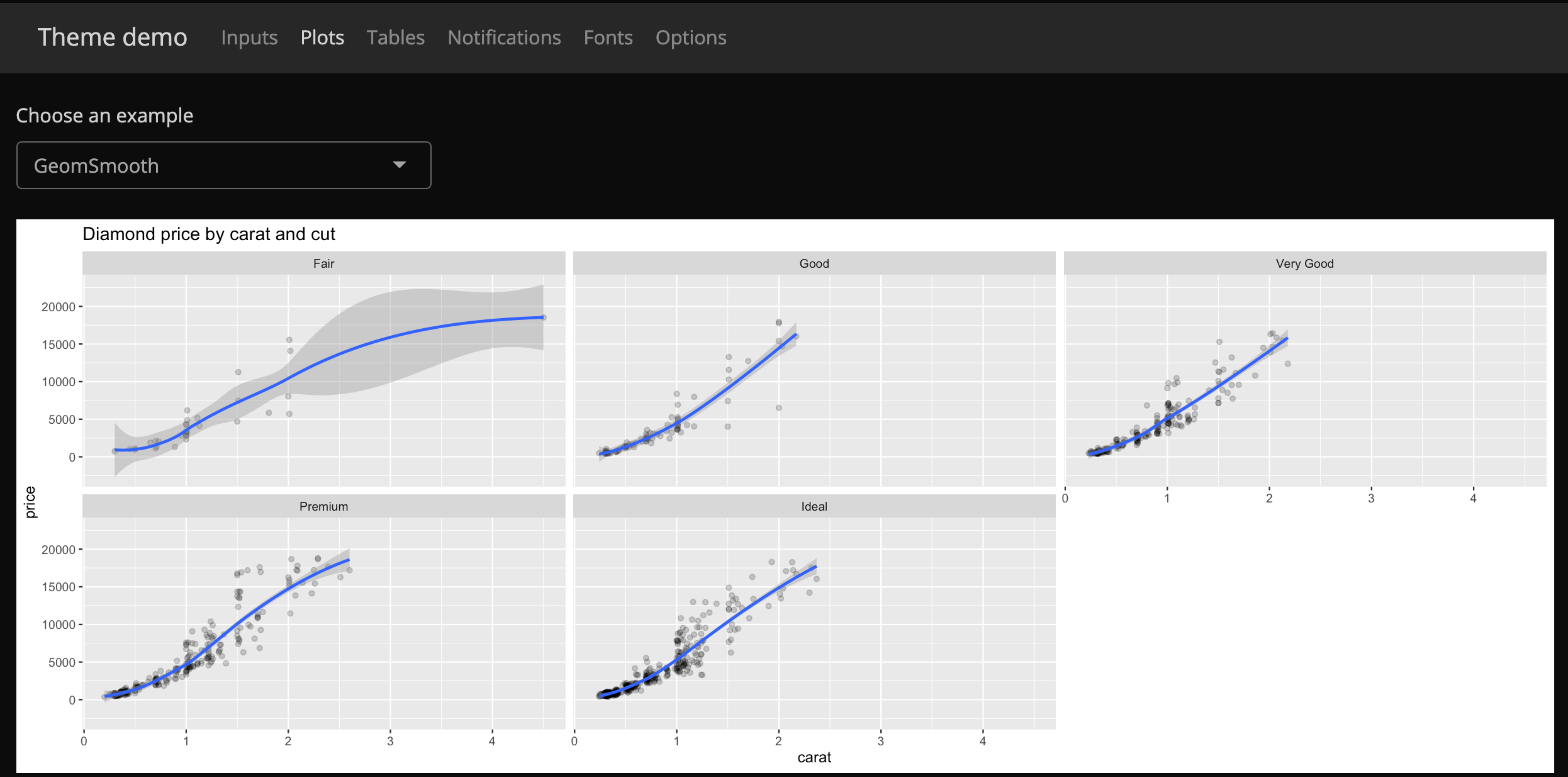
Su	Mo	Tu	We	Th	Fr	Sa
29	30	1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30	31	1	2
3	4	5	6	7	8	9

Below are the values bound to each input widget above

```
List of 5
 $ sliderInput        : int [1:2] 30 70
 $ selectizeInput     : chr "AL"
 $ selectizeMultiInput: chr "AZ"
 $ dateInput          : Date[1:1], format: "2020-12-07"
 $ dateRangeInput     : Date[1:2], format: "2020-12-24" ...
```

