

Tips for making faster and better looking Shiny apps

Carson Sievert
Software Engineer, RStudio
@cpsievert

Explore your weather

Return to previous city

Search for a city

Ann Arbor, MI

Try a random city



Stations contributing data *Click on station to go to its dataset.*

Data sourced from [NOAA Climate Normals](#) generated by taking average temperatures from weather stations over the years 1982-2010. For cities with multiple weather stations the average across all reporting stations is used.

Fetching data from NOAA
Downloading data from all found stations

Live app

Avoid redundant work with `bindCache()`

`bindCache()` adds persistent caching to a reactive expression

```
weatherData ← reactive({  
  fetchData(input$city)  
})
```

```
weatherData ← reactive({  
  fetchData(input$city)  
}) %>%  
bindCache(input$city)
```

```
weatherData ← reactive({  
  fetchData(input$city)  
}) %>%  
bindCache(input$city)  
  
output$plot ← renderPlot({  
  plot(weatherData())  
})
```

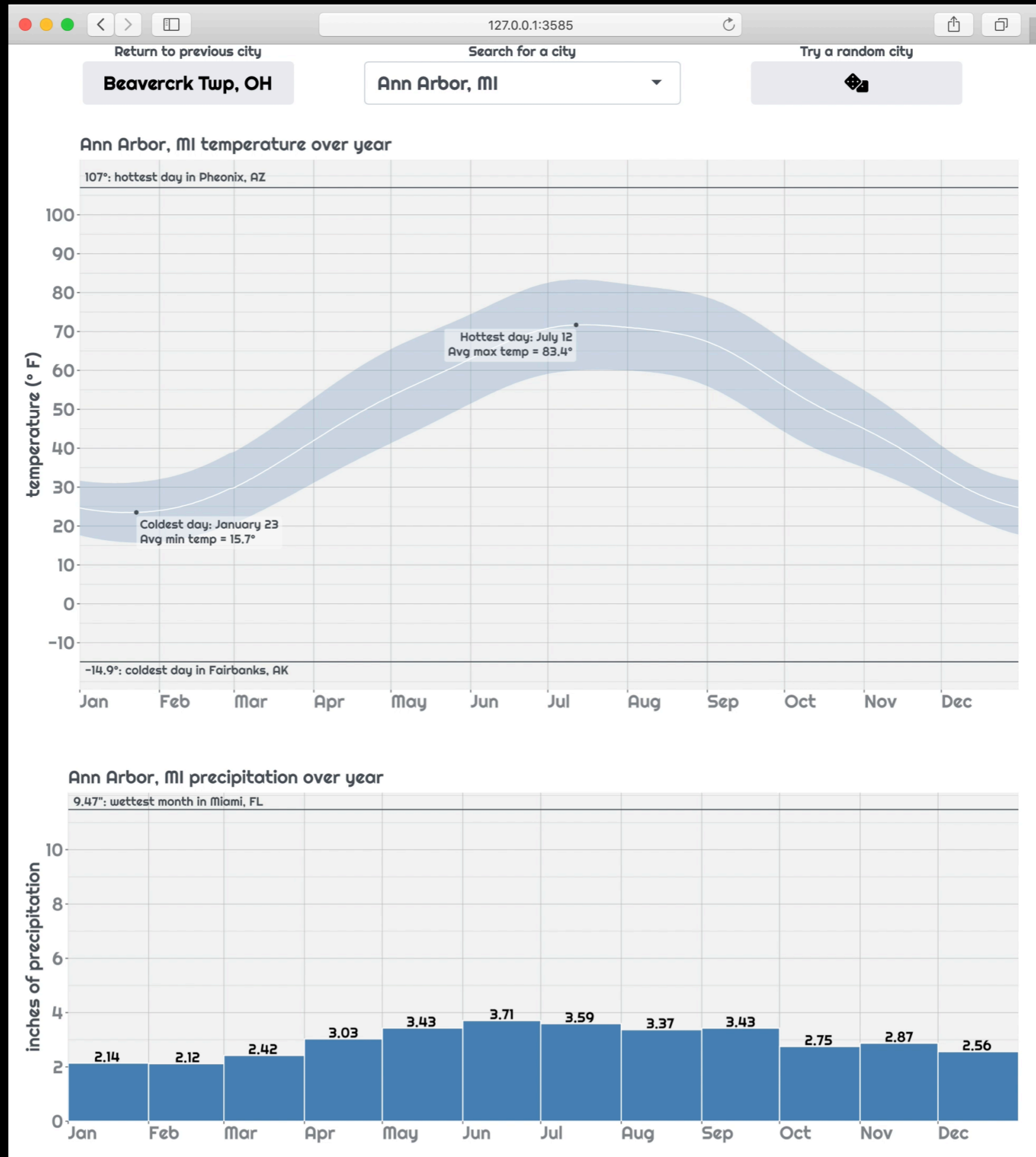
```
weatherData ← reactive({  
  fetchData(input$city)  
}) %>%
```

```
bindCache(input$city)
```

```
output$plot ← renderPlot({  
  plot(weatherData())  
}) %>%
```

```
bindCache(weatherData())
```

Previous cities are instantaneous!



```
reactive( ... ) %>%  
  bindCache(input$city)
```

```
renderPlot( ... ) %>%  
  bindCache(input$city)
```

```
renderText( ... ) %>%  
  bindCache(input$city)
```

```
plotly::renderPlotly( ... ) %>%  
  bindCache(input$city)
```

Share work with a disk-based cache

By default, cache is limited to 200 MB **in memory**.

Use a **disk-based** cache to persist across multiple users, R processes, and even system reboots.

