

A Package Stew

bit.ly/isu_package_stew

Ian Lyttle @ijlyttle

Analytics Applications and Programs

Life Is On



A Package Stew

Mixture of ingredients:

- `vembedr`: embed video into R Markdown/Shiny
- `bsplus`: work with Twitter Bootstrap for Rmd/Shiny
- `srlze`: help translate things in R to things in text
- `screenshotr`: a shiny gadget I'm starting to work on
- `packages`: what I've learned beyond Hadley's book

If you can find a unifying theme, please let me know :)

vembedr

Helps you embed video into HTML documents:

- YouTube
- Vimeo
- Microsoft Channel 9 (UseR! 2016 videos)

```
> install.packages("vembedr")
```

<https://ijlyttle.github.io/vembedr>

vembedr

Simplest way to use, grab the URL of the video

```
> embed_url( "https://www.youtube.com/watch?v=uV4UpCq2azs" )
```

```
> embed_url( "https://youtu.be/1-vcEr0PofQ?t=28s" )
```

Daily builds of RStudio IDE let you embed in viewer

- This appears to work only for YouTube and Vimeo

vembedr

Get the suggested R code to embed:

```
> suggest_embed( "https://www.youtube.com/watch?v=uV4UpCq2azs" )
```

```
> suggest_embed( "https://youtu.be/1-vcEr0PofQ?t=28s" )
```

vembedr

Fundamental functions:

```
> embed_youtube( "1-vcEr0PofQ" ) %>% use_start_time( "28s" )
```

`embed_youtube()`

`embed_vimeo()`

`embed_user2016()`

See also: tutor

RStudio tutor package (not yet CRAN)

<https://github.com/rstudio/tutor>

Use Markdown notation to embed video

```

```

bsplus

So, you want to fit more “stuff” into your Shiny app:

- More inputs, outputs
- More plots, pictures
- More help, documentation

bsplus

Package to wrap Twitter Bootstrap components:

- **Accordion sidebar** - more room for Shiny apps
- **Carousel** - more room for plots, images
- **Tooltip, popover, modal** - more documentation
- **Shiny help-links** - integrate extra documentation

bsplus

- Inspired by Eric Bailey's `shinyBS` package
- Implemented using the `htmltools` package
- Nothing depends on server part of `shiny`, all UI
- Can be used to make `rmarkdown` HTML documents
- Designed with `%>%` in mind
- Demo: https://ijlyttle.shinyapps.io/accordion_sidebar_navbar

Accordion sidebar

JOHN LENNON
PAUL MCCARTNEY
GEORGE HARRISON
Lead guitar, vocals
RINGO STARR

While My Guitar Gently Weeps

content_main

title_side

content_side

```
bs_accordion_sidebar(id = "beatles") %>%
  bs_append(
    title_side = "John Lennon",
    content_side = "Rhythm guitar, vocals",
    content_main = "Dear Prudence"
  ) %>%
  bs_append(
    title_side = "Paul McCartney",
    content_side = "Bass guitar, vocals",
    content_main = "Blackbird"
  ) %>%
  bs_append(
    title_side = "George Harrison",
    content_side = "Lead guitar, vocals",
    content_main = "While My Guitar Gently Weeps"
  ) %>%
  bs_append(
    title_side = "Ringo Starr",
    content_side = "Drums, vocals",
    content_main = "Don't Pass Me By"
  )
```

Accordion sidebar

JOHN LENNON

PAUL MCCARTNEY

GEORGE HARRISON

Lead guitar, vocals

RINGO STARR

While My Guitar Gently Weeps

- Create an accordion sidebar
- Append a panel set

```
bs_accordion_sidebar(id = "beatles") %>%  
  bs_append(  
    title_side = "John Lennon",  
    content_side = "Rhythm guitar, vocals",  
    content_main = "Dear Prudence"  
  ) %>%  
  bs_append(  
    title_side = "Paul McCartney",  
    content_side = "Bass guitar, vocals",  
    content_main = "Blackbird"  
  ) %>%  
  bs_append(  
    title_side = "George Harrison",  
    content_side = "Lead guitar, vocals",  
    content_main = "While My Guitar Gently Weeps"  
  ) %>%  
  bs_append(  
    title_side = "Ringo Starr",  
    content_side = "Drums, vocals",  
    content_main = "Don't Pass Me By"  
  )
```