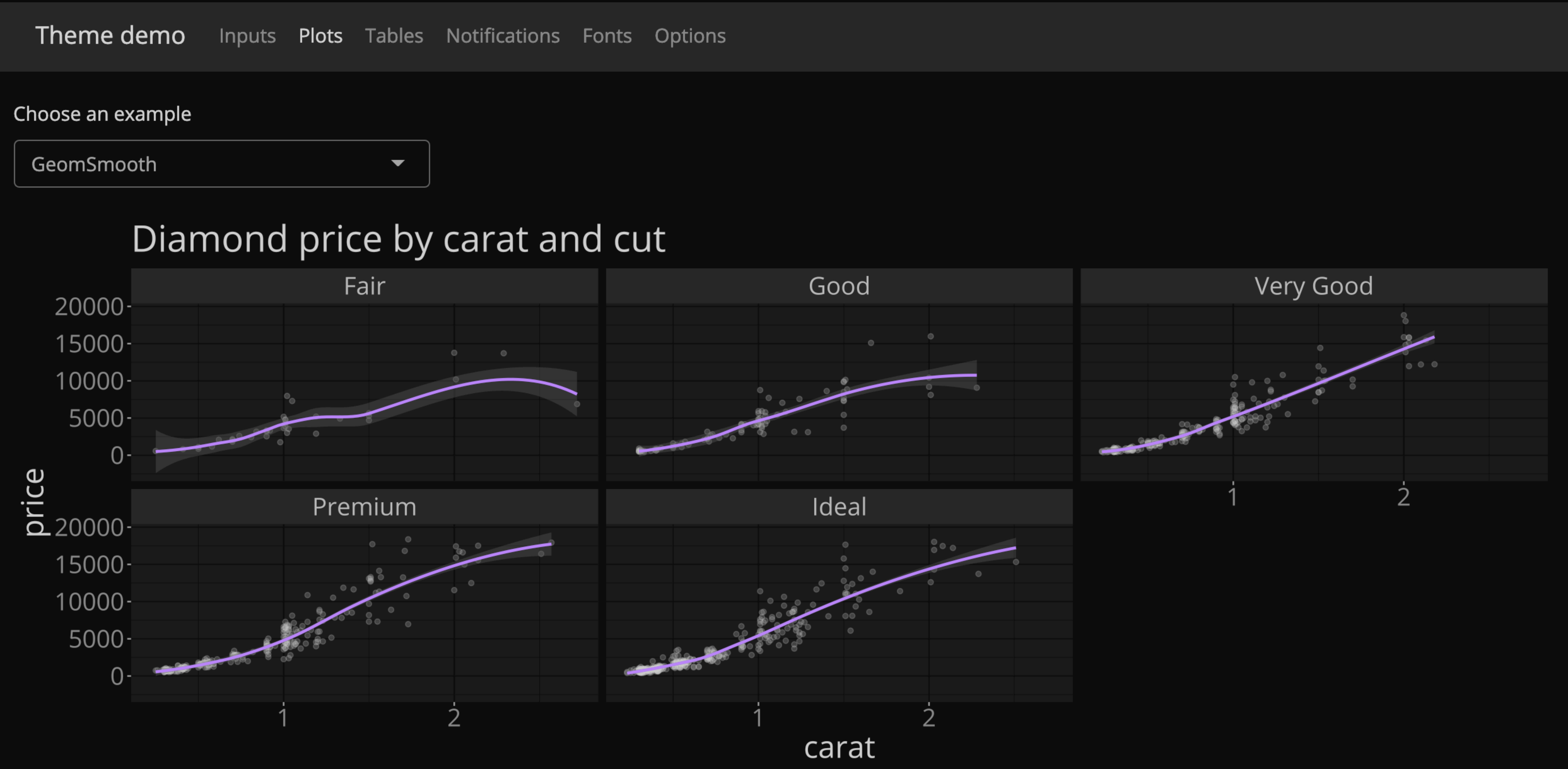
Here are some `actionButton()` s demonstrating different theme (i.e., accent) colors

