Tools for making faster and better looking Shiny apps

Carson Sievert
Software Engineer, RStudio
Slides bit.ly/wb-shiny-2021
@cpsievert

RStudio®

Shiny v1.6 major features

- Persistent caching via new `bindCache()` function
- Improved theming support via new `{bslib}` & `{thematic}` packages
- Accessibility improvements and many bug fixes

For details, see the blog post

Shiny v1.6 major features

- **Persistent caching via new `bindCache()` function**
- Improved theming support via new `{bslib}` & `{thematic}` packages
- Accessibility improvements and many bug fixes

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

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

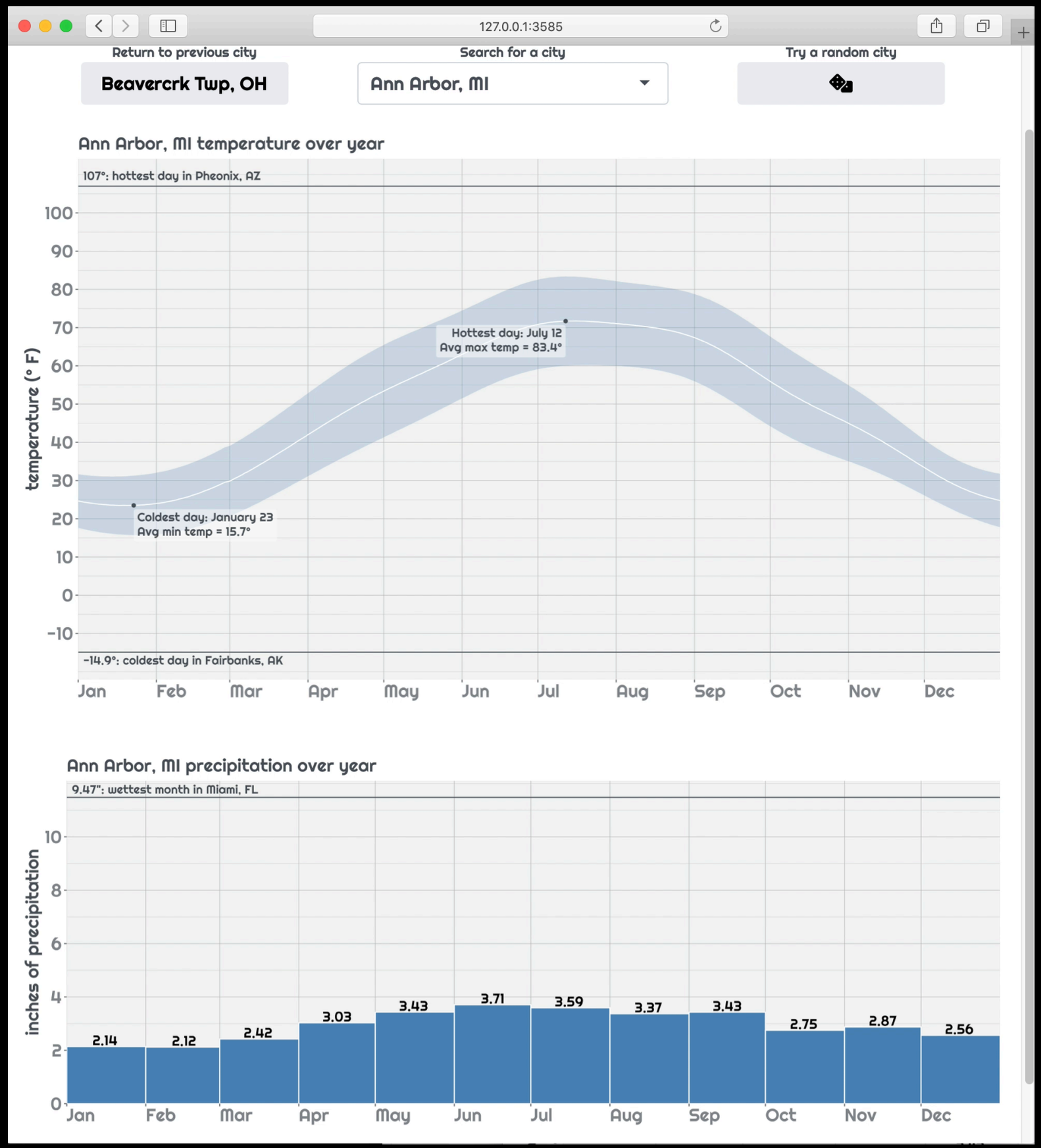
```
weatherData <- reactive({
    fetchData(input$city)
  }) %>%
  bindCache(input$city)
```

```
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())
```

See more in the app story

Previous cities are instantaneous!



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

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

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

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

Configuring the cache

Size: Default cache size is 200 MB in memory.

Scope: By default, memory cache is shared across *all user sessions,* but may be scoped to each user.

Lifetime: A memory cache is discarded when the app exits or restarts. A disk-based cache can persist across app restarts, and can be shared among multiple processes with Connect or Shiny Server Pro.

For details, see bindCache()'s reference

Shiny v1.6 major features

- Persistent caching via new `bindCache()` function

- **Improved theming support via new `{bslib}` & `{thematic}` packages**

- Accessibility improvements and many bug fixes

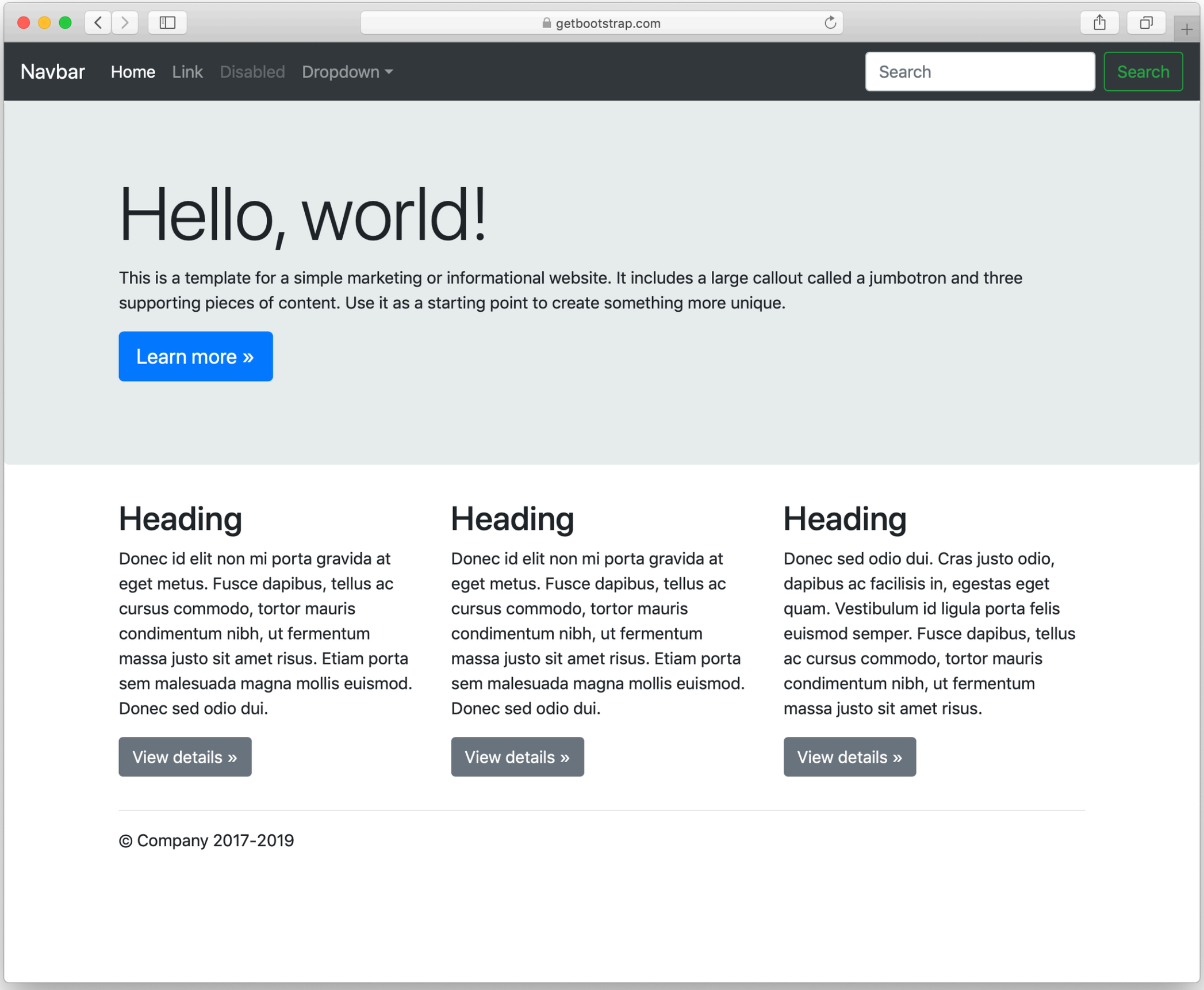
Before we get into usage, let's discuss Bootstrap CSS & Sass

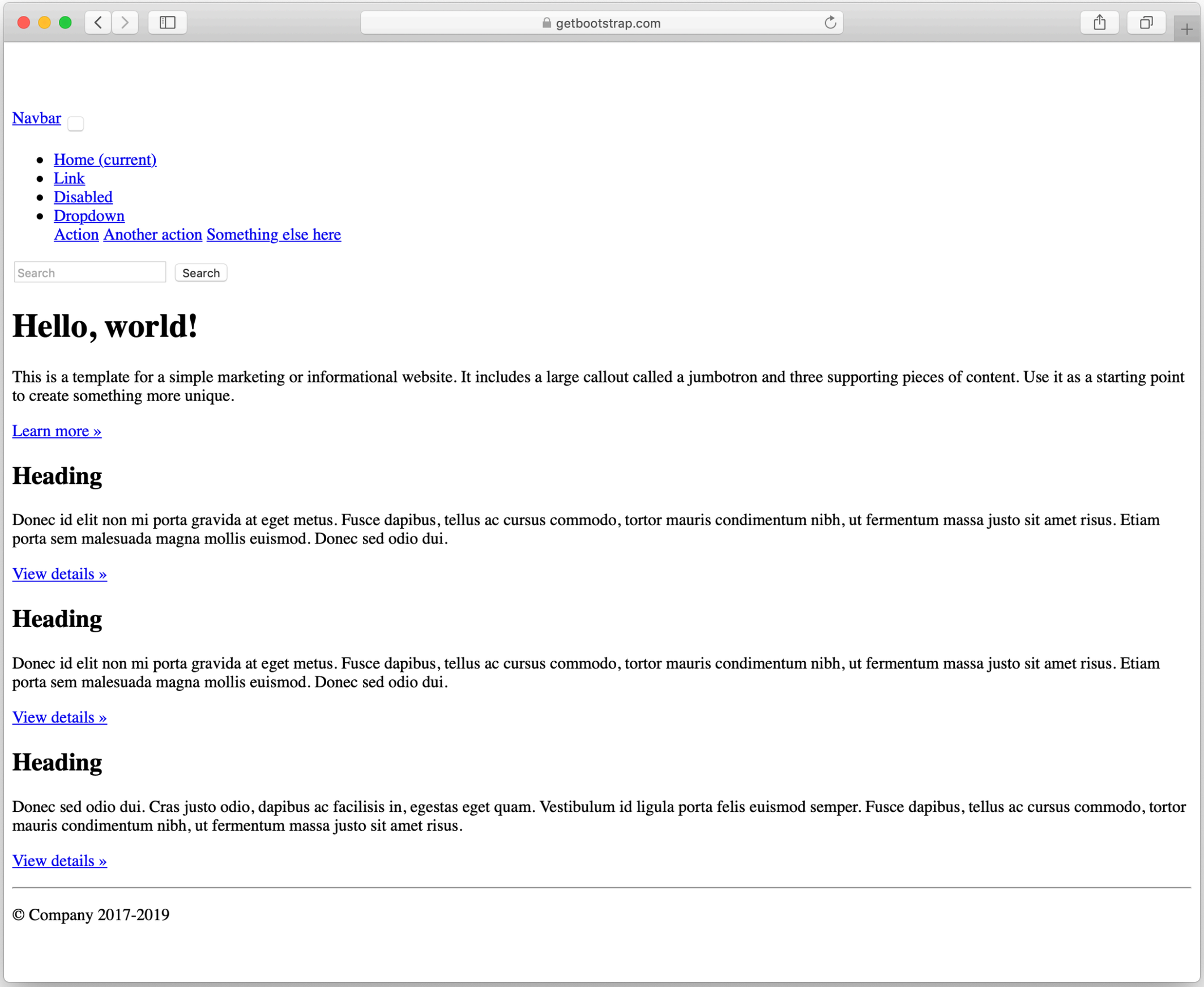
Shiny's default UI is powered by Bootstrap

- An open source CSS framework, originally started at Twitter

- Now ubiquitous: used by millions of websites & top 10 project on GitHub (by stars)

- Easy to customize (if you're a web programmer)
  - `{bslib}` package makes customization easy via R