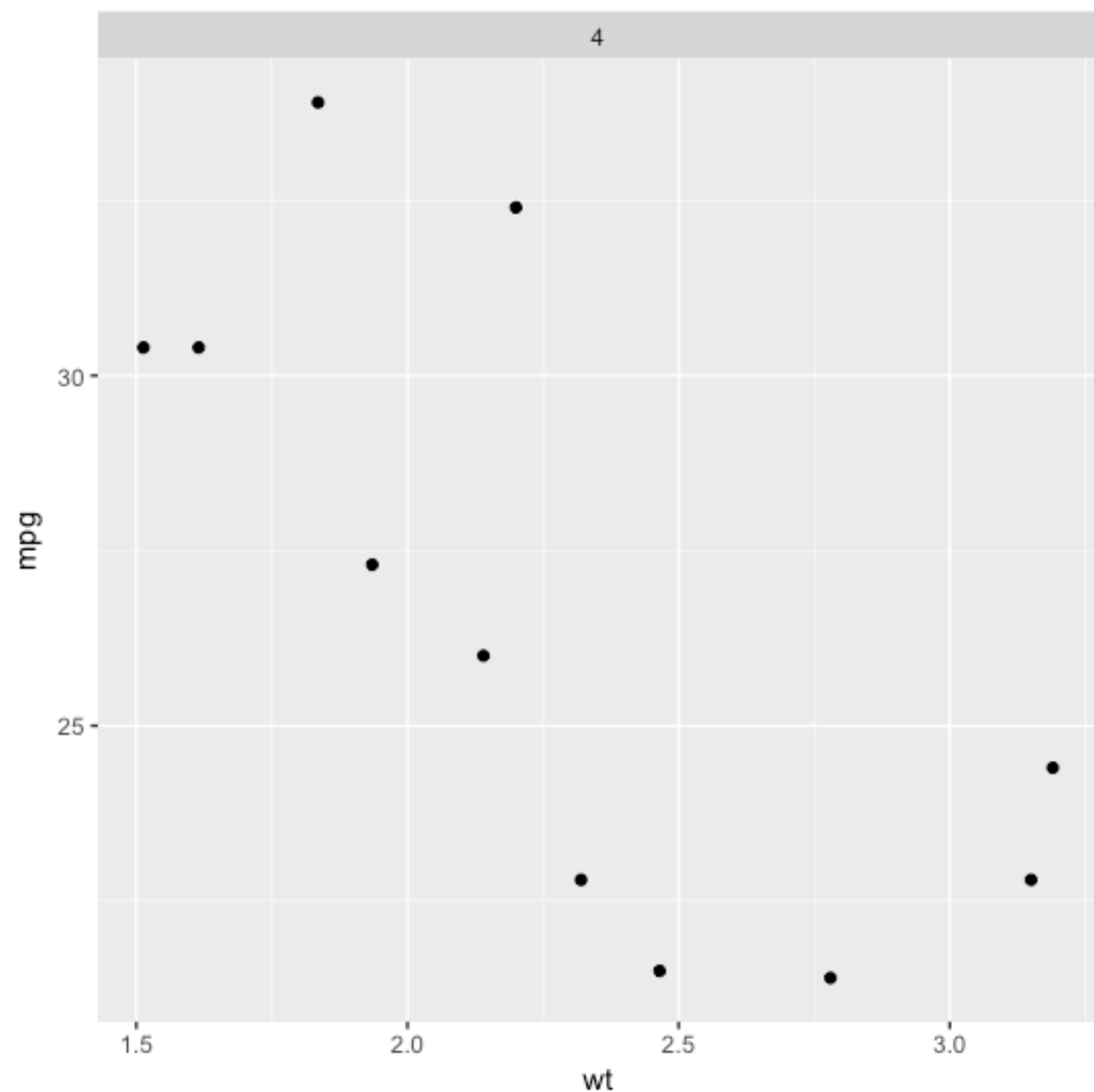
Carousel



```
bs_carousel(id = "with_the_beatles") %>%  
  bs_append(content = bs_carousel_image(src = "img/john.jpg")) %>%  
  bs_append(content = bs_carousel_image(src = "img/paul.jpg")) %>%  
  bs_append(content = bs_carousel_image(src = "img/george.jpg")) %>%  
  bs_append(content = bs_carousel_image(src = "img/ringo.jpg"))
```