For usage, see [bindCache\(\)'s reference](#)

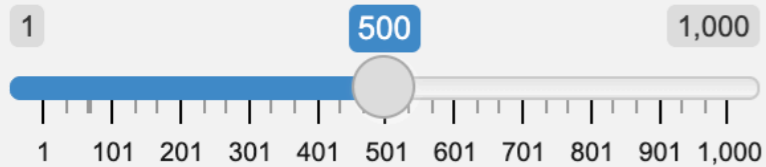
Shiny UI is powered by [Bootstrap](#)

Tabsets

Distribution type:

- Normal
- Uniform
- Log-normal
- Exponential

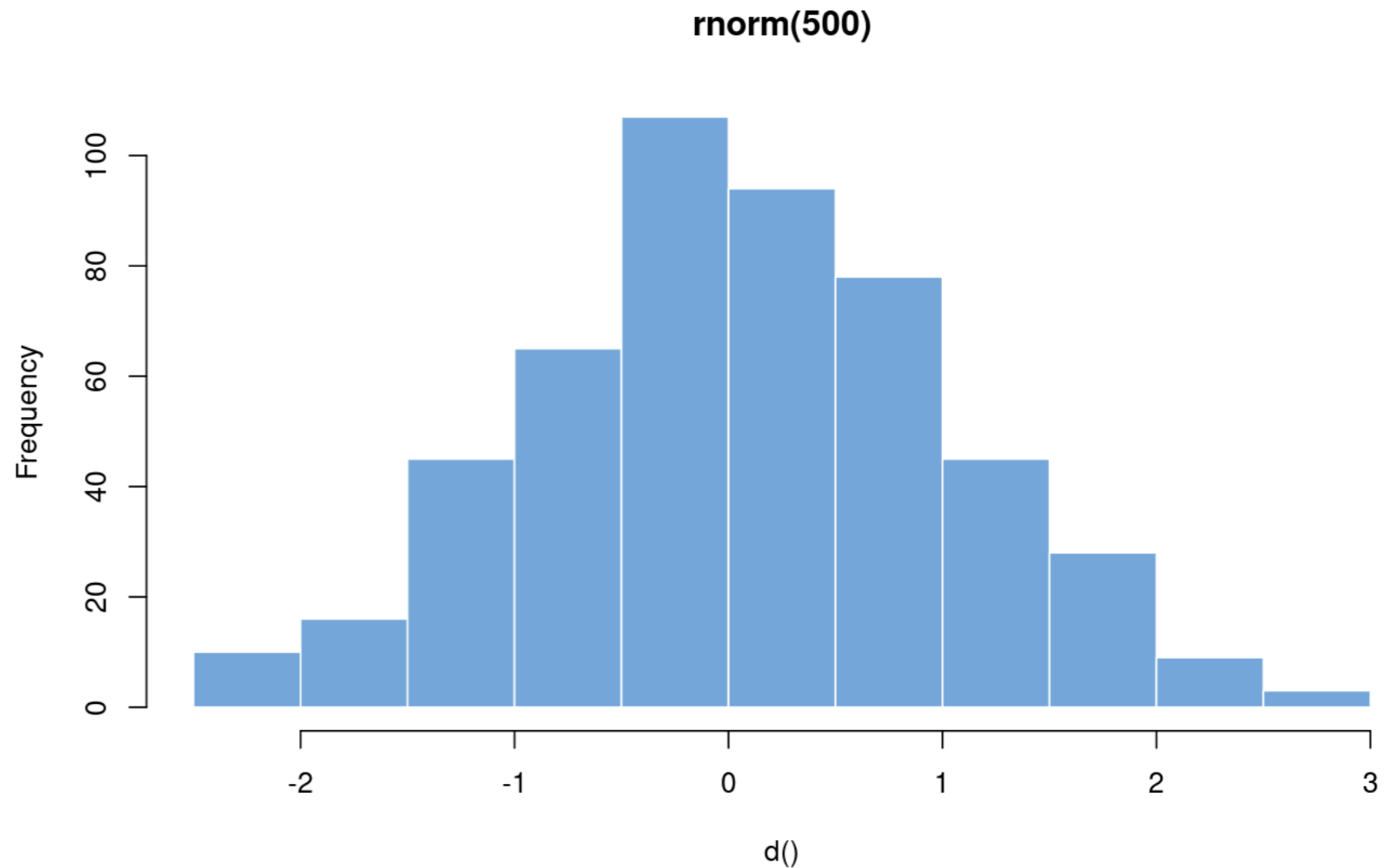
Number of observations:



Plot

Summary

Table



Shiny UI is powered by [Bootstrap](#)

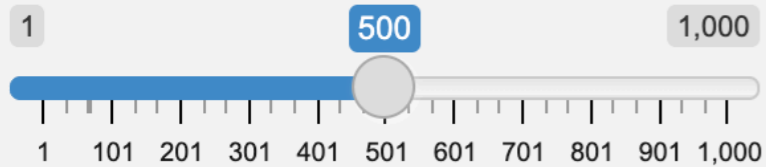
Bootstrap component

Tabsets

Distribution type:

- Normal
- Uniform
- Log-normal
- Exponential

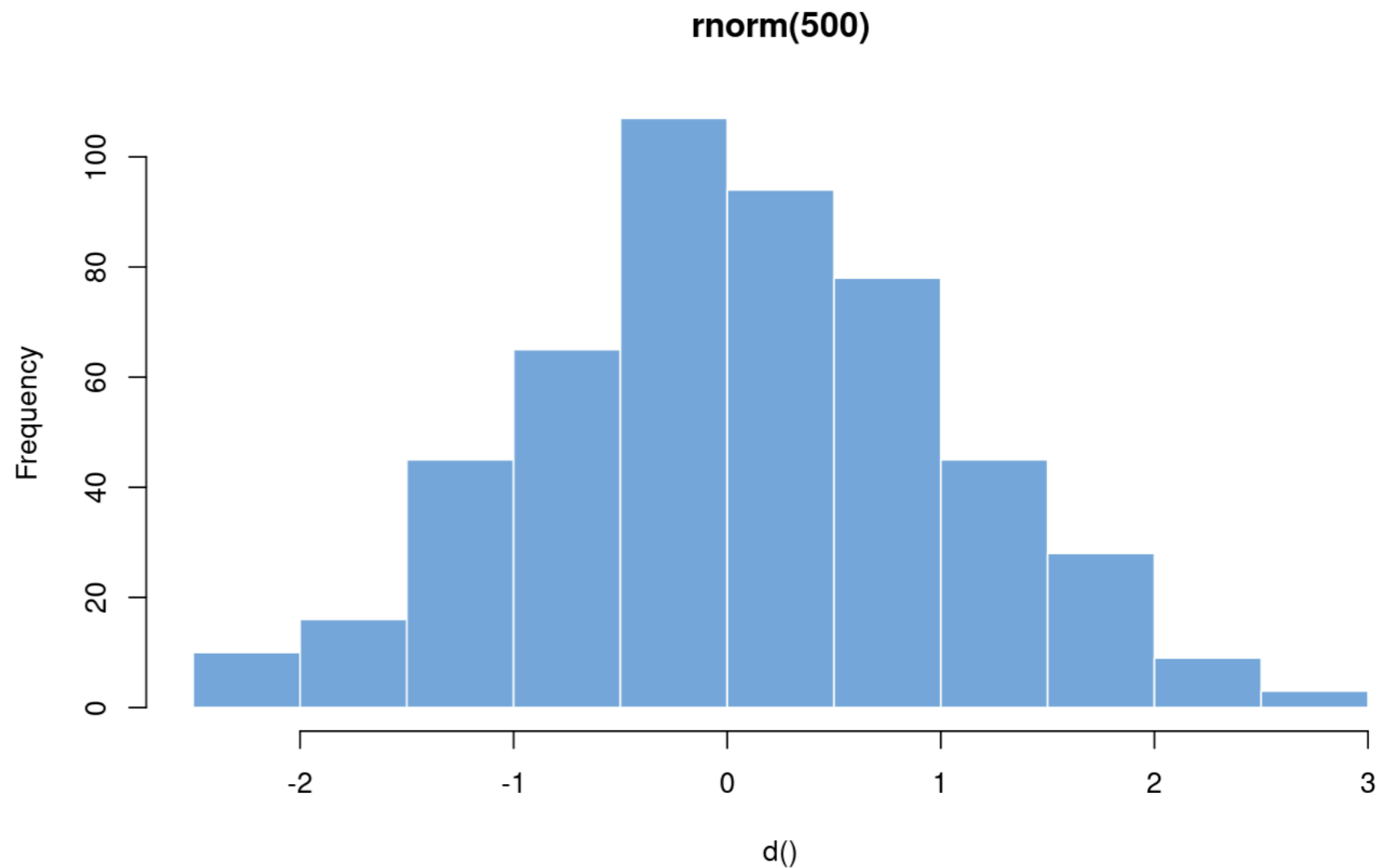
Number of observations:



Plot

Summary

Table



Shiny UI is powered by [Bootstrap](#)

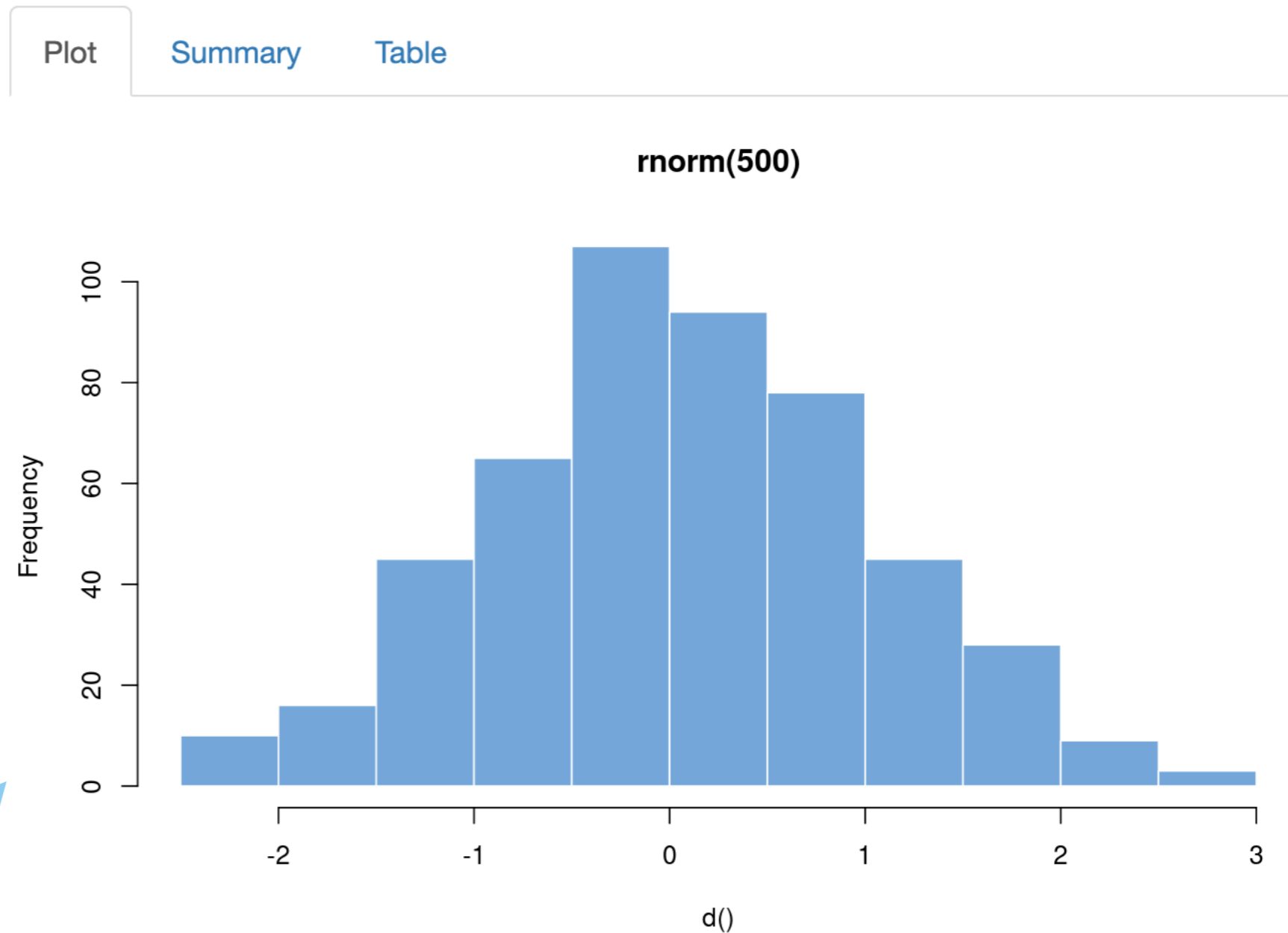
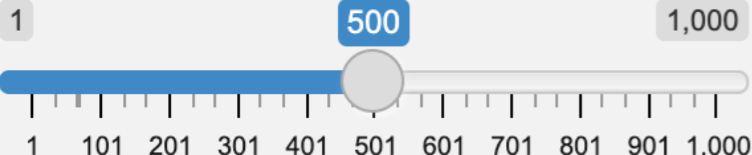
Tabsets