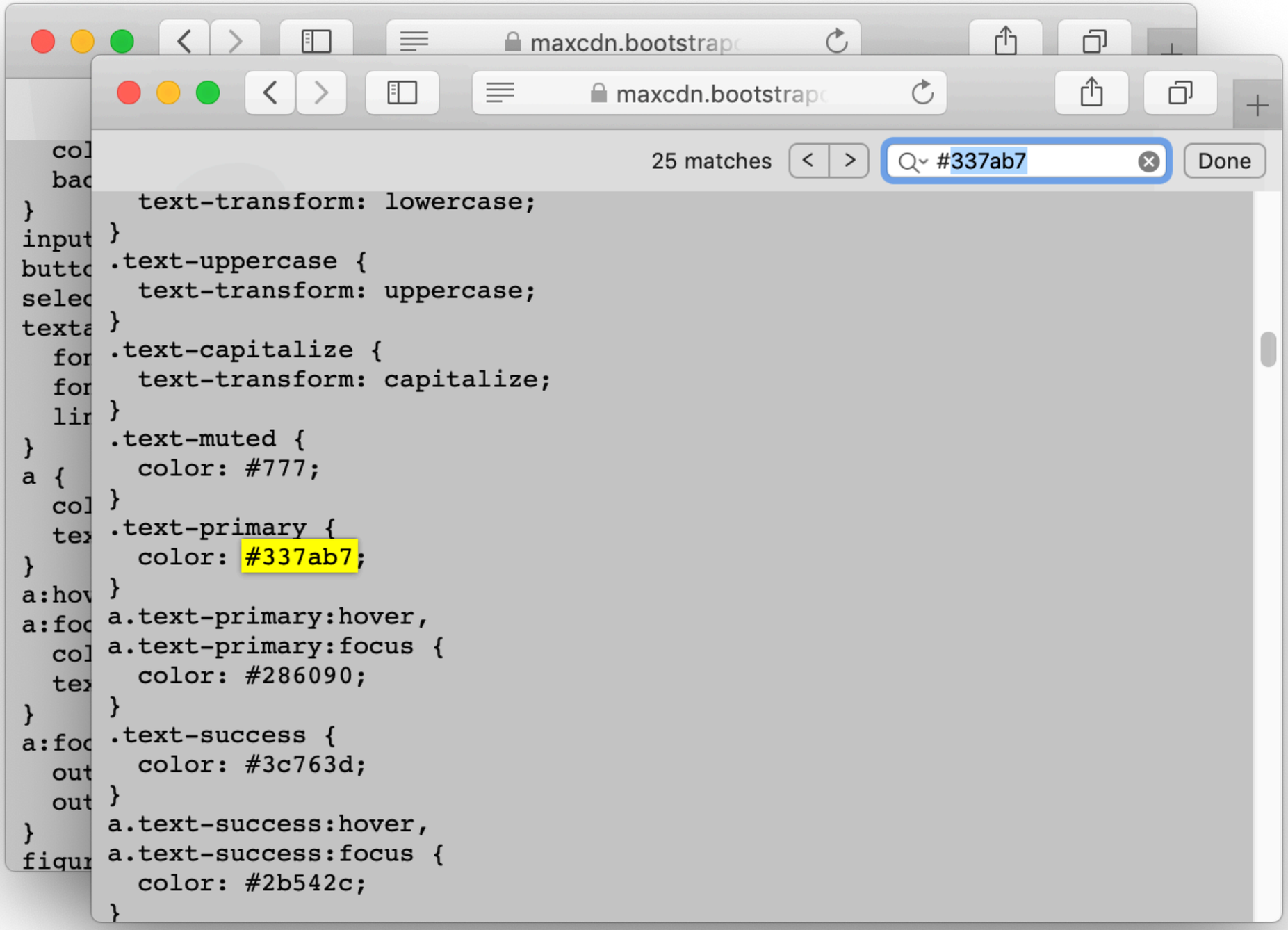

getbootstrap.com
Navbar
Home
Link
Disabled
Dropdown
Search
Search
Hello, world!
This is a template for a simple marketing or informational website. It includes a large callout called a jumbotron and three supporting pieces of content. Use it as a starting point to create something more unique.
Learn more »
Heading
Donec id elit non mi porta gravida at eget metus. Fusce dapibus, tellus ac cursus commodo, tortor mauris condimentum nibh, ut fermentum massa justo sit amet risus. Etiam porta sem malesuada magna mollis euismod. Donec sed odio dui.
View details »
Heading
Donec id elit non mi porta gravida at eget metus. Fusce dapibus, tellus ac cursus commodo, tortor mauris condimentum nibh, ut fermentum massa justo sit amet risus. Etiam porta sem malesuada magna mollis euismod. Donec sed odio dui.
View details »
Heading
Donec sed odio dui. Cras justo odio, dapibus ac facilisis in, egestas eget quam. Vestibulum id ligula porta felis euismod semper. Fusce dapibus, tellus ac cursus commodo, tortor mauris condimentum nibh, ut fermentum massa justo sit amet risus.
View details »
© Company 2017-2019

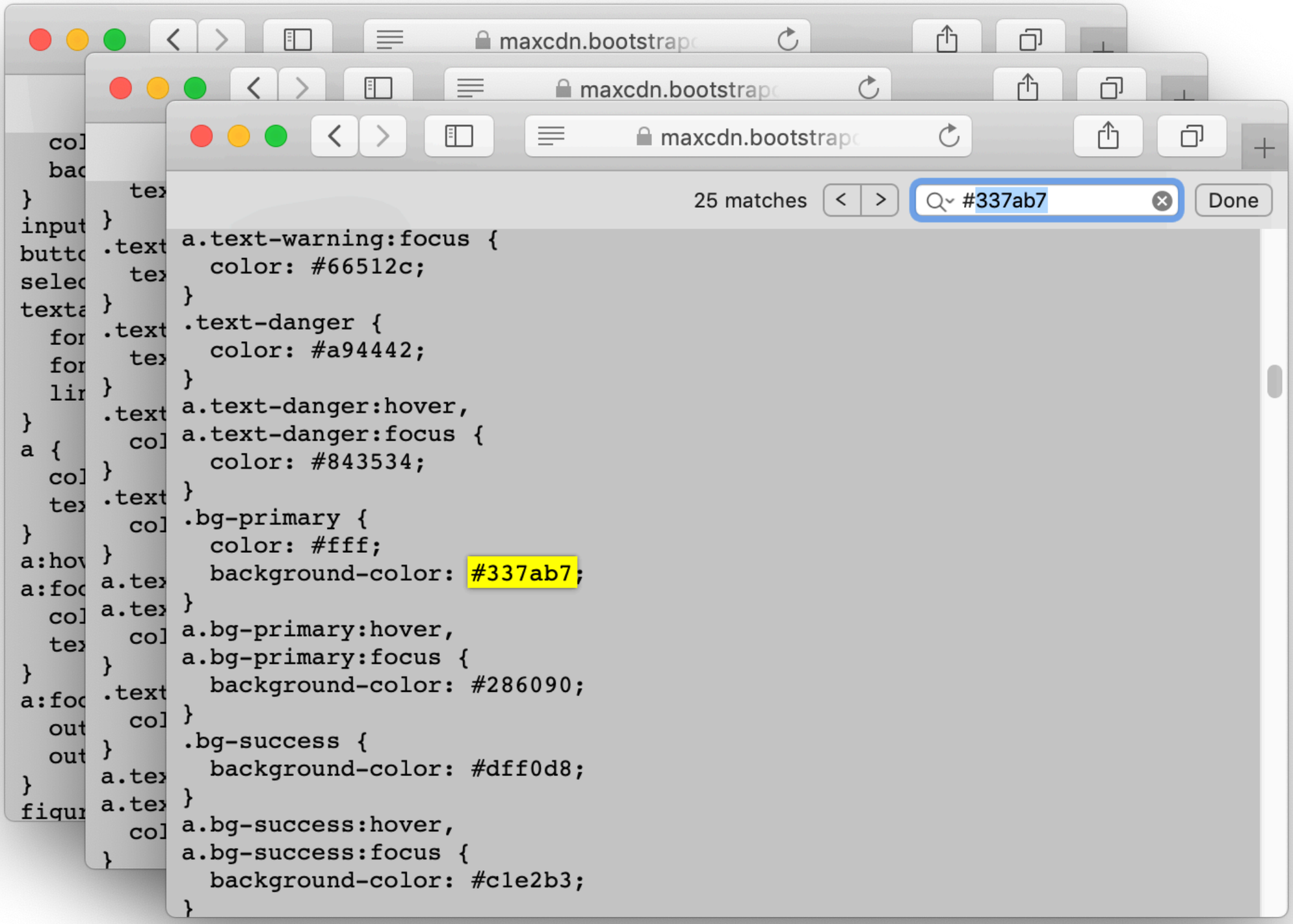

getbootstrap.com
Navbar
Home (current)
Link
Disabled
Dropdown
Action Another action Something else here
Search
Search
Hello, world!
This is a template for a simple marketing or informational website. It includes a large callout called a jumbotron and three supporting pieces of content. Use it as a starting point to create something more unique.
Learn more »
Heading
Donec id elit non mi porta gravida at eget metus. Fusce dapibus, tellus ac cursus commodo, tortor mauris condimentum nibh, ut fermentum massa justo sit amet risus. Etiam porta sem malesuada magna mollis euismod. Donec sed odio dui.
View details »
Heading
Donec id elit non mi porta gravida at eget metus. Fusce dapibus, tellus ac cursus commodo, tortor mauris condimentum nibh, ut fermentum massa justo sit amet risus. Etiam porta sem malesuada magna mollis euismod. Donec sed odio dui.
View details »
Heading
Donec sed odio dui. Cras justo odio, dapibus ac facilisis in, egestas eget quam. Vestibulum id ligula porta felis euismod semper. Fusce dapibus, tellus ac cursus commodo, tortor mauris condimentum nibh, ut fermentum massa justo sit amet risus.
View details »
© Company 2017-2019