- Create a carousel
- Append a slide

Carousel



```
bs_carousel(id = "with_the_beatles") %>%  
  bs_append(content = bs_carousel_image(src = "img/john.jpg")) %>%  
  bs_append(content = bs_carousel_image(src = "img/paul.jpg")) %>%  
  bs_append(content = bs_carousel_image(src = "img/george.jpg")) %>%  
  bs_append(content = bs_carousel_image(src = "img/ringo.jpg"))
```

I'd like to be able to:

- use **gganimate**
- make images using the **frame** aesthetic
- present the images in a **carousel**

without too much headache... (not today)

🎵 Baby, just use mtcars 🎵

Tooltip, popover

RHYTHM GUITAR, VOCALS

JOHN LENNON

```
library("htmltools")
```

```
tags$button(class = "btn btn-default", "John Lennon") %>%  
  bs_embed_tooltip(title = "Rhythm Guitar, vocals")
```

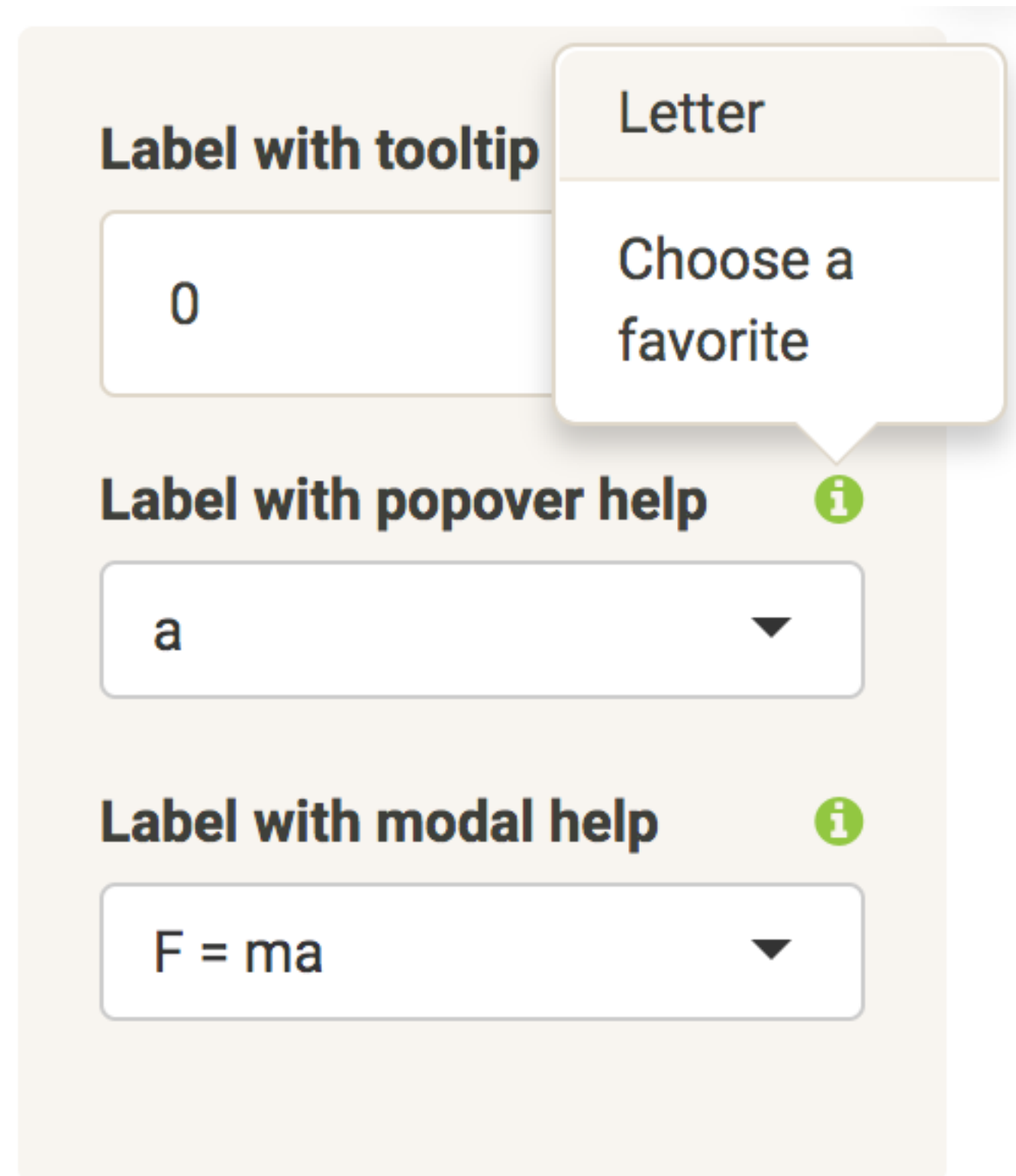
More information

Although he wrote "I Am the Walrus",
he later insisted that the Walrus was
Paul.

JOHN LENNON

```
tags$button(class = "btn btn-primary", "John Lennon") %>%  
  bs_embed_popover(  
    title = "More information",  
    content = "Although he wrote \"I Am the Walrus\",  
              he later insisted that the Walrus was Paul."  
  )
```

Shiny help-links



```
library("shiny")

selectInput(
  inputId = "letter",
  label = "Label with popover help",
  choices = c("a", "b", "c")
) %>%
shinyInput_label_embed(
  shiny_iconlink() %>%
  bs_embed_popover(title = "Letter", content = "Choose a favorite")
)
```

- Create an icon wrapped in a link
- Embed a popover into the link
- Embed the link into the label of a Shiny input

https://ijlyttle.shinyapps.io/accordion_sidebar_navbar

verbs

- **append**
You can *append* more than one of a thing to a tag, i.e. more than one slide to a carousel.
- **attach**
You can *attach* the id of a thing to more than one tag, but a tag can have only one id attached to it, i.e. you attach an id of a modal-window to a button-tag.
- **embed**
You can *embed* a thing into a tag, i.e. a tooltip into a link.

More API philosophy

- I tried to follow, as closely as I could, to the Twitter Bootstrap examples:
- <http://getbootstrap.com/javascript/>