Distribution type:

- Normal
- Uniform
- Log-normal
- Exponential

Number of observations:

1 500 1,000



Bootstrap
grid layout

Shiny UI is powered by [Bootstrap](#)

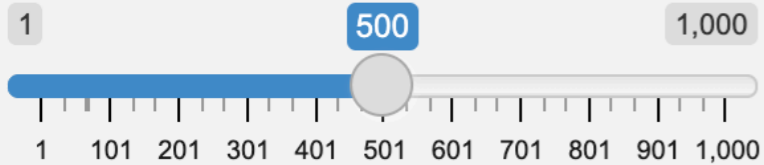
Smarter CSS defaults
(e.g., typography)

Tabsets

Distribution type:

- Normal
- Uniform
- Log-normal
- Exponential

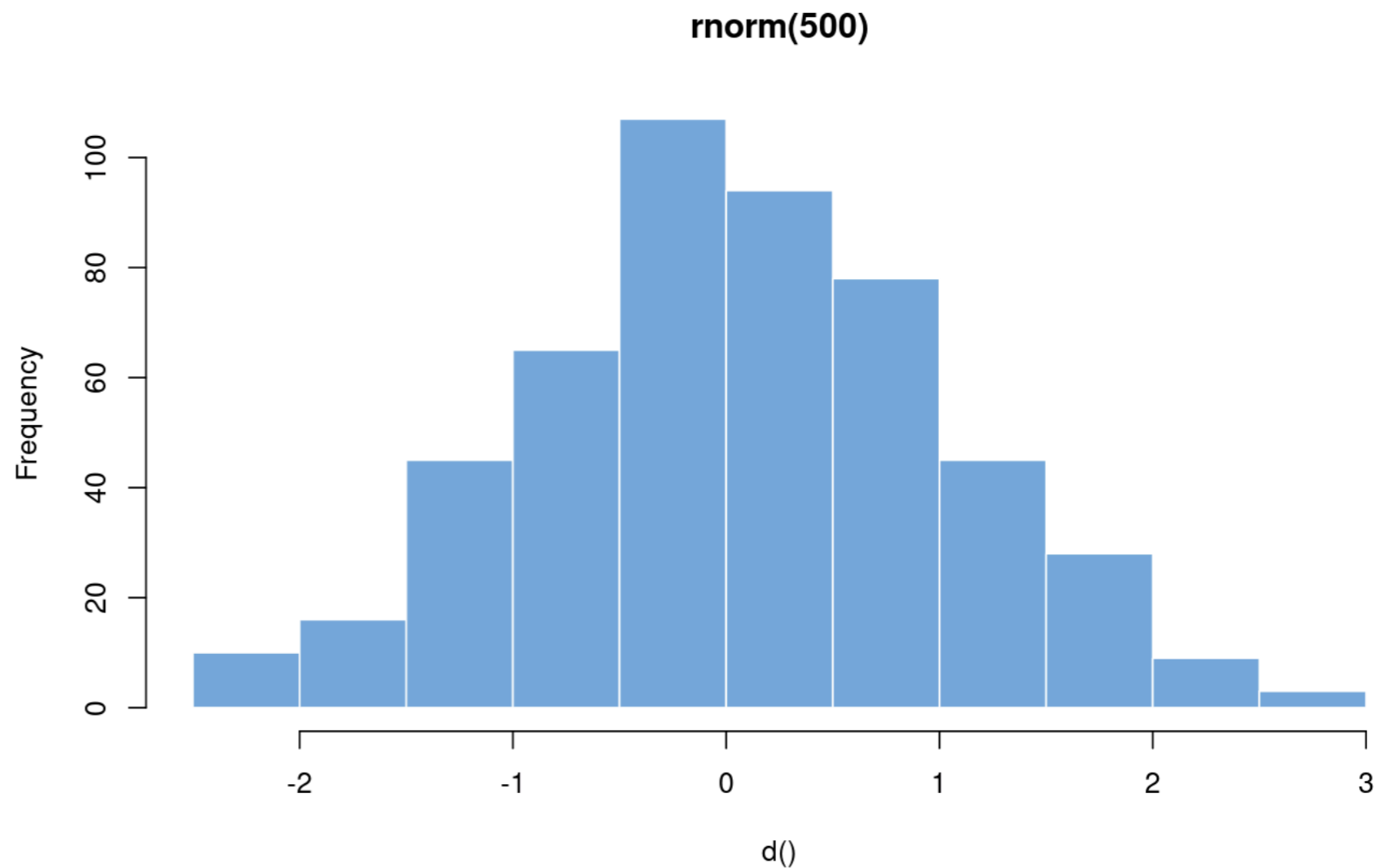
Number of observations:



Plot

Summary

Table



Shiny UI is powered by [Bootstrap](#)

- Widely used CSS/JavaScript framework, [estimated](#) to be used in **>1/4 of websites**

Shiny UI is powered by [Bootstrap](#)

- Widely used CSS/JavaScript framework, [estimated](#) to be used in **>1/4 of websites**
- Shiny defaults to Bootstrap 3, which was released in 2013 and EOL in 2019

Shiny UI is powered by [Bootstrap](#)

- Widely used CSS/JavaScript framework, [estimated](#) to be used in **>1/4 of websites**
- Shiny defaults to Bootstrap 3, which was released in 2013 and EOL in 2019
- Changing the default to Bootstrap 4+ would break lots of existing apps
 - We can shim *our* code, but not *your* code

The `{bslib}` R package

- Seamlessly upgrade from Bootstrap 3 to 4+

The `{bslib}` R package

- Seamlessly upgrade from Bootstrap 3 to 4+
- Easily customize main colors & fonts
 - Create them in interactively real-time!