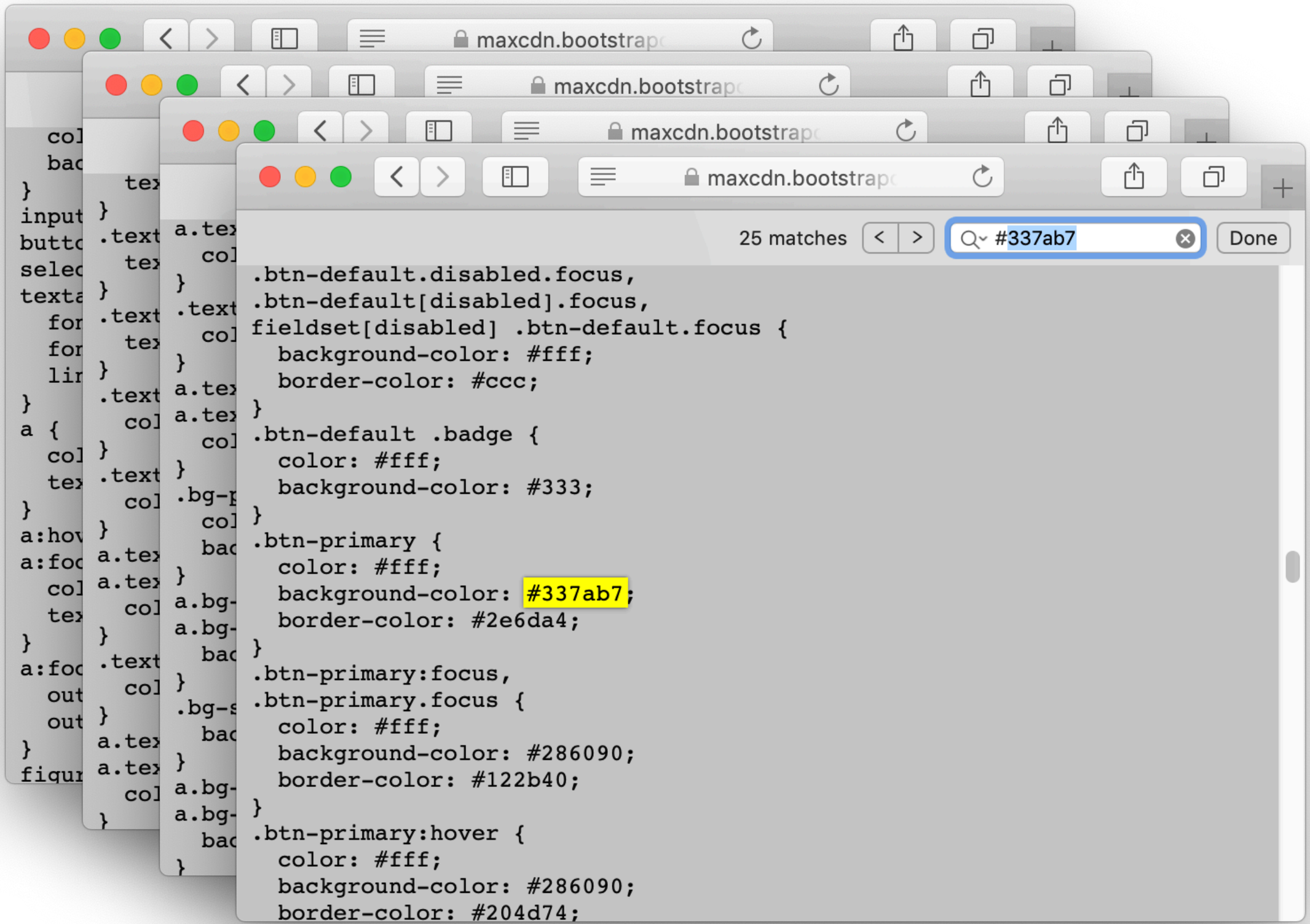
maxcdn.bootstrap

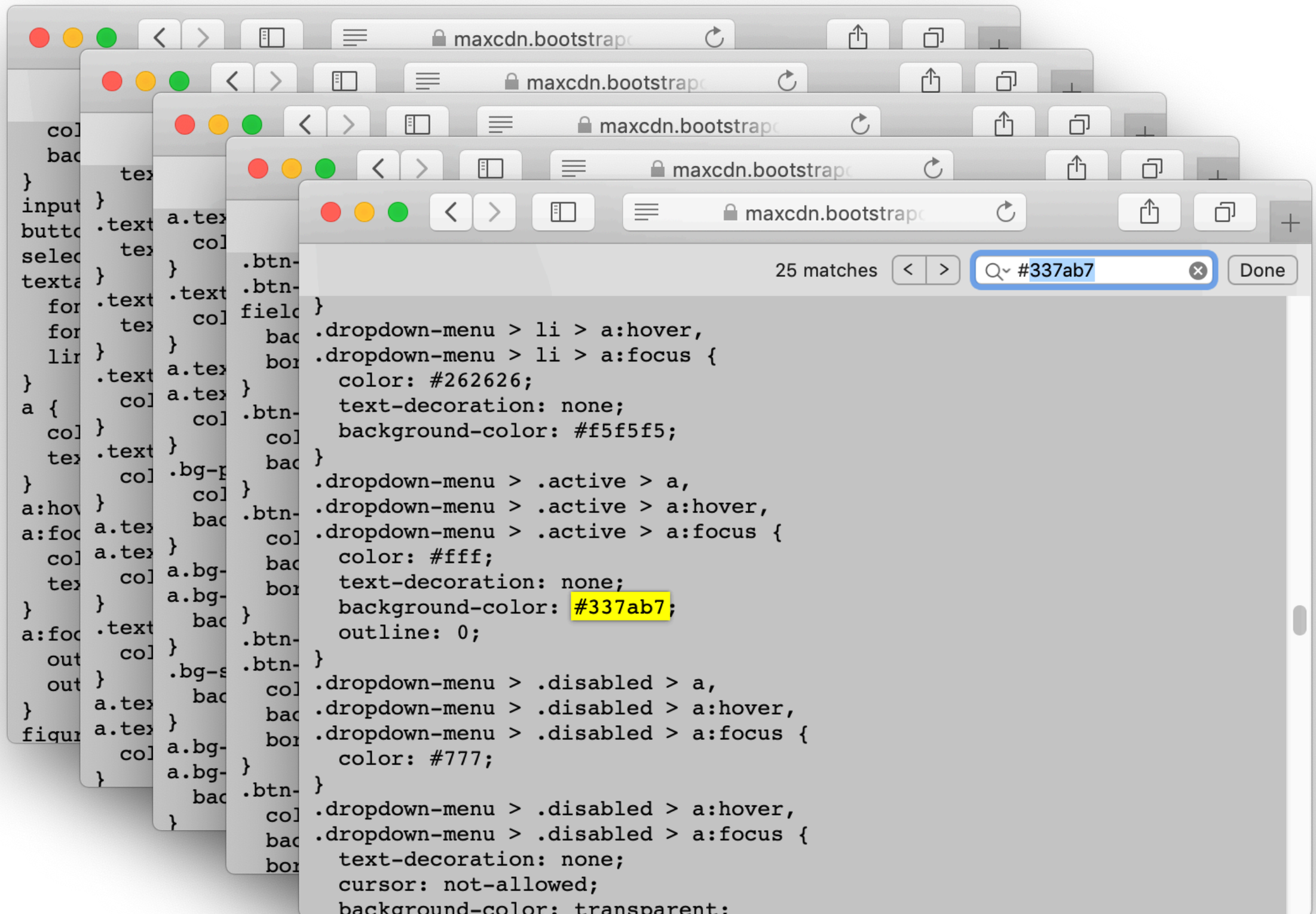
25 matches #337ab7 Done

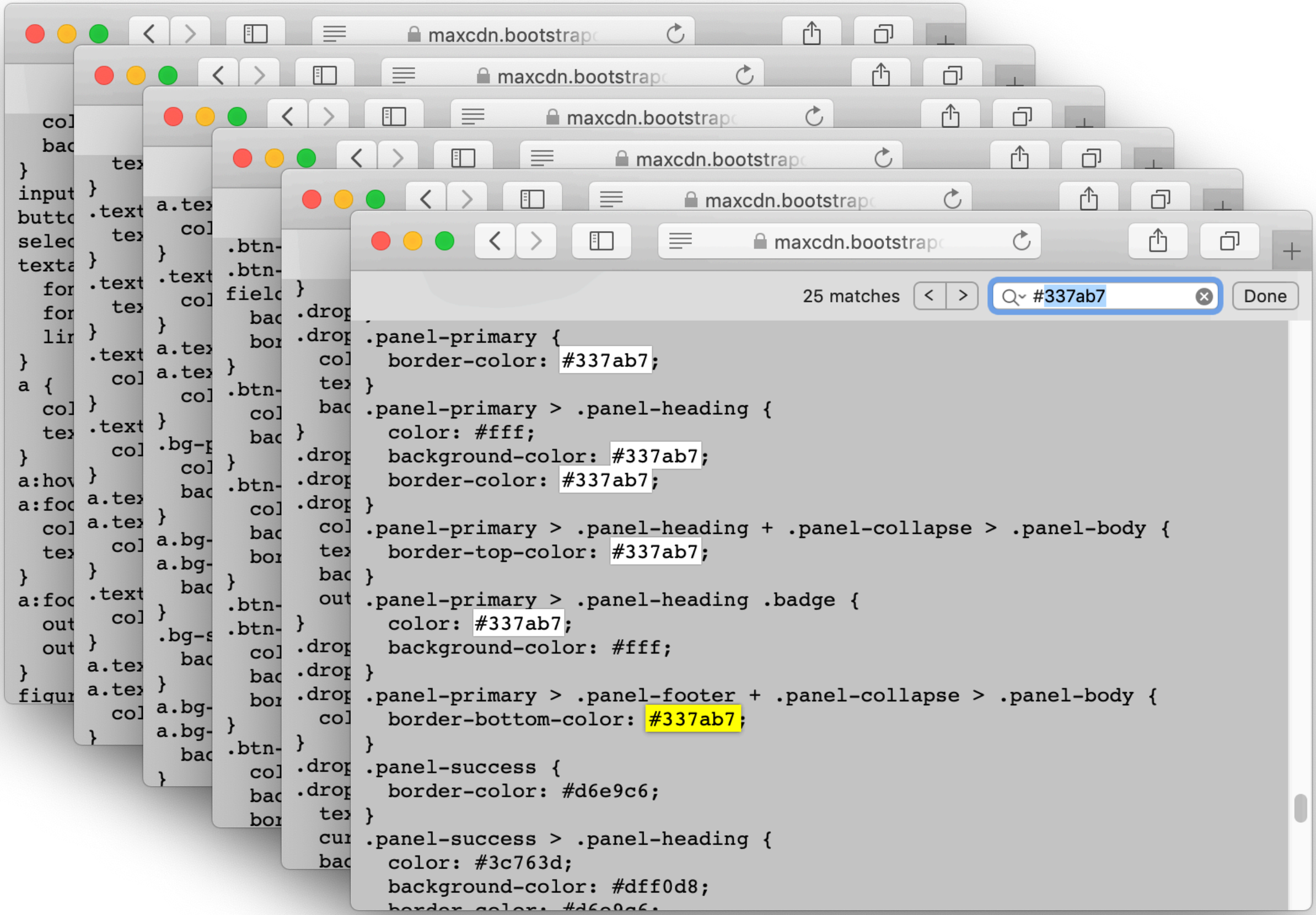
```
  color: #333;
  background-color: #fff;
}
input,
button,
select,
textarea {
  font-family: inherit;
  font-size: inherit;
  line-height: inherit;
}
a {
  color: #337ab7;
  text-decoration: none;
}
a:hover,
a:focus {
  color: #23527c;
  text-decoration: underline;
}
a:focus {
  outline: 5px auto -webkit-focus-ring-color;
  outline-offset: -2px;
}
figure {
```

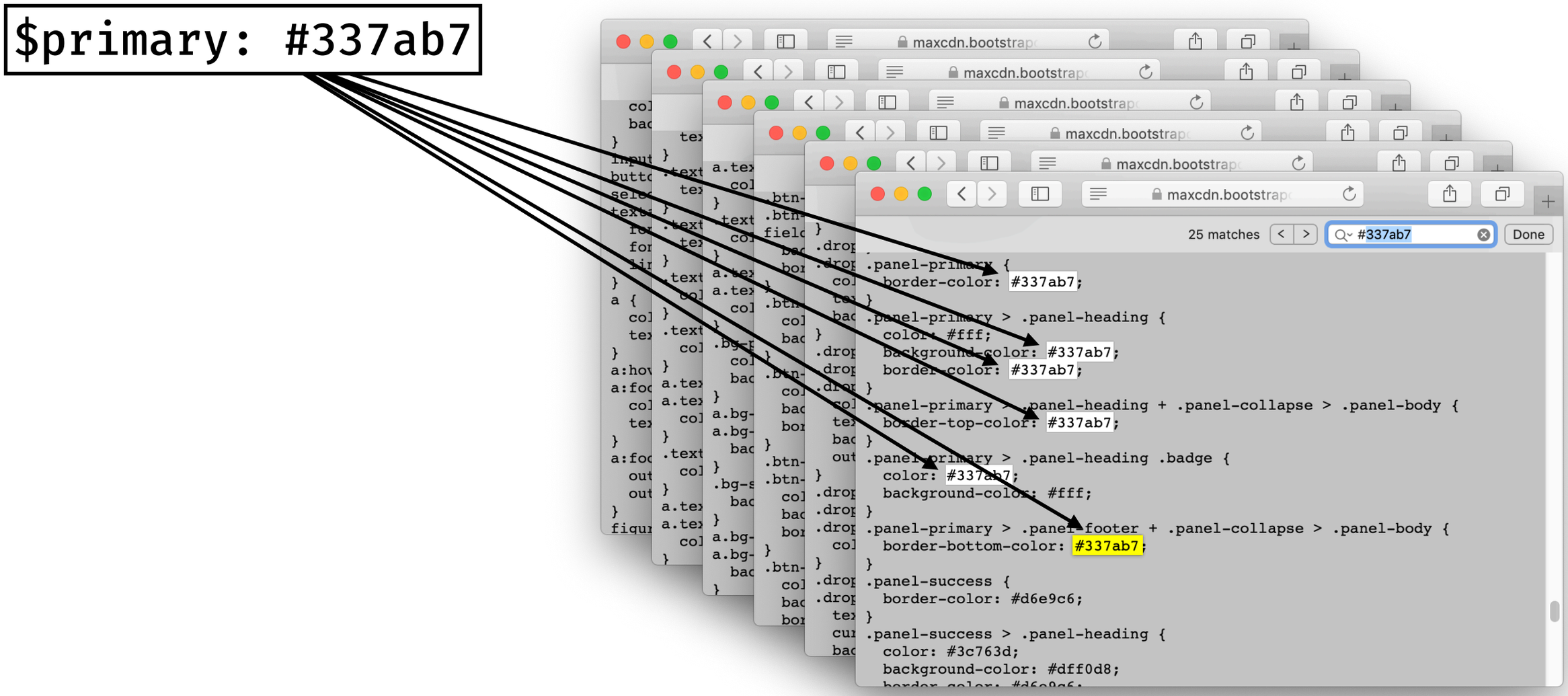

maxcdn.bootstrap
maxcdn.bootstrap
25 matches
#337ab7
Done
text-transform: lowercase;
}
.text-uppercase {
text-transform: uppercase;
}
.text-capitalize {
text-transform: capitalize;
}
.text-muted {
color: #777;
}
.text-primary {
color: #337ab7;
}
a.text-primary:hover,
a.text-primary:focus {
color: #286090;
}
.text-success {
color: #3c763d;
}
a.text-success:hover,
a.text-success:focus {
color: #2b542c;
}


maxcdn.bootstrap
25 matches
#337ab7
Done
a.text-warning:focus {
color: #66512c;
}
.text-danger {
color: #a94442;
}
a.text-danger:hover,
a.text-danger:focus {
color: #843534;
}
.bg-primary {
color: #fff;
background-color: #337ab7;
}
a.bg-primary:hover,
a.bg-primary:focus {
background-color: #286090;
}
.bg-success {
background-color: #dff0d8;
}
a.bg-success:hover,
a.bg-success:focus {
background-color: #c1e2b3;
}


maxcdn.bootstrap
25 matches
#337ab7
Done
.btn-default.disabled.focus,
.btn-default[disabled].focus,
fieldset[disabled] .btn-default.focus {
background-color: #fff;
border-color: #ccc;
}
.btn-default .badge {
color: #fff;
background-color: #333;
}
.btn-primary {
color: #fff;
background-color: #337ab7;
border-color: #2e6da4;
}
.btn-primary:focus,
.btn-primary.focus {
color: #fff;
background-color: #286090;
border-color: #122b40;
}
.btn-primary:hover {
color: #fff;
background-color: #286090;
border-color: #204d74;


maxcdn.bootstrap
25 matches
#337ab7
Done
}
.dropdown-menu > li > a:hover,
.dropdown-menu > li > a:focus {
color: #262626;
text-decoration: none;
background-color: #f5f5f5;
}
.dropdown-menu > .active > a,
.dropdown-menu > .active > a:hover,
.dropdown-menu > .active > a:focus {
color: #fff;
text-decoration: none;
background-color: #337ab7;
outline: 0;
}
.dropdown-menu > .disabled > a,
.dropdown-menu > .disabled > a:hover,
.dropdown-menu > .disabled > a:focus {
color: #777;
}
.dropdown-menu > .disabled > a:hover,
.dropdown-menu > .disabled > a:focus {
text-decoration: none;
cursor: not-allowed;
background-color: transparent;

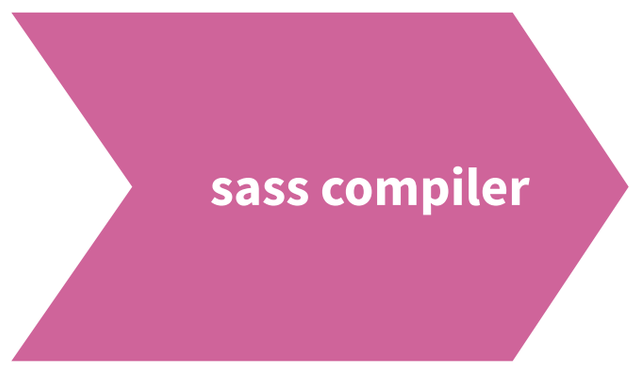

maxcdn.bootstrap
25 matches
#337ab7
Done
.panel-primary {
border-color: #337ab7;
}
.panel-primary > .panel-heading {
color: #fff;
background-color: #337ab7;
border-color: #337ab7;
}
.panel-primary > .panel-heading + .panel-collapse > .panel-body {
border-top-color: #337ab7;
}
.panel-primary > .panel-heading .badge {
color: #337ab7;
background-color: #fff;
}
.panel-primary > .panel-footer + .panel-collapse > .panel-body {
border-bottom-color: #337ab7;
}
.panel-success {
border-color: #d6e9c6;
}
.panel-success > .panel-heading {
color: #3c763d;
background-color: #dff0d8;