Plots don’t reflect theme 😞



Now they do! 🎉

```
thematic::thematic_shiny()
```

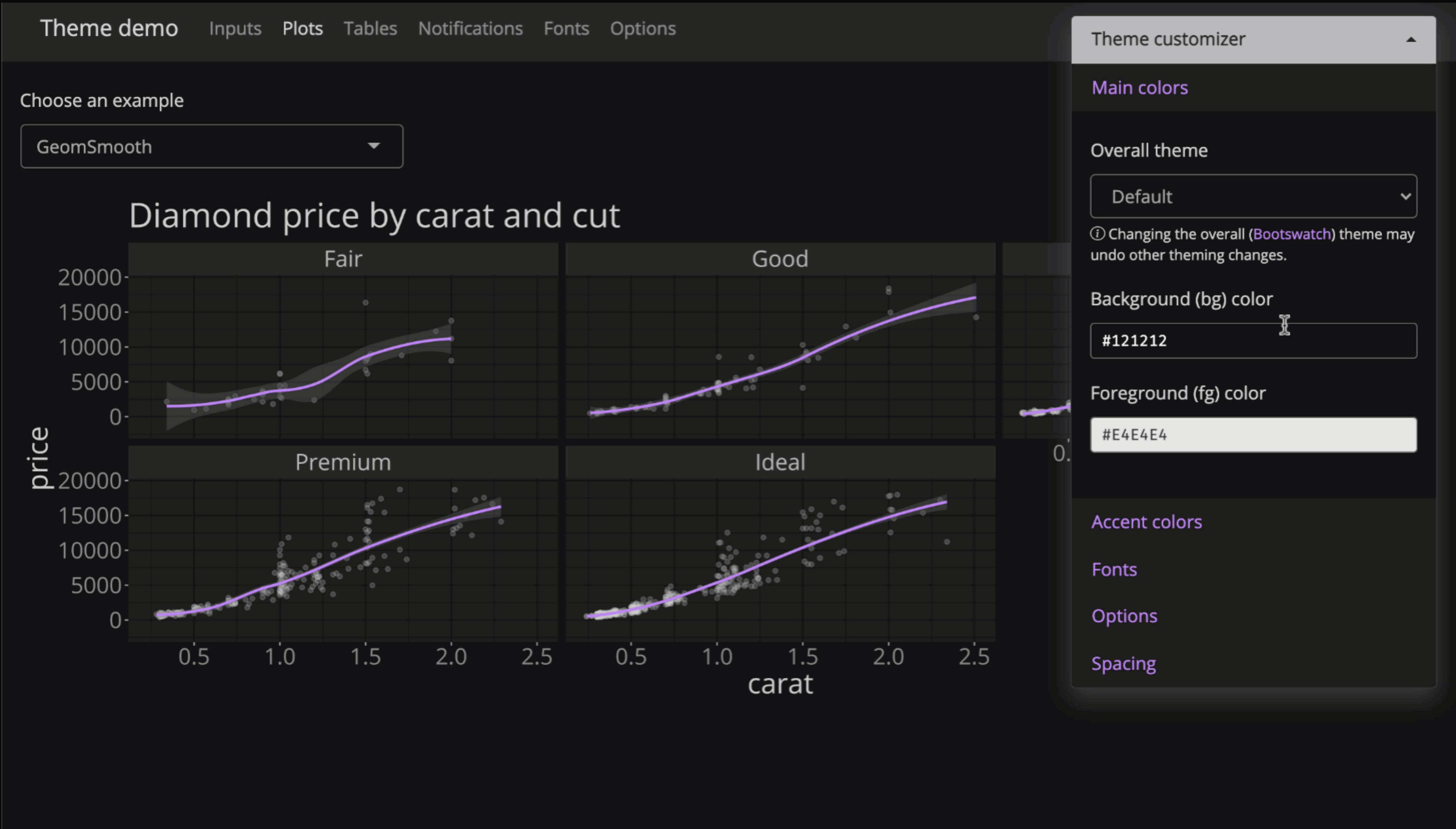


Choose an example

GeomSmooth

Interactive theming

```
bslib::bs_themer()
```



Choose an example

GeomSmooth

Theme customizer

Overall theme

Default

ⓘ Changing the overall (Bootswatch) theme may undo other theming changes.