The `{bslib}` R package

- Seamlessly upgrade from Bootstrap 3 to 4+
- Easily customize main colors & fonts
 - Create them in interactively real-time!
- Serves as a foundation for Shiny, RMarkdown, Quarto, and extensions.
 - Including `{flexdashboard}`, `{pkgdown}`, & `{bookdown}`

Start using {bslib} with Shiny

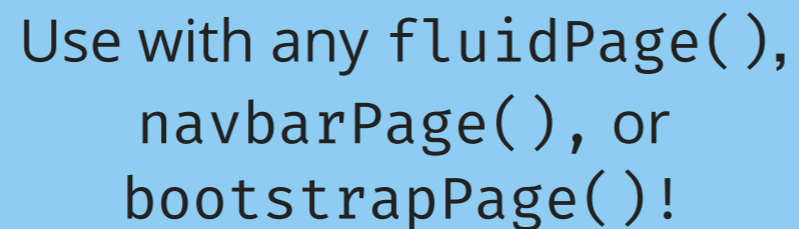
```
library(shiny)
```

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

Start using {bslib} with Shiny

```
library(shiny)
```

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

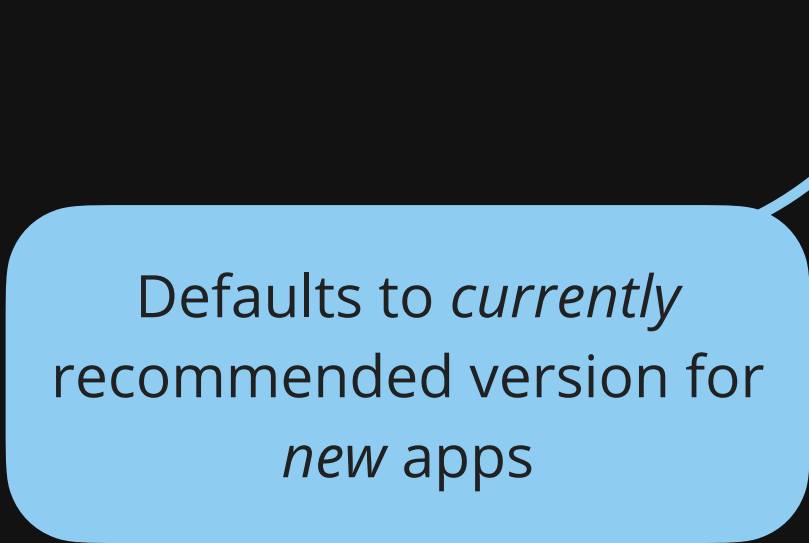


Use with any `fluidPage()`,
`navbarPage()`, or
`bootstrapPage()`!

Bootstrap versioning

```
library(shiny)
```

```
ui ← fluidPage(  
  theme = bslib::bs_theme(version = ),  
  ...  
)
```

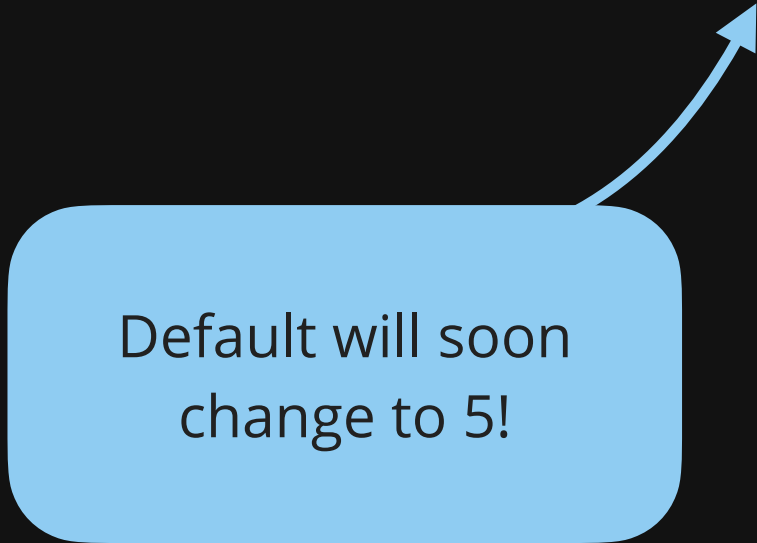


Defaults to *currently*
recommended version for
new apps

Bootstrap versioning

```
library(shiny)
```

```
ui ← fluidPage(  
  theme = bslib::bs_theme(version = 5),  
  ...  
)
```



Default will soon
change to 5!

Bootstrap versioning

```
library(shiny)
```

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

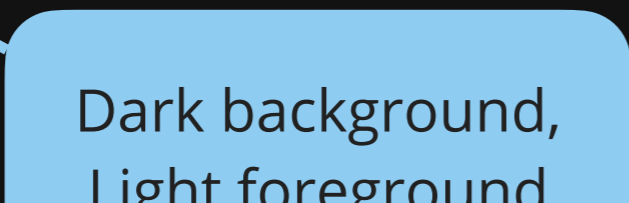


Specify version for better
reproducibility

Custom base colors

```
library(shiny)
```

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



Dark background,
Light foreground

The diagram consists of a light blue rounded rectangular callout box containing the text "Dark background, Light foreground". Two curved arrows originate from the top corners of this box and point towards the **bg** and **fg** parameters in the code snippet above.