$primary: #337ab7
maxcdn.bootstrap
25 matches
#337ab7
Done
.panel-primary {
border-color: #337ab7;
}
.panel-primary > .panel-heading {
color: #fff;
background-color: #337ab7;
border-color: #337ab7;
}
.panel-primary > .panel-heading + .panel-collapse > .panel-body {
border-top-color: #337ab7;
}
.panel-primary > .panel-heading .badge {
color: #337ab7;
background-color: #fff;
}
.panel-primary > .panel-footer + .panel-collapse > .panel-body {
border-bottom-color: #337ab7;
}
.panel-success {
border-color: #d6e9c6;
}
.panel-success > .panel-heading {
color: #3c763d;
background-color: #dff0d8;

Sass is a better way to write CSS

styles.**scss**

```
$primary: #337ab7;

a {
  color: $primary;
  text-decoration: n
}

button {
  color: $primary;
}
```



styles.**css**

```
a {
  color: #337ab7;
  text-decoration: n
}

button {
  color: #337ab7;
}
```

Sass is a better way to write CSS

styles.**scss**

```
$primary: #337ab7;

a {
  color: $primary;
  text-decoration: n
}

button {
  color: $primary;
}
```

styles.**css**

```
a {
  color: #337ab7;
  text-decoration: n
}

button {
  color: #337ab7;
}
```

The {sass} R package provides a general Sass compiler for R

The {bslib} R package

- Easily customize Bootstrap Sass from R
  - Works with Shiny, R Markdown, `{flexdashboard}`, etc

- Seamless upgrading from Bootstrap 3 to 4 (and beyond)

- Easily leverage ‘pre-packaged’ Bootswatch themes

rstudio.github.io/bslib

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

```r
library(shiny)

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

Bootstrap 4 is *currently* recommended

Bootstrap versioning

```r
library(shiny)

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

Specify version for better reproducibility (default may change to 5, 6, etc)

Legacy apps

```r
library(shiny)

ui <- fluidPage(
  theme = bslib::bs_theme(version = 3),
    ...
)
```

Shiny currently defaults to Bootstrap 3

Bootswatch themes

```
library(shiny)

ui ← fluidPage(
  theme = bslib::bs_theme(
    bootswatch = “minty”
  ),
  …
)
```

21 Bootswatch 4 themes available!

bootswatch.com

Bootswatch THEMES HELP BLOG

Bootswatch Themes Download Help Blog

Cerulean

A calm blue sky

Primary Secondary Success Info Warning

Cerulean

A calm blue sky

PREVIEW DOWNLOAD

Bootswatch Themes Download Help Blog

Cosmo

An ode to Metro

Primary Secondary Success Info Warning Danger

Cosmo

An ode to Metro

PREVIEW DOWNLOAD

Bootswatch Themes Download Help Blog

Cyborg

Jet black and electric blue

Primary Secondary Success Info Warning

Cyborg

Jet black and electric blue

PREVIEW DOWNLOAD

Bootswatch Themes Download Help Blog

Darkly

Flatly in night mode

Primary Secondary Success Info Warning

Darkly

Flatly in night mode

Bootswatch Themes Download Help Blog

Flatly

Flat and modern

Primary Secondary Success Info Warning

Flatly

Flat and modern

BOOTSWATCH Themes Download Help Blog

Journal

Crisp like a new sheet of paper

Primary Secondary Success Info Warning Danger

Journal

Crisp like a new sheet of paper

Display a menu

Custom base colors

```
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

```r
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()

0 30 70 100

0 10 20 30 40 50 60 70 80 90 100

selectizeInput()

AL

selectizeInput(multiple=T)

AZ

dateInput()

2020-12-07

« 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

dateRangeInput()

2020-12-24 to 2020-12-31

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

Primary Secondary (default) Success Info warning Danger Dark Light

Real-time theming

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

Theme demo Inputs Plots Tables Notifications Fonts Options

inputPanel() wellPanel()

sliderInput()

0 30 70 100

0 10 20 30 40 50 60 70 80 90 100

selectizeInput()

AL

selectizeInput(multiple=T)

dateInput()

2020-10-06

dateRangeInput()

2020-10-06 to 2020-10-13

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" ...
```

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

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

Primary Secondary (default) Success Info warning Danger

Changes captured as code

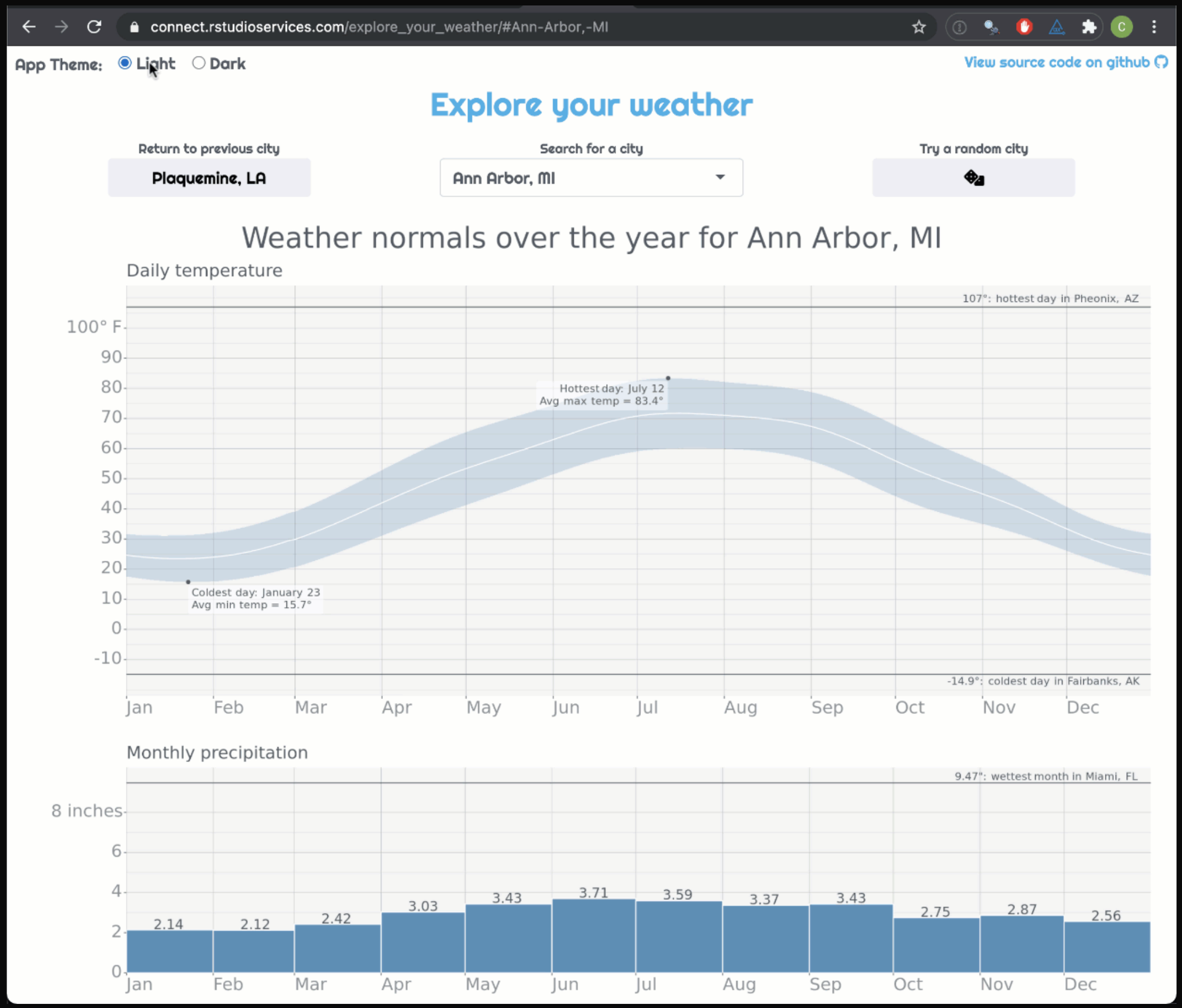
```
> bs_theme_preview(theme)