- Make as few explicit arguments as practicable, make other options available through helper function:

```
bs_set_data( )
```

bs_set_data()?

- Consider the options for tooltips:
<http://getbootstrap.com/javascript/#tooltips-options>
- Set the delay by setting the attribute `data-delay` to the number of milliseconds
- Using `bs_set_data()`, use the name `delay` and express the value in an R-sensible way.

```
tags$button(class = "btn btn-default", "John Lennon") %>%  
  bs_embed_tooltip(title = "Rhythm Guitar, vocals") %>%  
  bs_set_data(delay = dseconds(1))
```

some links

- **pkgdown** 🙌🎉🚀 documentation:
- <https://ijlyttle.github.io/bsplus>
- **shiny** demonstration:
- https://ijlyttle.shinyapps.io/tooltip_popover_modal
- https://ijlyttle.shinyapps.io/accordion_sidebar_navbar

Other Projects

srize

- An API should make it easy to translate from R-native objects to whatever is on the other side
- Example, web-services tend to like time in milliseconds, logicals as “true”, “false”

screenshotr

- a lot of the screenshotting process can be automated
- potential application for a shiny gadget

Package Advice

- I don't have a "whole lot" of experience here
- Couple packages on CRAN
- Hadley's R-packages book is indispensable
- These are a few additional observations

pkgdown

From Hadley (of course): <http://hadley.github.io/pkgdown/>

- Makes a good-looking website from
 - README
 - vignettes
 - function documentation (man pages)
- Helps you
 - confront your documentation,
 - think about how the functions “fit” together
 - discover inconsistencies in your API
- Not on CRAN yet, and there are still some “fiddly” bits

Argue with your API!

- A consistent API makes things easier to
 - program
 - document
 - use
- Sketching it out beforehand can save a lot of grief
 - <https://github.com/hadley/http/issues/351>
 - <https://github.com/ijlyttle/bsplus/issues/22>

Use r-hub

- Cloud service to check your R-package
- Developed by Gábor Csárdi as an R-Consortium project
- **rhub** is on CRAN
- Register for a free account, then launch from command-line:

```
> rhub::check_for_cran( )
```