Custom accent colors

```
library(shiny)
```

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

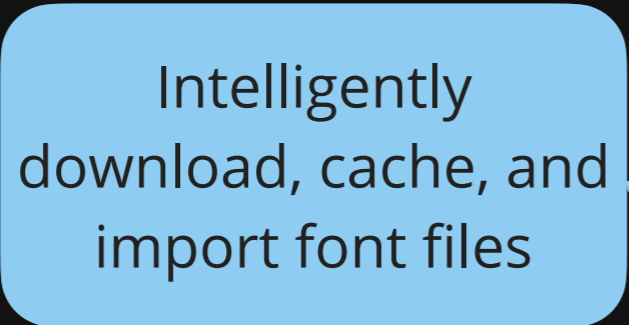


The most important
accent colors

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

Targeted theming

```
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 Bootstrap Sass variables are available

Preview a theme

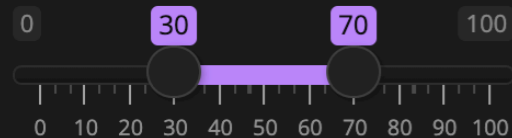
```
bslib::bs_theme_preview(theme)
```

Theme demo Inputs Plots Tables Notifications Fonts Options

inputPanel()

wellPanel()

sliderInput()



selectizeInput()

selectizeInput(multiple=T)

dateInput()

dateRangeInput()

 to

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

P Primary

Secondary (default)

✓ Success

i Info

! warning

⚠ Danger

🌙 Dark

☀ Light

Real-time theming

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

Theme demo Inputs Plots Tables Notifications Fonts Options

inputPanel() wellPanel()

sliderInput()



selectizeInput()

selectizeInput(multiple=T)

dateInput()

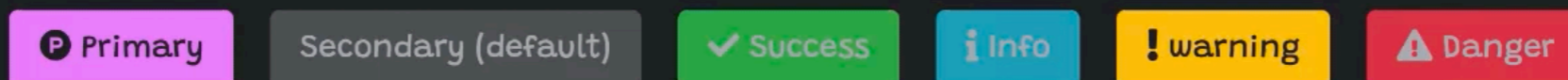
dateRangeInput()

 to

Below are the values bound to each input widget above

```
List of 5
 $ sliderInput      : int [1:2] 30 70
 $ selectizeInput   : chr "AL"
 $ selectizeMultiInput: NULL
 $ dateInput        : Date[1:1], format: "2020-10-06"
 $ dateRangeInput   : Date[1:2], format: "2020-10-06" ...
```

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



Theme customizer

Basic colors

Background color

#202123

The background color for the page

Foreground color

#B8BCC2

The color for foreground elements, including text

Accent colors

Fonts

Options

Spacing

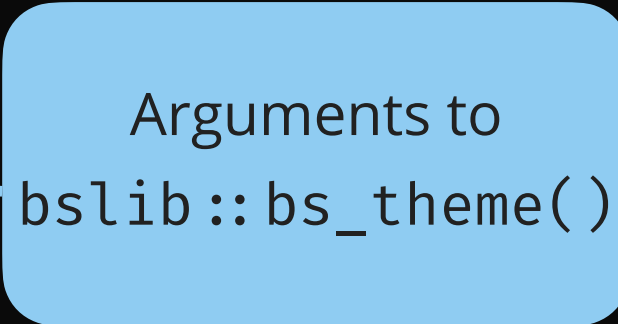
Changes captured as code

```
> bs_theme_preview(theme)
```

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

Using `{bslib}` with R Markdown

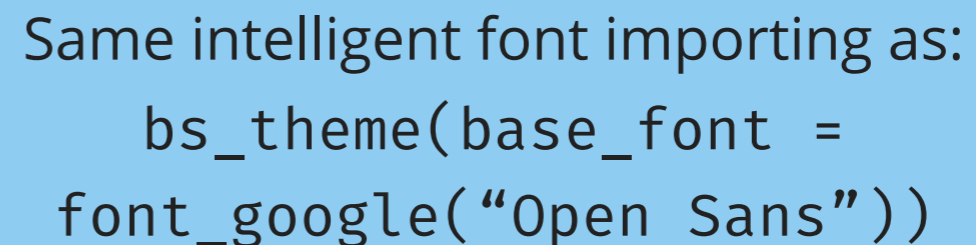
```
---  
output:  
  html_document:  
    theme:  
      bg: "#121212"  
      fg: "#E4E4E4"  
      primary: "#BB86FC"  
---
```



Arguments to
`bslib::bs_theme()`

Using `{bslib}` with R Markdown


```
---  
output:  
  html_document:  
    theme:  
      bg: "#121212"  
      fg: "#E4E4E4"  
      primary: "#BB86FC"  
      base_font:  
        google: Open Sans  
---
```



Same intelligent font importing as:
`bs_theme(base_font =
font_google("Open Sans"))`

Using `{bslib}` with R Markdown

```
---  
output:  
  flexdashboard::flex_dashboard:  
    theme:  
      bg: "#121212"  
      fg: "#E4E4E4"  
      primary: "#BB86FC"  
      base_font:  
        google: Open Sans  
---
```

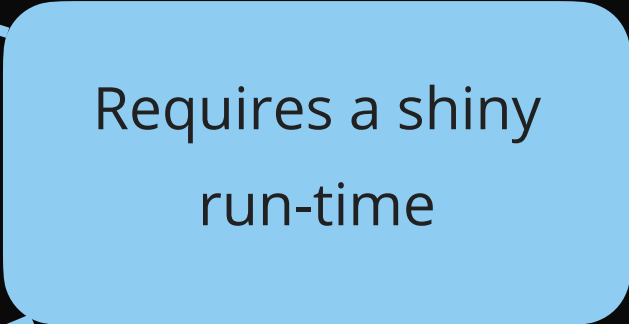


`{flexdashboard}`,
`{pkgdown}`, and
`{bookdown}` support the
same/similar usage

Real-time theme Rmd docs