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

Real-time theme your app

```
library(shiny)

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

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

Requires Bootstrap 4

Implement your own theming widget



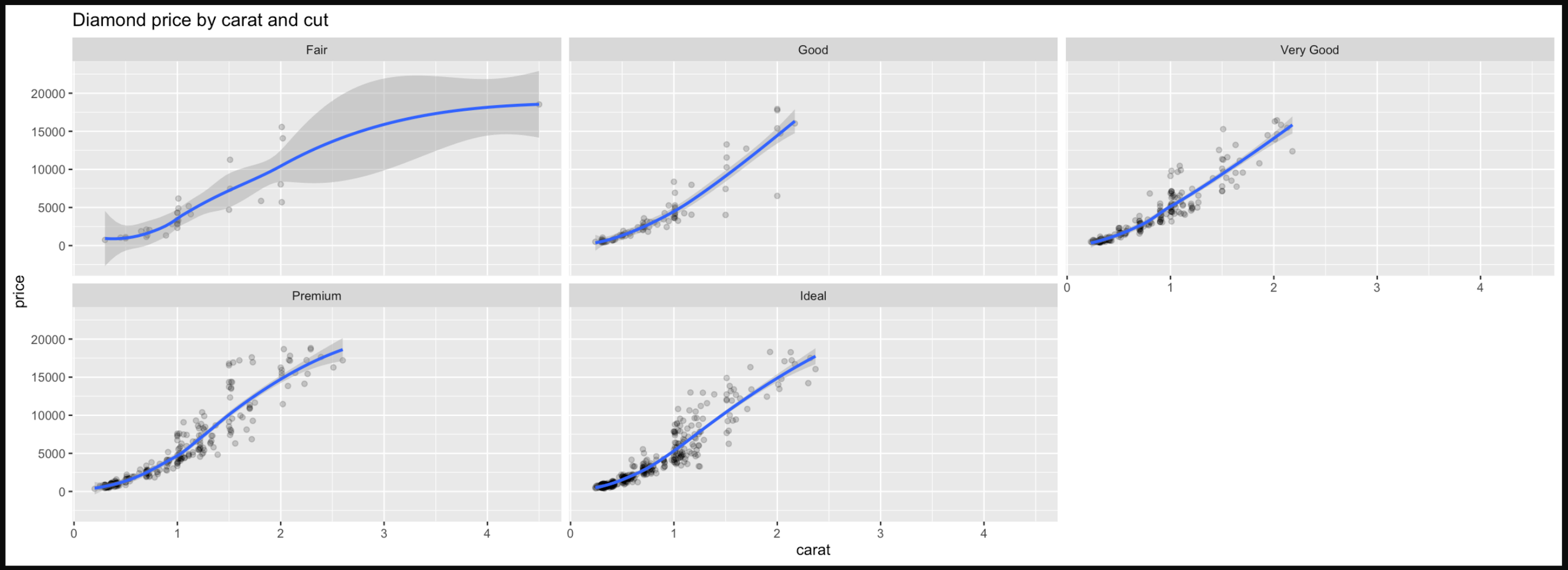
Minimal example

Plots don’t reflect theme 😞

Theme demo Inputs Plots Tables Notifications Fonts Options

Choose an example

GeomSmooth



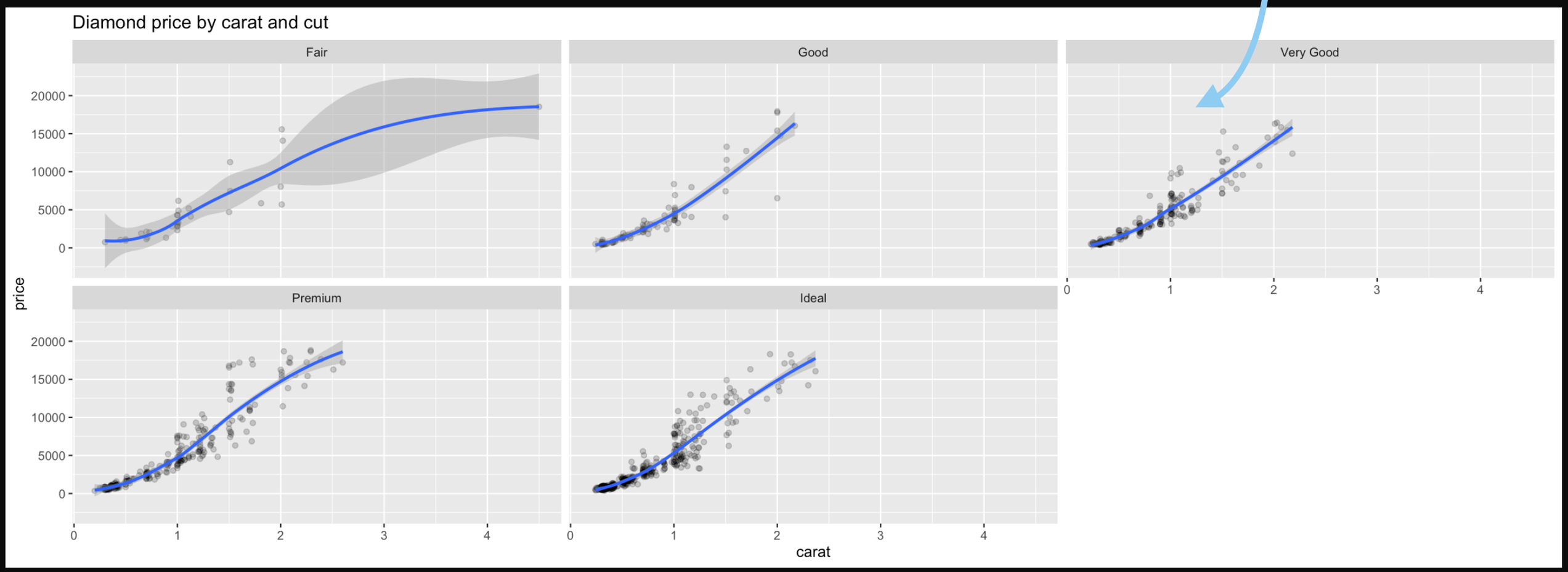
Plots don't reflect theme 😞

Theme demo Inputs Plots Tables Notifications Fonts Options

Choose an example

GeomSmooth

Plots rendered by R, not the browser!!

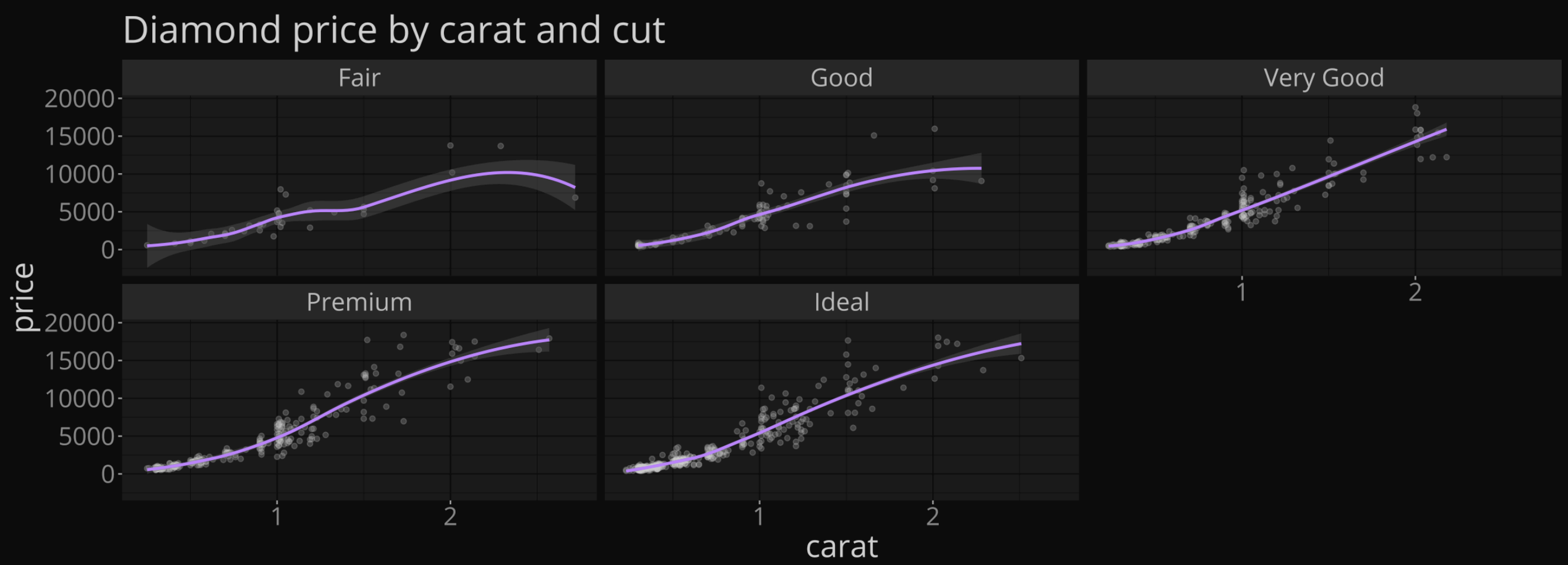


`thematic::thematic_shiny()` 🎉

Theme demo Inputs Plots Tables Notifications Fonts Options

Choose an example

GeomSmooth



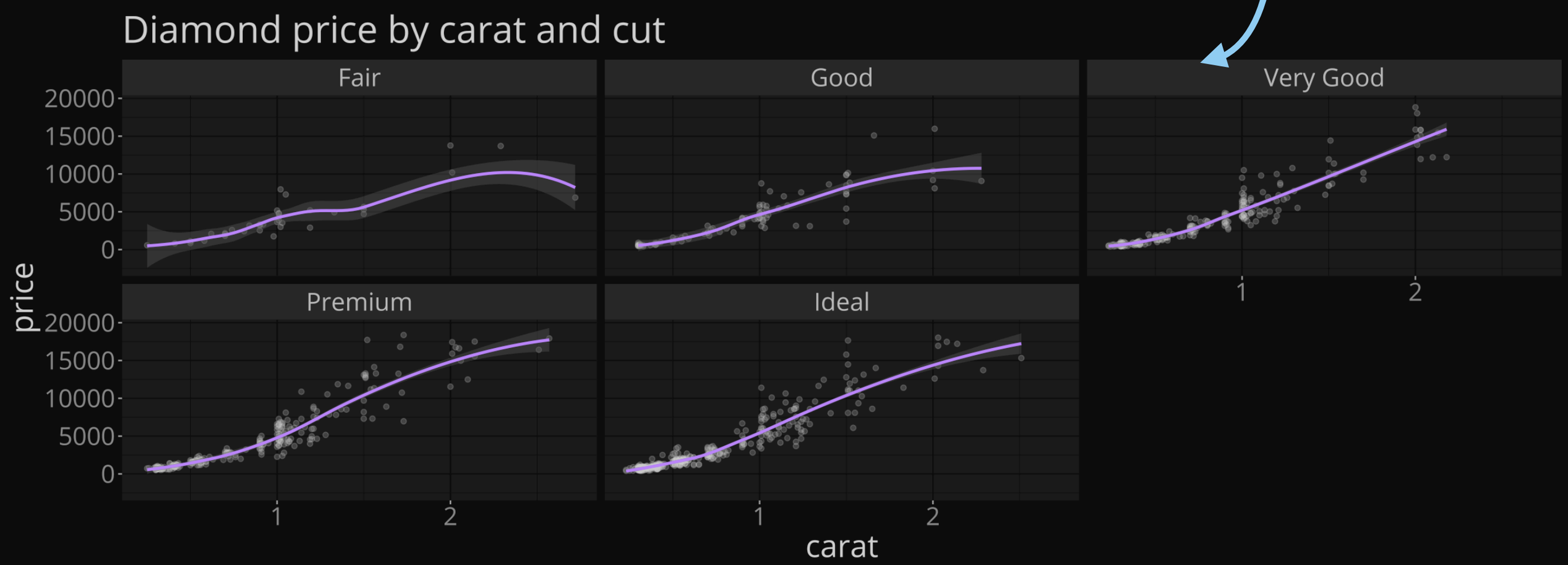
`thematic::thematic_shiny()` 🎉

Theme demo Inputs Plots Tables Notifications Fonts Options

Choose an example

GeomSmooth

Plot styling defaults now informed by CSS



Auto-theme any plotOutput()

```
library(shiny)
thematic::thematic_shiny()


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

shinyApp(ui, function(…) { })
```

Enables auto-coloring until app exits

Auto-theme any plotOutput()

```
library(shiny)
thematic::thematic_shiny(font='auto')

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

shinyApp(ui, function(…) { })
```

Translate fonts as well

Auto-theme any plotOutput()

```
library(shiny)
thematic::thematic_shiny(font='auto')

ui ← fluidPage(
  theme = bslib::bs_theme(),
  …
)

shinyApp(ui, function(…) { })
```

Works only if using Google Fonts or fonts known to R

The {thematic} R package

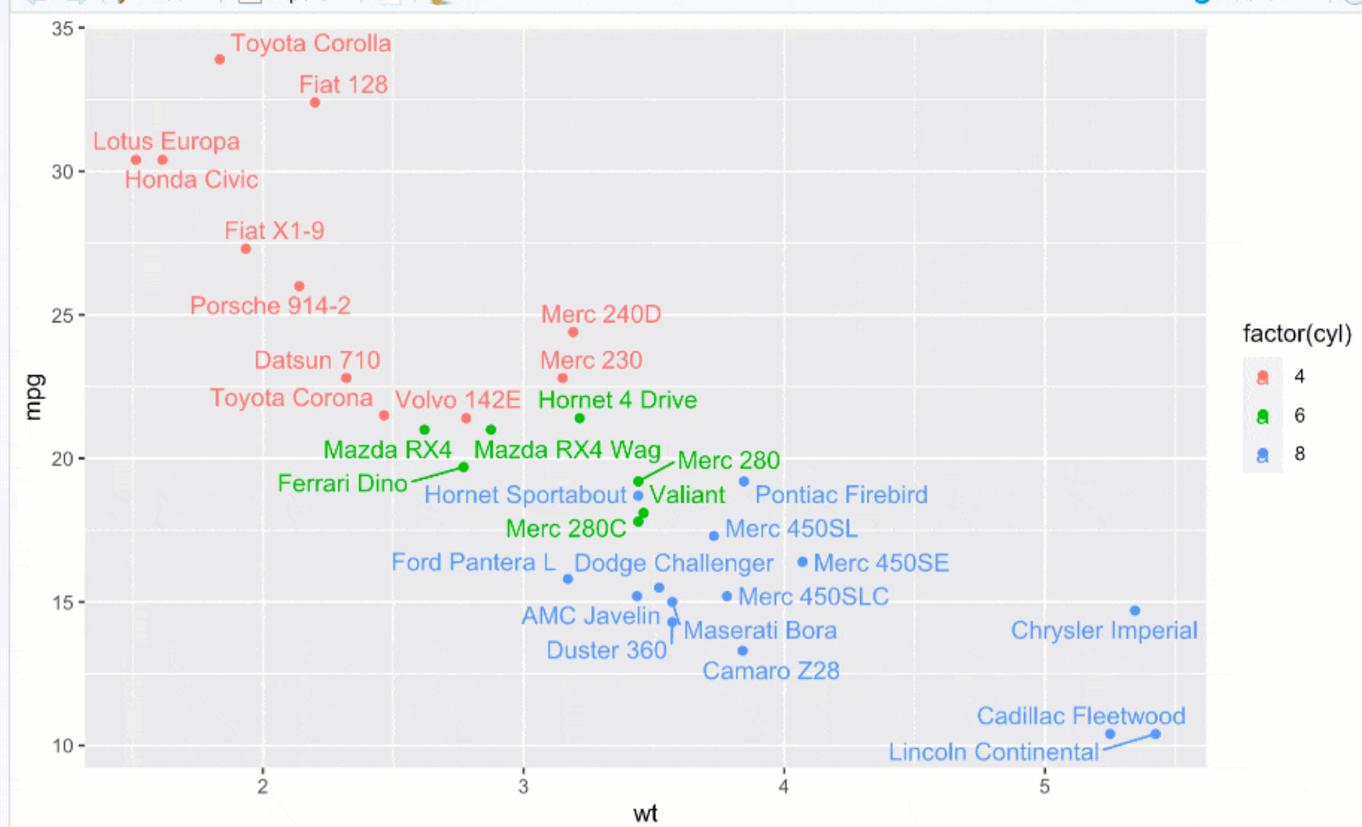
- Auto theme R plots in Shiny, R Markdown, and RStudio.
- Provides a simplified interface for theming `{ggplot2}`, `{lattice}`, and `{base}` graphics in any R runtime.

rstudio.github.io/thematic

Auto-theme plots in RStudio

thematic - master - RStudio

```r
thematic::thematic_on()

demo_plot <- function() {
  rsthemes::use_theme_favorite()

  ggplot(mtcars, aes(wt, mpg, label = rownames(mtcars), color = factor(cyl))) +
    geom_point() +
    ggrepel::geom_text_repel(max.overlaps = 25)
}

demo_plot()
```



Using {bslib} with R Markdown

```yaml
---
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
---
```

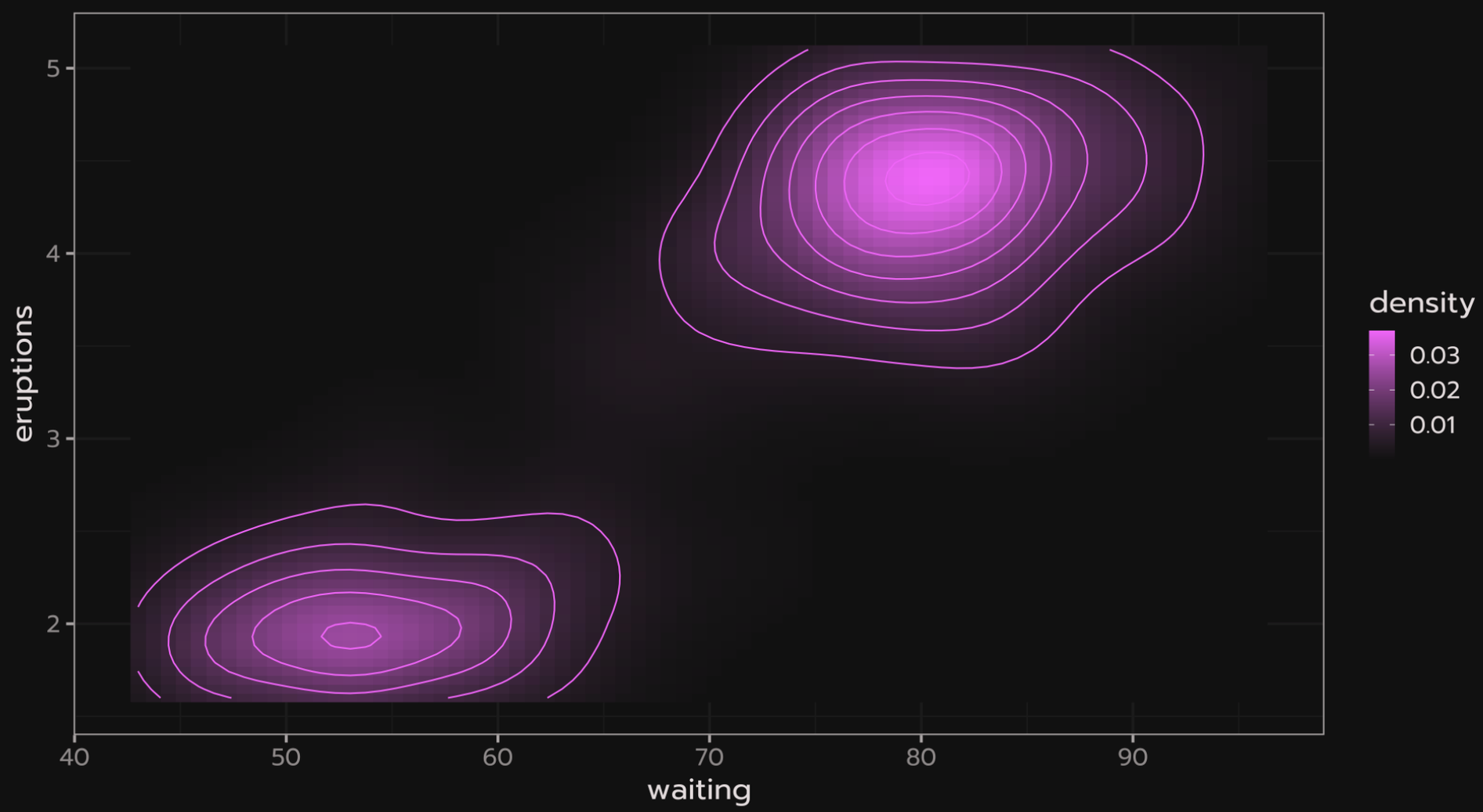
{flexdashboard}, {pkgdown}, and {bookdown} support coming to CRAN soon

flexdashboard-pkg.netlify.app/articles/theme.html

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

0 91

0 10 20 30 40 50 60 70 80 90 100

Choose a state

AL

Choose a date

2021-03-18 to 2021-03-18

1 primary

2 info

3 success

4 warning

5 dange

Success Rate

91 %

0 100

Warning metric

3.4

0 5

Dange

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

#ED79F9

Links and highlighted navigation

Secondary

#9E9B9B

Success

#28a745

Info

#17a2b8

Warning

#ffc107

Danger

#dc3545

Auto-theme plots in HTML