Background (bg) color

#121212

Foreground (fg) color

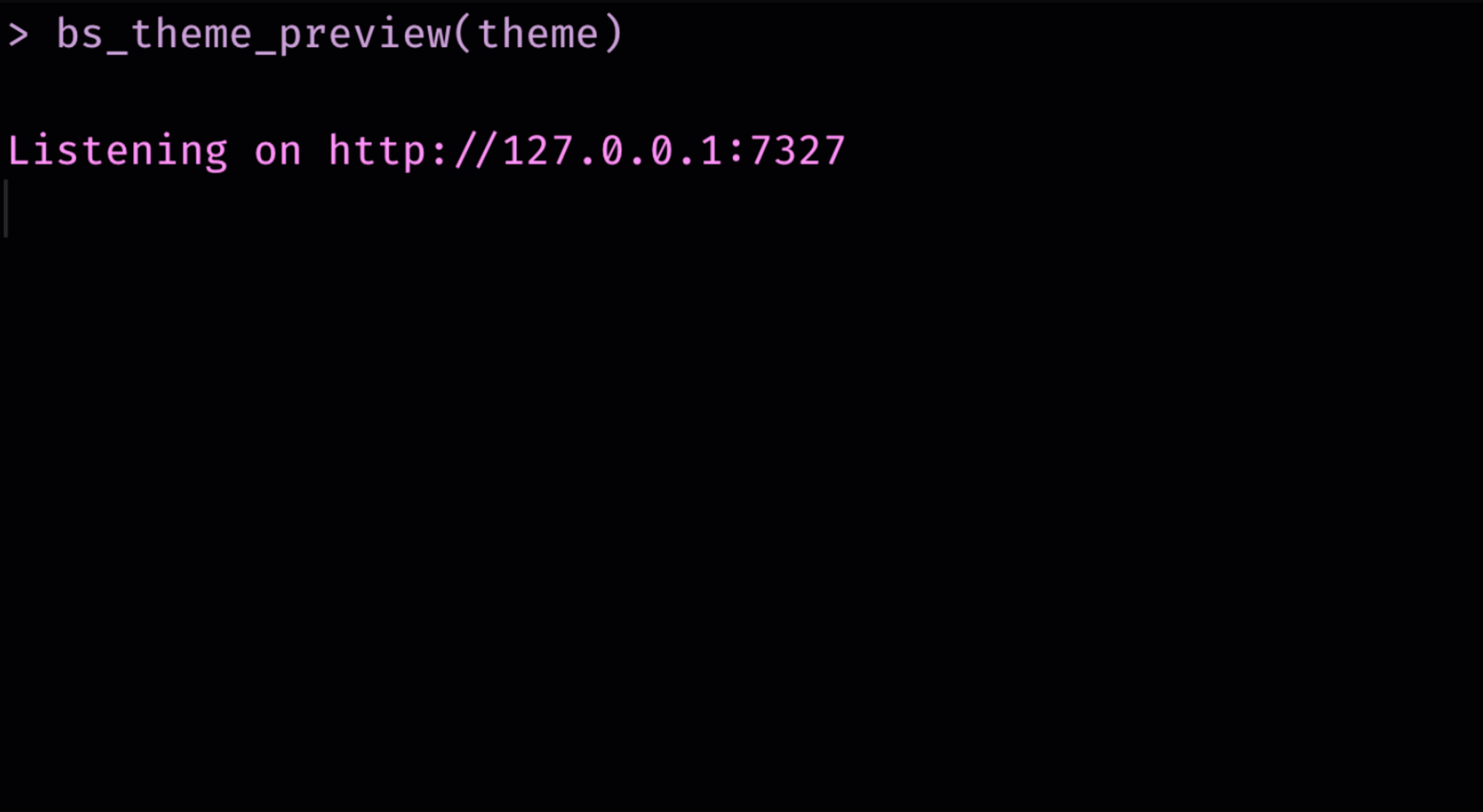
#E4E4E4

Interactive theming in your app

```r
library(shiny)

ui <- fluidPage(
  theme = bslib::bs_theme(),
  ...
)

server <- function(input, output) {
  bslib::bs_themer()
  ...
}
```

Requires Bootstrap 4

Theming updates as code



```
> bs_theme_preview(theme)

Listening on http://127.0.0.1:7327
```