```
---  
output:  
  flexdashboard::flex_dashboard:  
    theme:  
      bg: "#121212"  
      fg: "#E4E4E4"  
      primary: "#BB86FC"  
runtime: shiny
```

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



Requires a shiny  
run-time

# Real-time theme flexdashboard

Flexdashboard theming demo Components Storyboard Cards About

Set contact rate

Choose a state

Choose a date  to

1 primary

2 info

3 success

4 warning

5 danger

Success Rate

Warning metric

Basic Table Interactive Table

|                   | mpg  | cyl | disp  | hp  | drat | wt    | q     |   |   |   |   |
|-------------------|------|-----|-------|-----|------|-------|-------|---|---|---|---|
| Mazda RX4         | 21.0 | 6   | 160.0 | 110 | 3.90 | 2.620 | 16    |   |   |   |   |
| Mazda RX4 Wag     | 21.0 | 6   | 160.0 | 110 | 3.90 | 2.875 | 17    |   |   |   |   |
| Datsun 710        | 22.8 | 4   | 108.0 | 93  | 3.85 | 2.320 | 18    |   |   |   |   |
| Hornet 4 Drive    | 21.4 | 6   | 258.0 | 110 | 3.08 | 3.215 | 19    |   |   |   |   |
| Hornet Sportabout | 18.7 | 8   | 360.0 | 175 | 3.15 | 3.440 | 17    |   |   |   |   |
| Valiant           | 18.1 | 6   | 225.0 | 105 | 2.76 | 3.460 | 20    |   |   |   |   |
| Duster 360        | 14.3 | 8   | 360.0 | 245 | 3.21 | 3.570 | 15    |   |   |   |   |
| Merc 240D         | 24.4 | 4   | 146.7 | 62  | 3.69 | 3.190 | 20.00 | 1 | 0 | 4 | 2 |
| Merc 230          | 22.8 | 4   | 140.8 | 95  | 3.92 | 3.150 | 22.90 | 1 | 0 | 4 | 2 |
| Merc 280          | 19.2 | 6   | 167.6 | 123 | 3.92 | 3.440 | 18.30 | 1 | 0 | 4 | 4 |
| Merc 280C         | 17.8 | 6   | 167.6 | 123 | 3.92 | 3.440 | 18.90 | 1 | 0 | 4 | 4 |

Theme customizer