```
---
output:
  flexdashboard::flex_dashboard:
    theme:
      bg: "#121212"
      fg: "#E4E4E4"
      primary: “#BB86FC”
---

```{r}
thematic::thematic_rmd()
```
```

{thematic} can auto-theme with {bslib} in Rmd

Auto-theme plots in HTML

Flexdashboard theming demo Components Storyboard Cards About Home Source Code

Static Plots Interactive Plots Learn more

2d density estimate of Old Faithful data



Put `thematic_rmd()` for auto-plot theming!

Theme R plots anywhere

```
---
output:
  pdf_document
---

```{r}
thematic::thematic_rmd(
  bg = "#121212",
  fg = "#E4E4E4",
  accent = "#BB86FC"
)
```
```

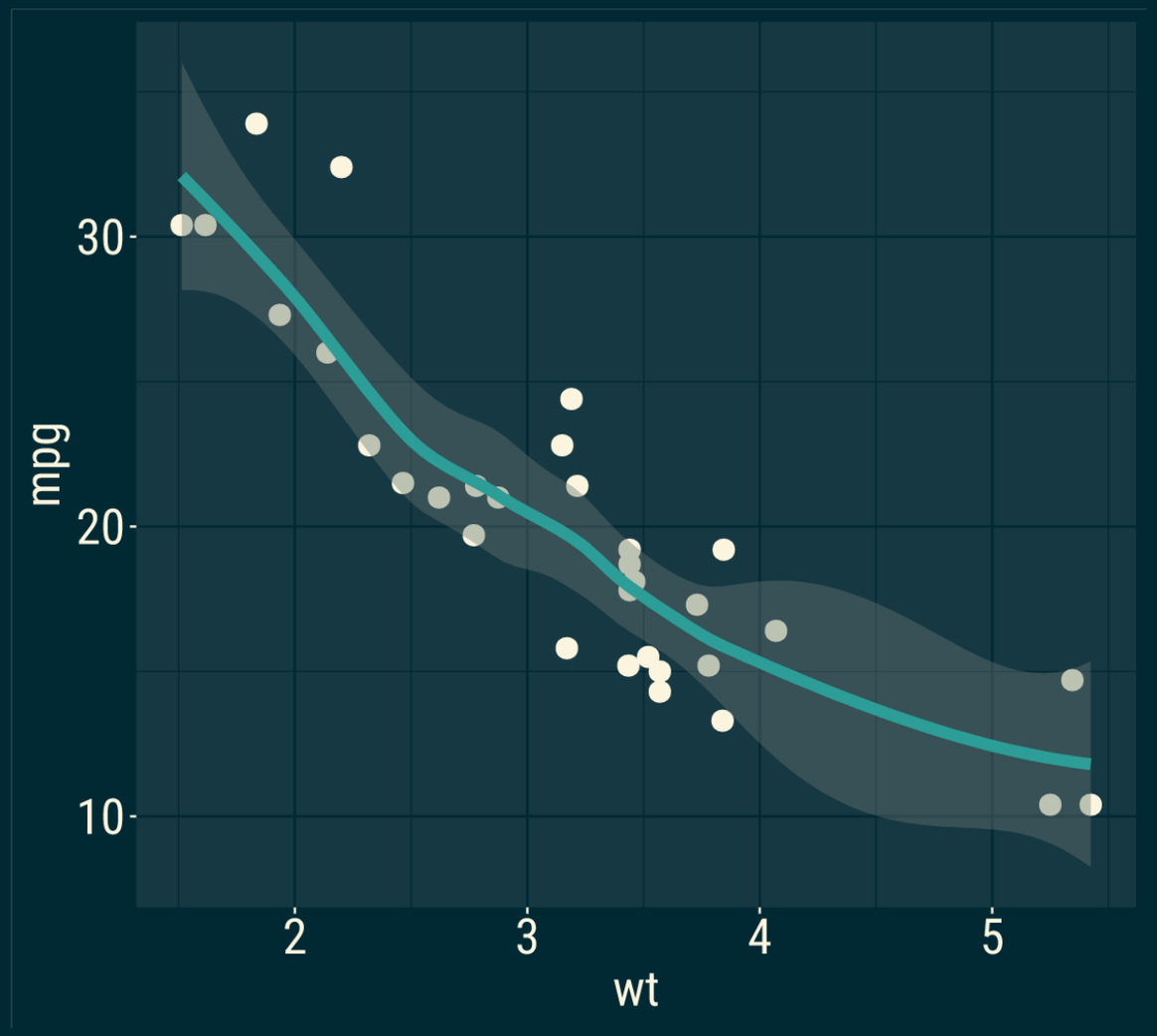
{thematic}'s {bslib}-like theme interface can be used anywhere

General {thematic} usage

- Three ways to enable globally: `thematic_shiny()`, `thematic_rmd()`, and `thematic_on()`
  - You can also enable thematic for one-time use
- Auto theming is the default behavior, but you can also directly specify colors and fonts.

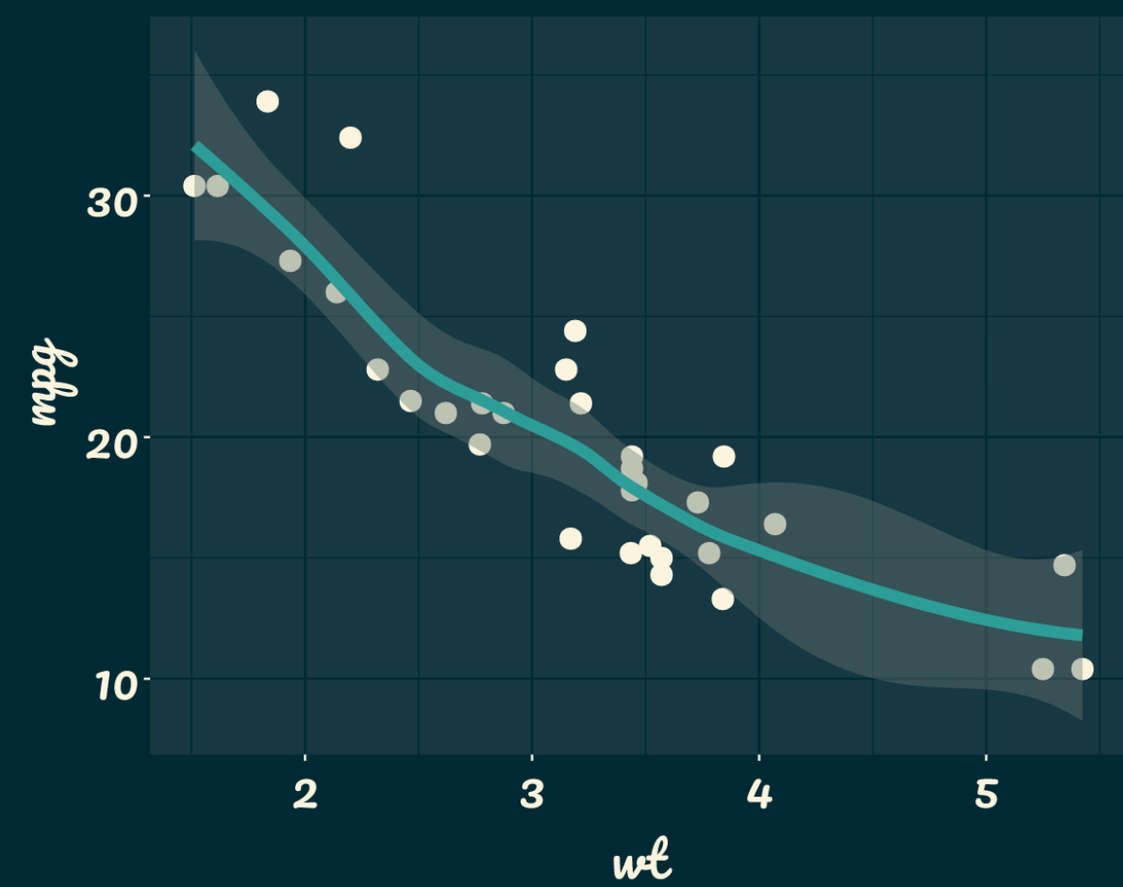
Provide colors directly