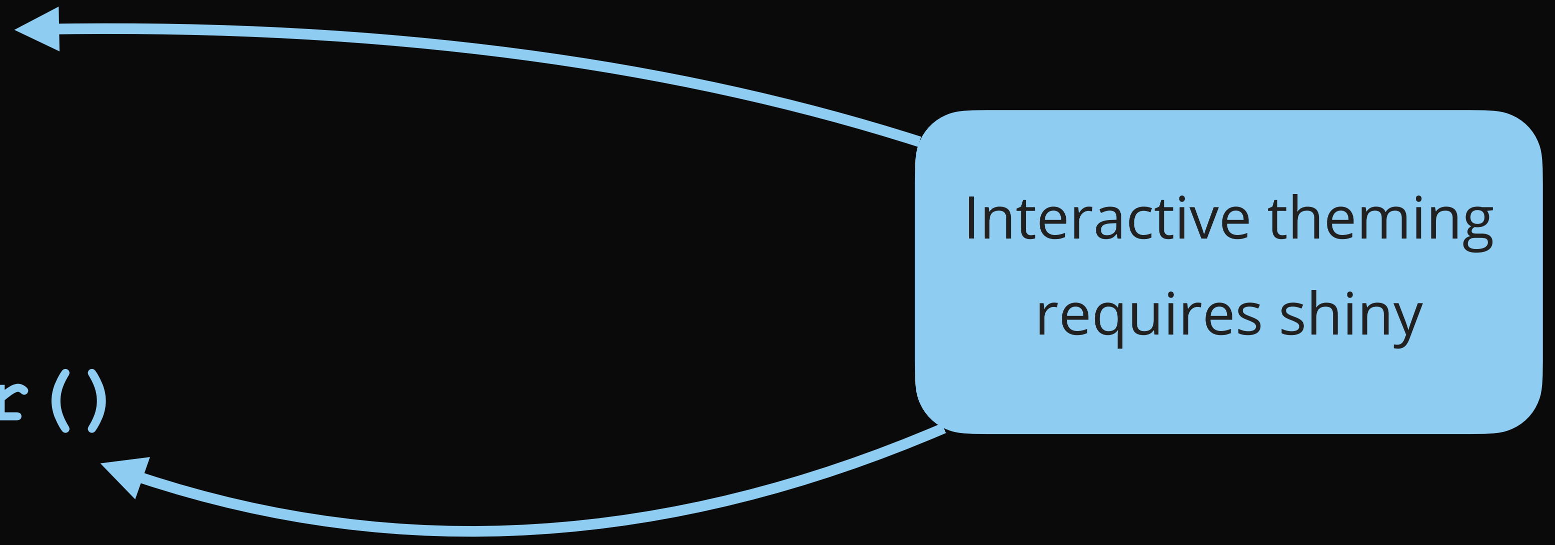
Custom R Markdown themes

```
---
output:
  html_document:
    theme:
      bg: "#121212"
      fg: "#E4E4E4"
      primary: "#BB86FC"
---
```

Arguments to `bslib::bs_theme()`

Interactive Rmd theming

~~~
---
output:
  html_document:
    theme:
      bg: "#121212"
      fg: "#E4E4E4"
      primary: “#BB86FC"
runtime: shiny
---

```{r}
bslib::bs_themer()
```
~~~



Interactive theming requires shiny

Auto theme static R plots

~~~
---
output:
  html_document:
    theme:
      bg: "#121212"
      fg: "#E4E4E4"
      primary: "#BB86FC"
runtime: shiny
---

```{r}
bslib::bs_themer()
thematic::thematic_shiny()
~~~

```{r}
renderPlot(...)
```
```

Translate CSS to R plots
```

# Use thematic anywhere

```
---
output:
  tufte::tufte_handout
---
```

Not an HTML format!

~~~
```{r}
thematic::thematic_rmd(
  bg = "#121212",
  fg = “#E4E4E4",
  accent = "#BB86FC",
  font = "Open Sans"
)
```
~~~

Similar theming options to `bs_theme()`

Thank you!

Learn more:

rstudio.github.io/bslib
rstudio.github.io/thematic

Slides:

bit.ly/rstudio-global-theming

Contact:

🐦 @cpsievert

✉ carson@rstudio.com

🖥 cpsievert.me