Main colors

Accent colors

Primary color

Links and highlighted navigation

Secondary

Success

Info

Warning

Danger

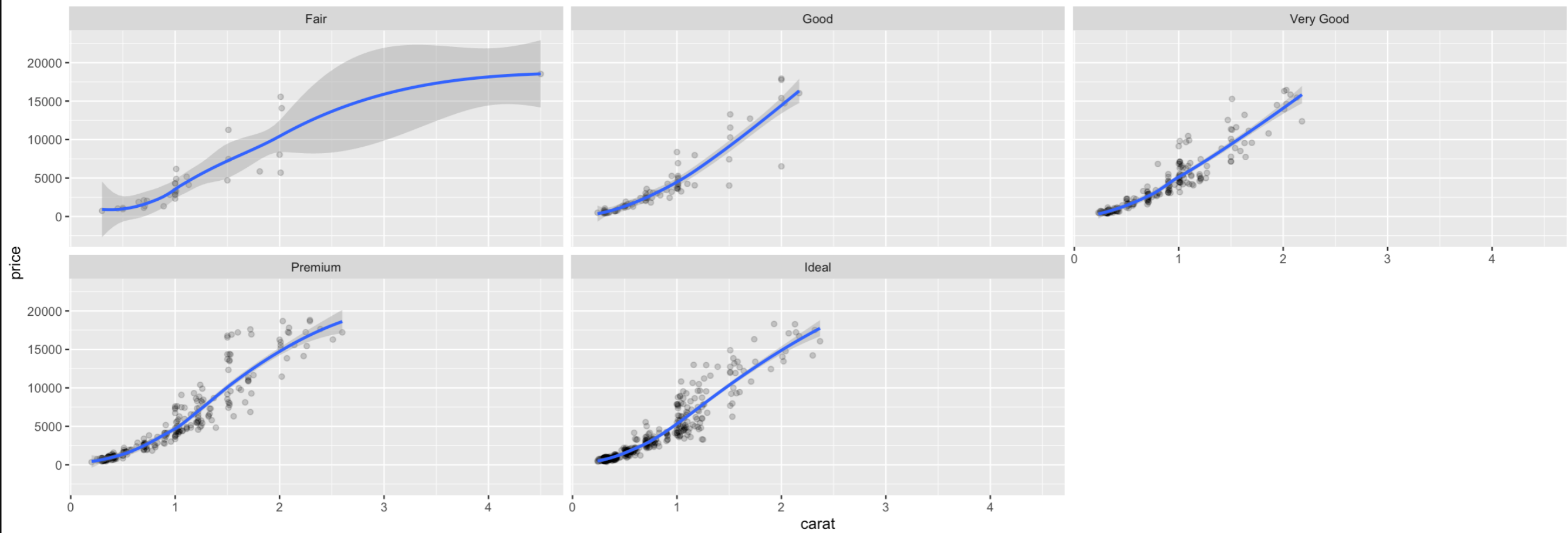
# Plots don't reflect theme 🙄

Theme demo Inputs Plots Tables Notifications Fonts Options

Choose an example

GeomSmooth

Diamond price by carat and cut



# Plots don't reflect theme 🙄

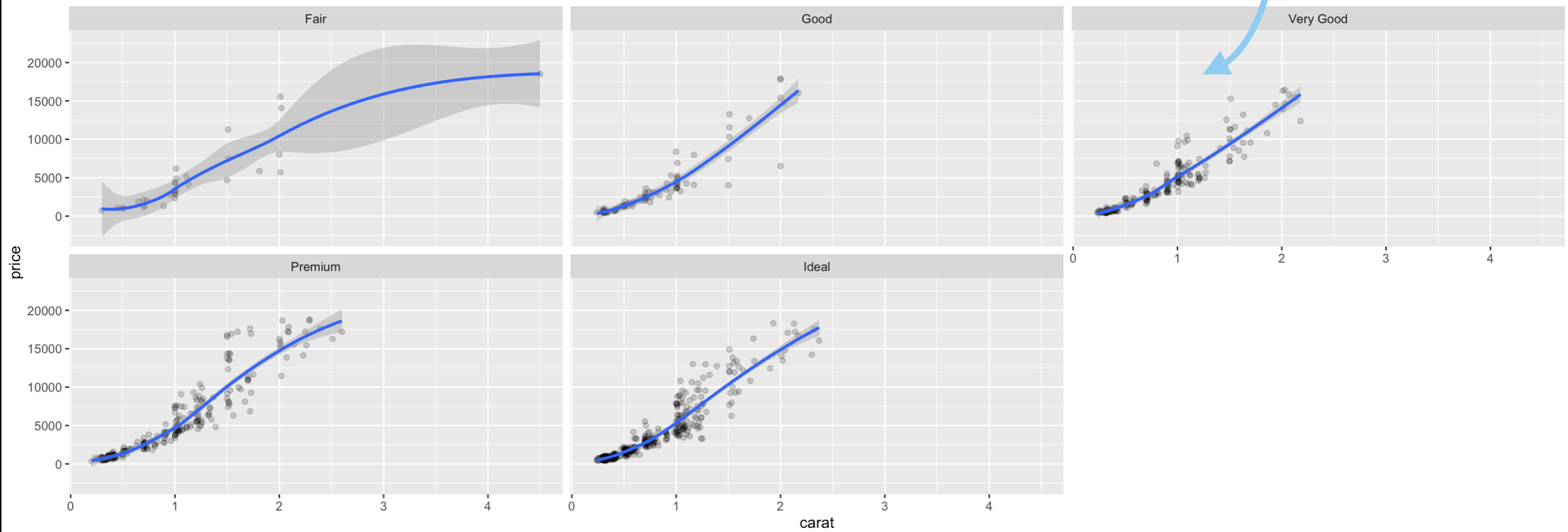
Theme demo Inputs Plots Tables Notifications Fonts Options

Choose an example

GeomSmooth

Plots rendered by R,  
not the browser!!

Diamond price by carat and cut



# thematic :: thematic\_shiny()



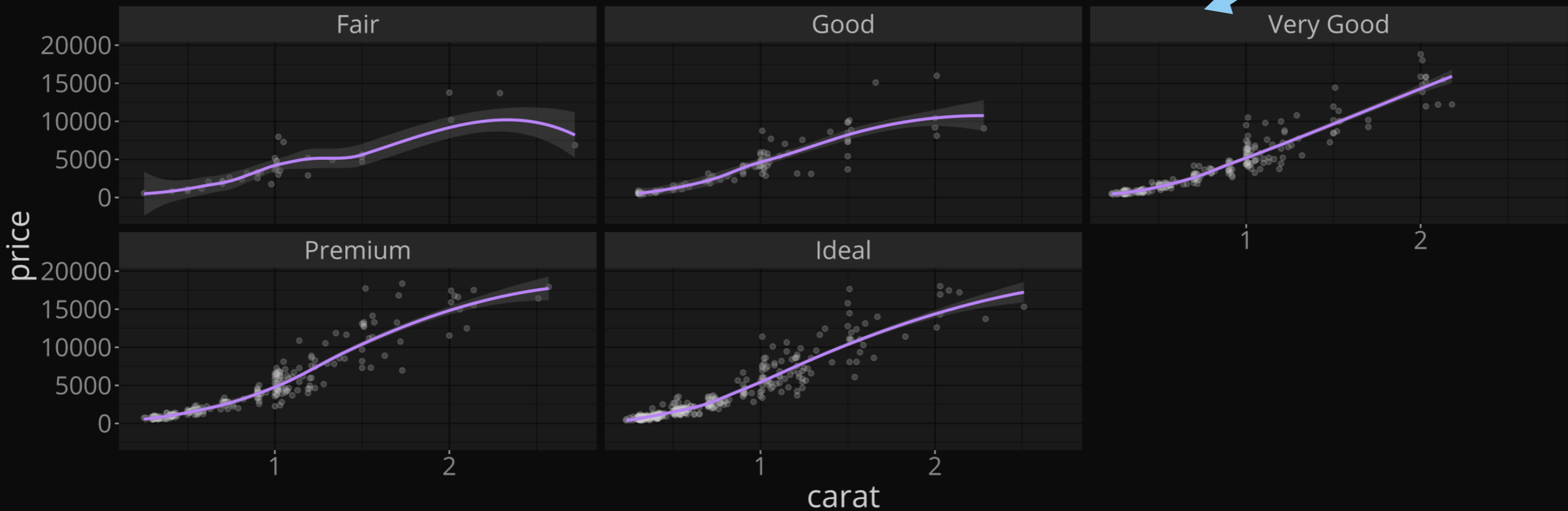
Theme demo Inputs Plots Tables Notifications Fonts Options

Choose an example

GeomSmooth

Plot styling defaults now informed by CSS

## Diamond price by carat and cut



# The `{thematic}` R package

- Auto theme R plots in Shiny, R Markdown, and RStudio.
- Provides a simplified interface for theming `{ggplot2}`, `{lattice}`, and `{base}` graphics in any R runtime.



[rstudio.github.io/thematic](https://rstudio.github.io/thematic)

# In summary

- Use `shiny::bindCache()` to avoid redundant computation
- Use `bslib::bs_theme()` to customize Bootstrap
  - In R Markdown, provide arguments to theme:
- Use `bslib::bs_themer()` to customize in real-time
- Translate CSS to R plots with `thematic_shiny()`

# Thank you!

Learn more:

[rstudio.github.io/bslib](https://rstudio.github.io/bslib)  
[rstudio.github.io/thematic](https://rstudio.github.io/thematic)

Slides:

[bit.ly/wb-shiny-2021](https://bit.ly/wb-shiny-2021)

Contact:

 @cpsievert

 [carson@rstudio.com](mailto:carson@rstudio.com)

 [cpsievert.me](https://cpsievert.me)