```
thematic_on(
  bg = "#002B36",
  fg = "#FDF6E3",
  accent = "#2AA198"
)
library(ggplot2)
ggplot(mtcars, aes(wt, mpg)) +
  geom_point() +
  geom_smooth()
```



Provide fonts directly

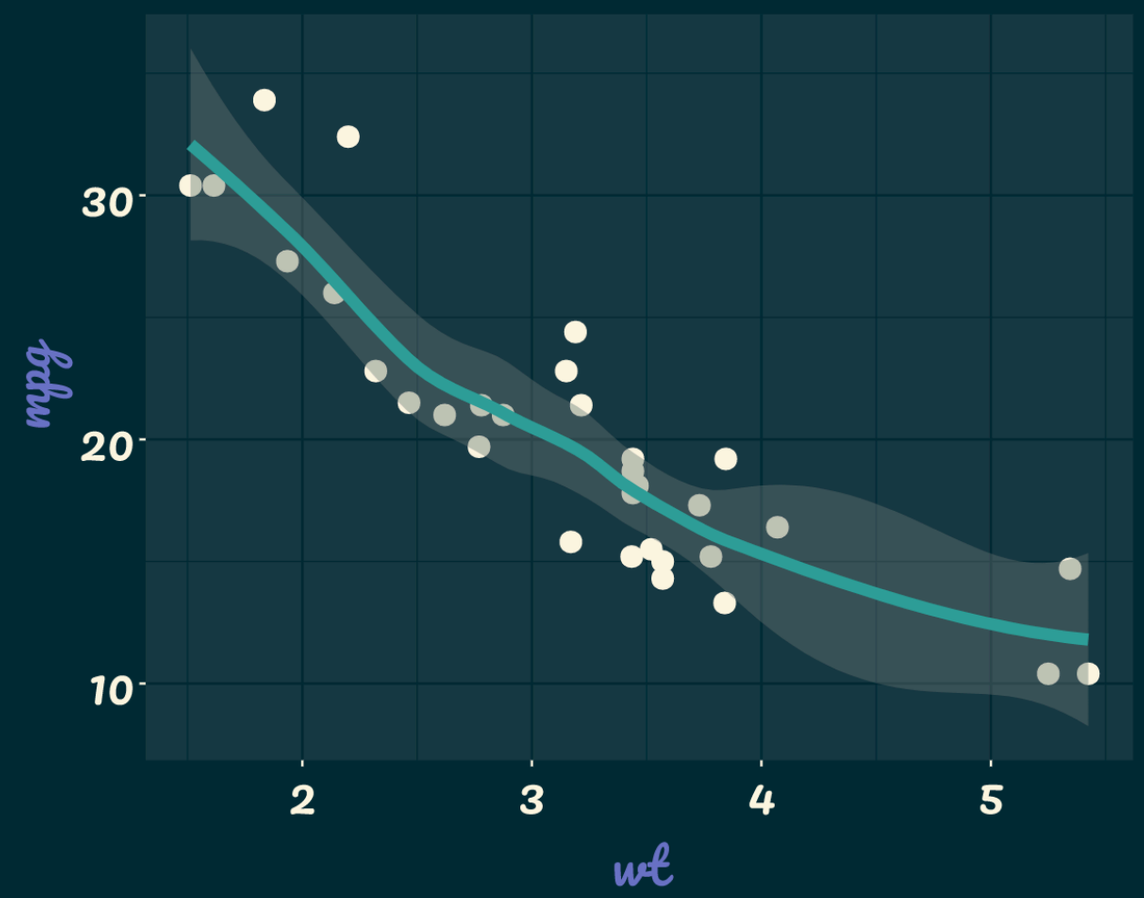
```
thematic_on(
  bg = "#002B36",
  fg = "#FDF6E3",
  accent = "#2AA198",
  font = "Pacifico"
)
library(ggplot2)
ggplot(mtcars, aes(wt, mpg)) +
  geom_point() +
  geom_smooth()
```



Any `font` known to R or any Google Font works so long as {showtext} or {ragg} is installed

{thematic} sets global theme() defaults

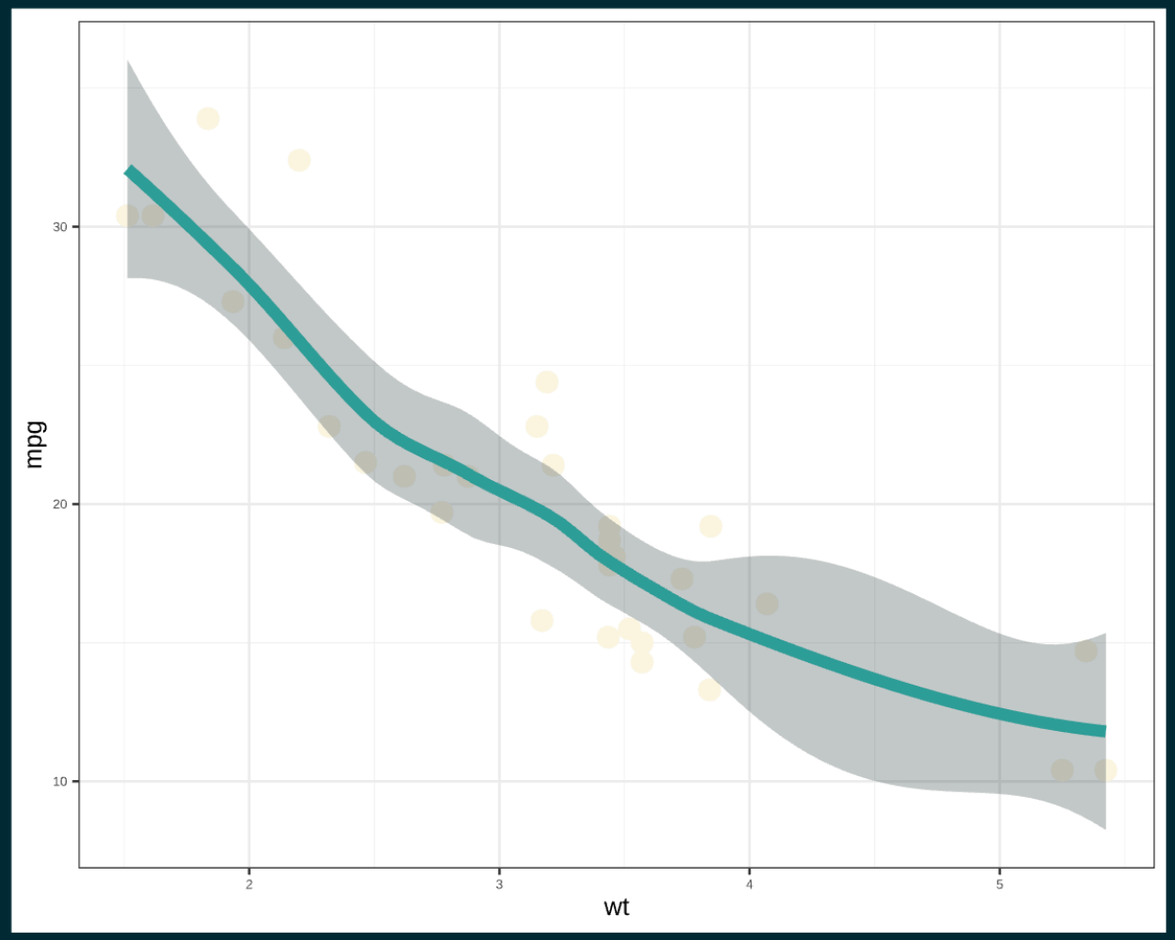
```
thematic_on(
  bg = "#002B36",
  fg = "#FDF6E3",
  accent = "#2AA198",
  font = "Pacifico"
)
library(ggplot2)
ggplot(mtcars, aes(wt, mpg)) +
  geom_point() +
  geom_smooth() +
  theme(axis.title =
element_text(color = "#6C71C4"))
```



Plot specific styles takes priority over global defaults

Don’t add complete theme to plot

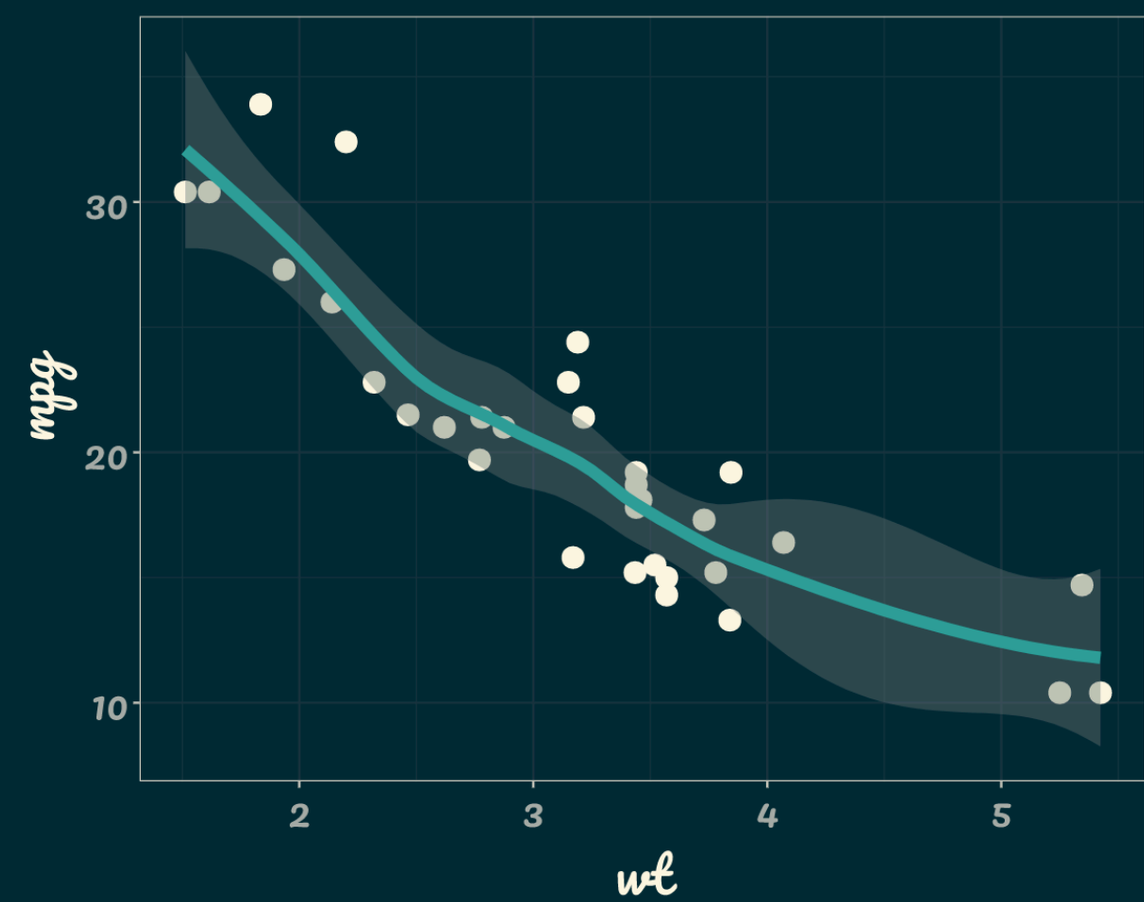
```
thematic_on(
  bg = "#002B36",
  fg = "#FDF6E3",
  accent = "#2AA198",
  font = "Pacifico"
)
library(ggplot2)
ggplot(mtcars, aes(wt, mpg)) +
  geom_point() +
  geom_smooth() +
  theme_bw()
```



`theme_bw()` ‘overrides’ all the global `theme()` defaults

Set complete themes globally

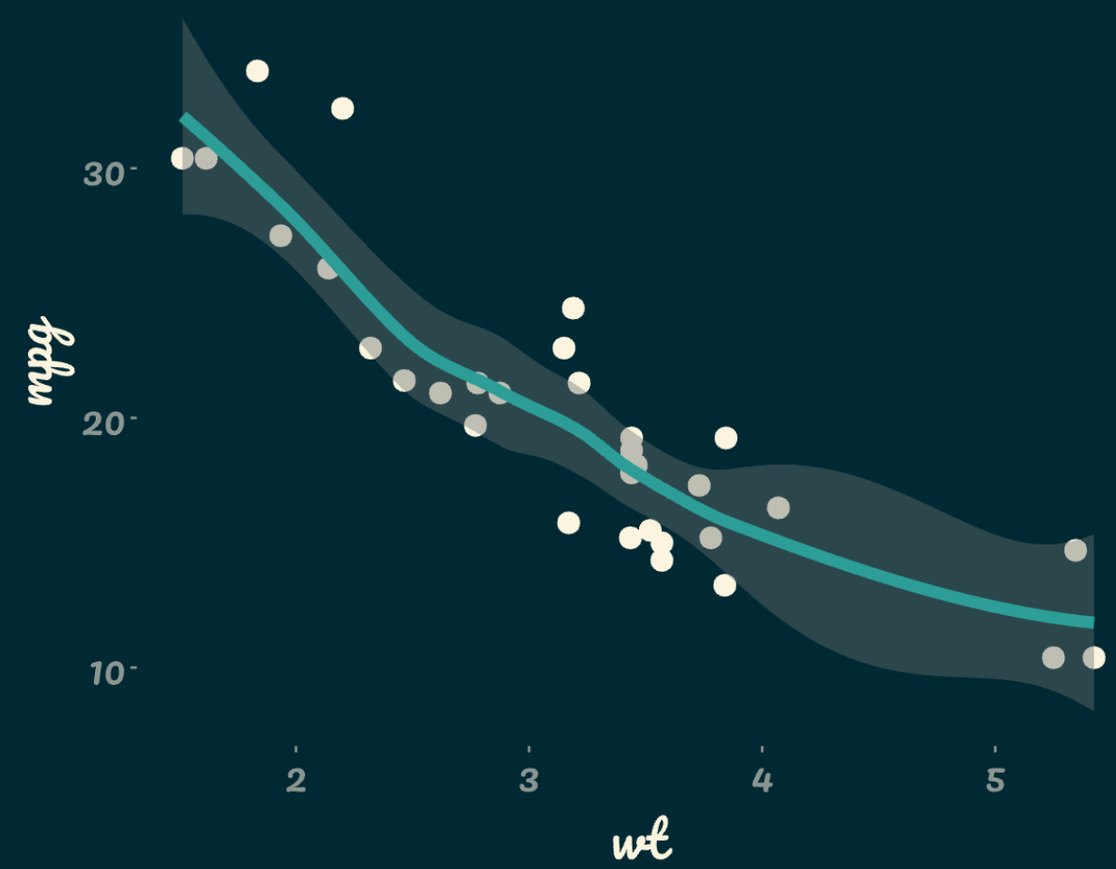
```r
theme_set(theme_bw())
thematic_on(
  bg = "#002B36",
  fg = "#FDF6E3",
  accent = "#2AA198",
  font = "Pacifico"
)
library(ggplot2)
ggplot(mtcars, aes(wt, mpg)) +
  geom_point() +
  geom_smooth()
```



`{thematic}` will even 'respect' the complete theme's semantics

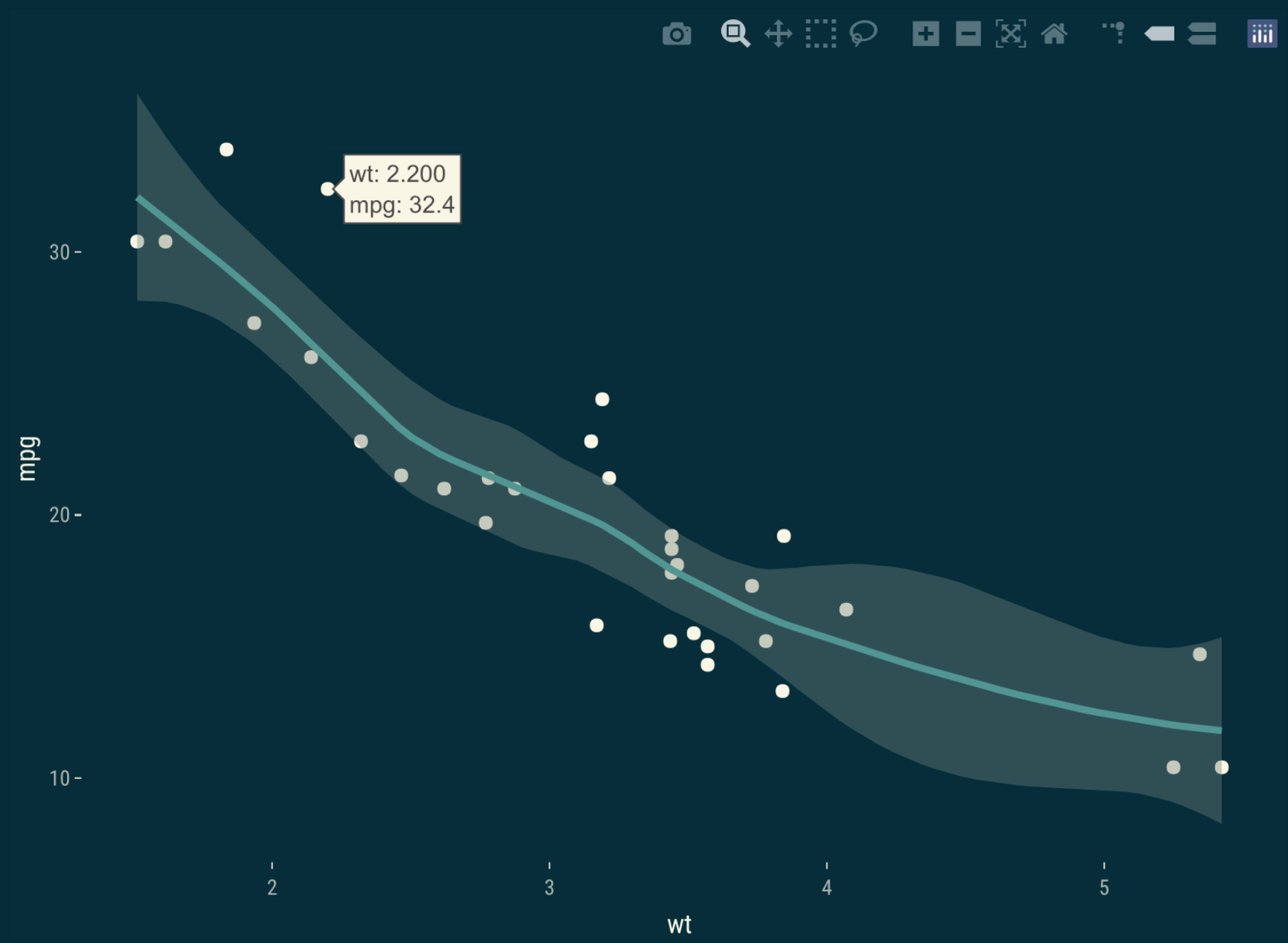
Set complete themes globally

```r
theme_set(ggthemes::theme_tufte())
thematic_on(
  bg = "#002B36",
  fg = "#FDF6E3",
  accent = "#2AA198",
  font = "Pacifico"
)
library(ggplot2)
ggplot(mtcars, aes(wt, mpg)) +
  geom_point() +
  geom_smooth()
```



`{thematic}` will even 'respect' the complete theme's semantics

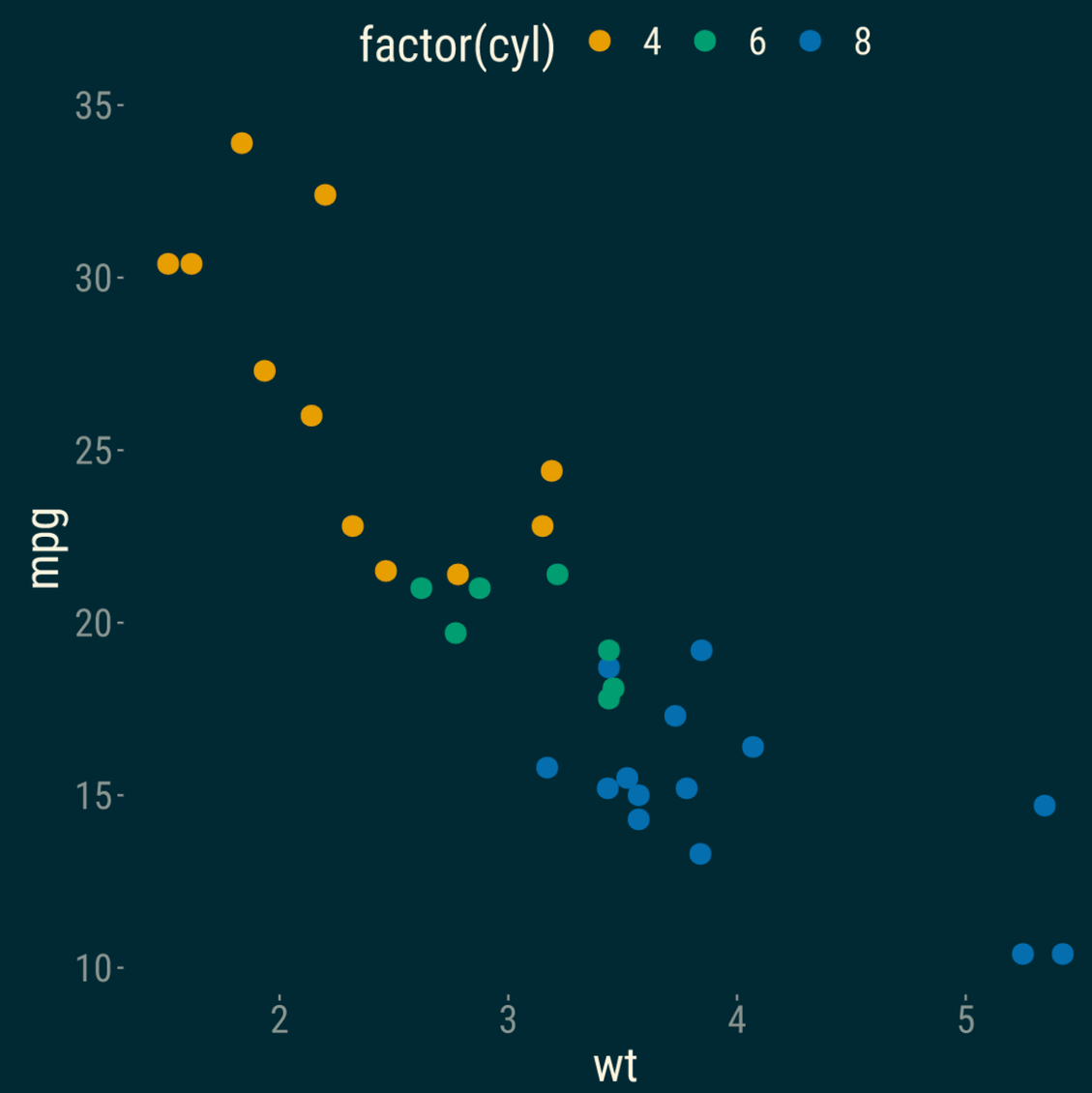
Also works with plotly::ggplotly()



{thematic} also sets global scales

Sets qualitative colorscale to (colour-blind safe) Okabe-Ito

```
thematic_on(
  bg = "#002B36",
  fg = "#FDF6E3",
  accent = "#2AA198",
  font = "Roboto Condensed"
)
library(ggplot2)
ggplot(mtcars, aes(wt, mpg)) +
  geom_point(aes(color=factor(cyl)))
```

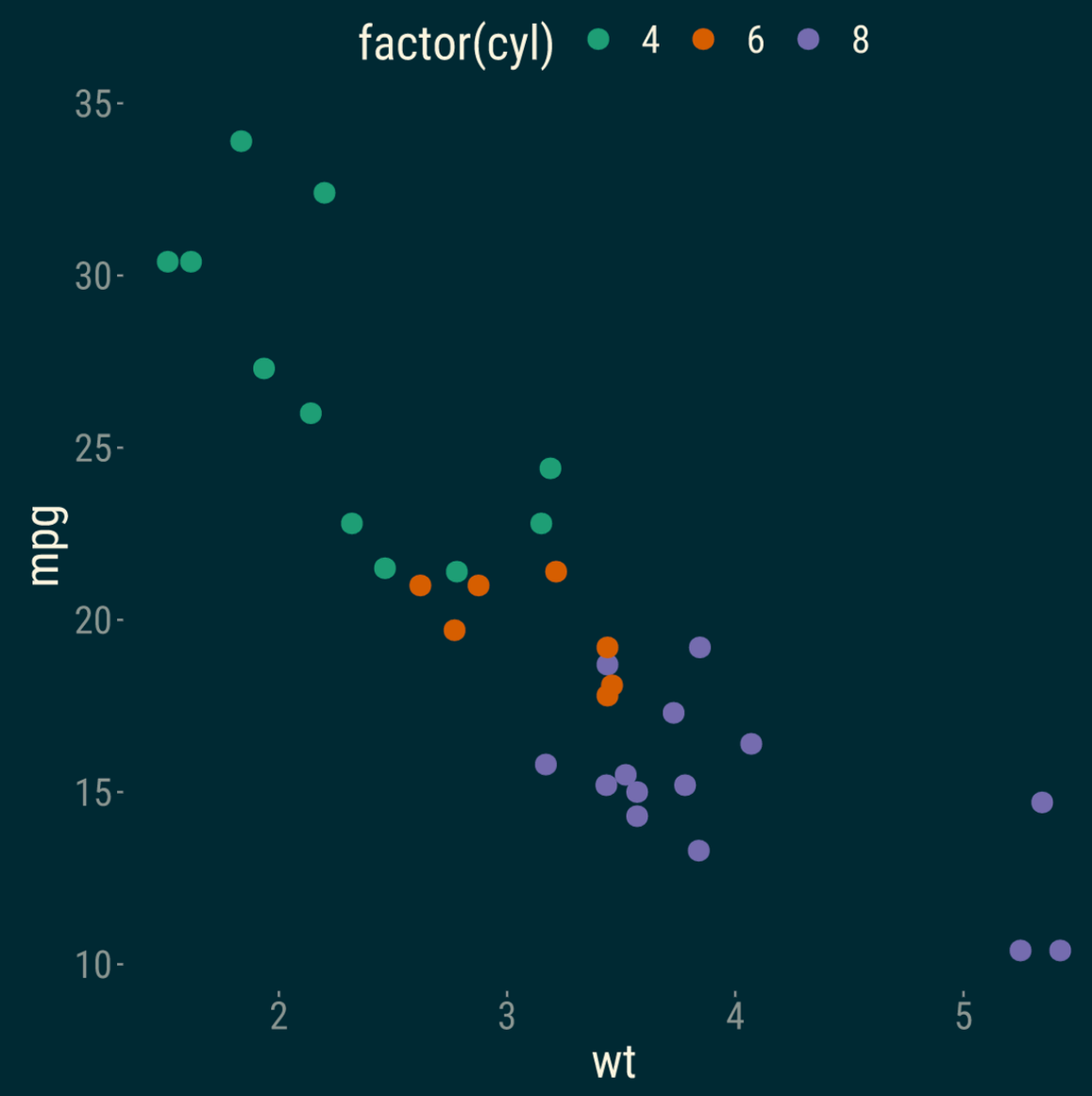


Scale defaults are also customizable

Customize with any vector of color codes

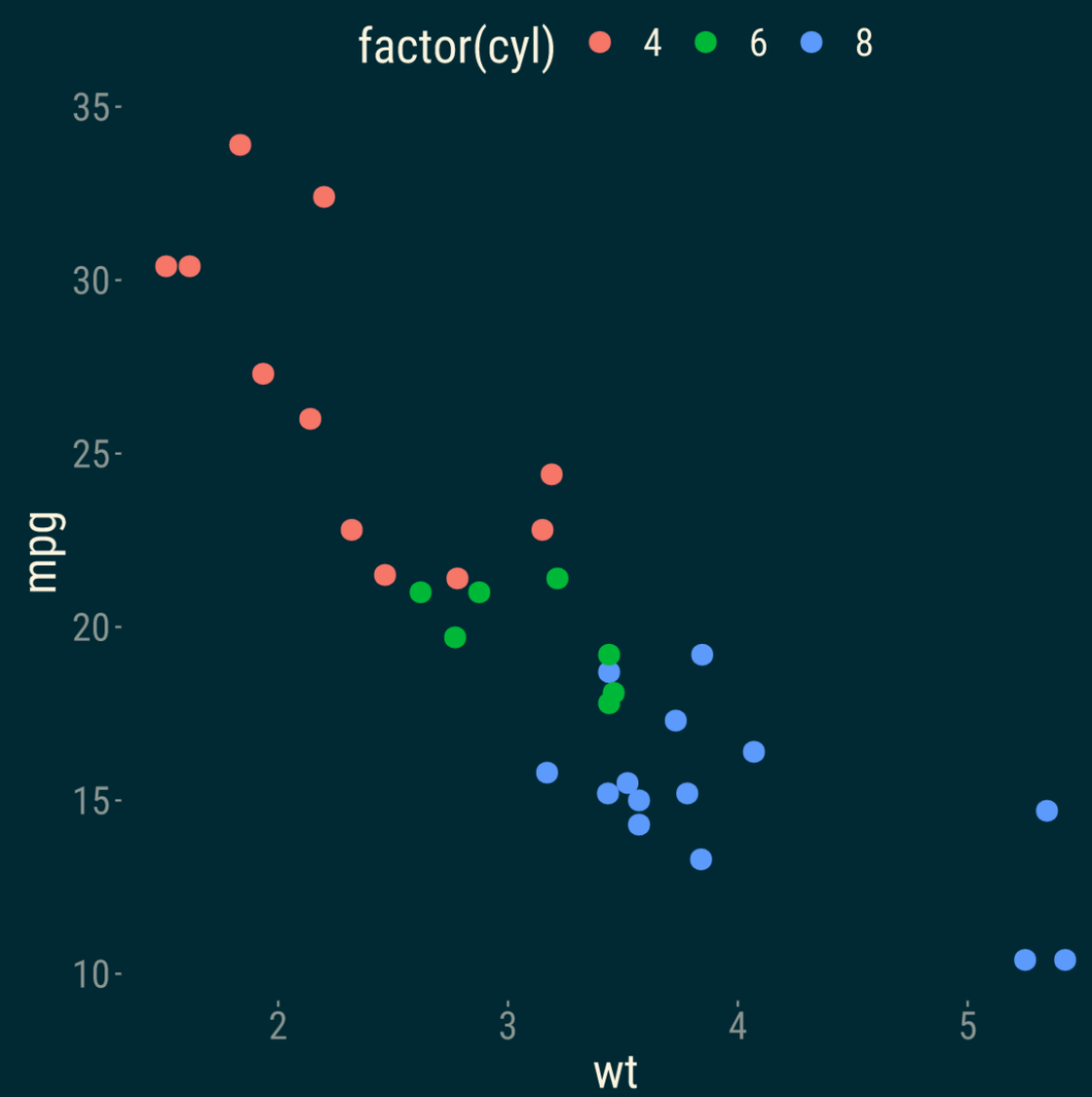
```
thematic_on(
  bg = "#002B36",
  fg = "#FDF6E3",
  accent = "#2AA198",
  font = "Roboto Condensed",
  qualitative = c(
    "#1B9E77", "#D95F02", "#7570B3"
  )
)
library(ggplot2)
ggplot(mtcars, aes(wt, mpg)) +
  geom_point(aes(color=factor(cyl)))
```

For **ggplot2**, `qualitative` is only relevant if there are enough colors to encode the levels.



Use {ggplot2} defaults (if you really need to)

```
thematic_on(
  bg = "#002B36",
  fg = "#FDF6E3",
  accent = "#2AA198",
  font = "Roboto Condensed",
  qualitative = NA
)
library(ggplot2)
ggplot(mtcars, aes(wt, mpg)) +
  geom_point(aes(color=factor(cyl)))
```



In summary

- Use `shiny::bindCache()` to avoid redundant computation
- Use `bslib::bs_theme()` to customize Bootstrap CSS
  - In R Markdown, provide arguments to `theme:`
- Use `bslib::bs_themer()` to customize in real-time
- Translate CSS to R plots with `thematic_shiny()`
- `{thematic}`'s auto-theming also works in RStudio (`thematic_on()`) and R Markdown (`thematic_rmd()`)
- Use and customize `{thematic}` themes anywhere by providing colors and fonts directly

Thank you!

Learn more:
rstudio.github.io/bslib
rstudio.github.io/thematic

Slides:
bit.ly/wb-shiny-2021

Contact:

@cpsievert

carson@rstudio.com

cpsievert.me