Building R-Shiny Applications

FAS6932 – Special Topics

Summer C 2025

Day 2 – UI Design

- Morning Session
  - UI Design principles
  - Guided build session
- Lunch
- Afternoon Session
  - App Design
  - Open design session
  - Small group breakout

Icebreaker

- What’s your design goal?
  - A particular visualization (maps, networks, phylogenies)?
  - A particular interactivity (simulations, interactive plots)?
  - A particular product (data entry, user tracking, data visualization)?

UI Design principles

- Simplicity
- Consistency
- Visual Hierarchy
- Feedback and Responsiveness
- Accessibility
- Clarity
- User Control
- Error Prevention and Recovery
- Aesthetics and Visual Appeal
- Scalability and Adaptability

UI Design principles

- Simplicity
- Consistency
- Visual Hierarchy
- Feedback and Responsiveness
- Accessibility
- Clarity
- User Control
- Error Prevention and Recovery
- Aesthetics and Visual Appeal
- Scalability and Adaptability

- UI Design choices
- Really for family of apps or b/w pages
- Layout choice
- Alerts & Reactivity
- Walkthrough/Tooltips
- Layout design choices
- Widget choice
- App design
- Aesthetic choices
- Layout design choices

UI Design principles

- Simplicity
- Consistency
- Visual Hierarchy
- Feedback and Responsiveness
- Accessibility
- Clarity
- User Control
- Error Prevention and Recovery
- Aesthetics and Visual Appeal
- Scalability and Adaptability

- UI Design choices
- ~~Really for family of apps or b/w pages~~
- Layout choice
- ~~Alerts &~~ Reactivity >> tomorrow!
- ~~Walkthrough/Tooltips~~
- Layout design choices
- Widget choices
- App design >> this afternoon!
- Aesthetic choices
- Layout design choices

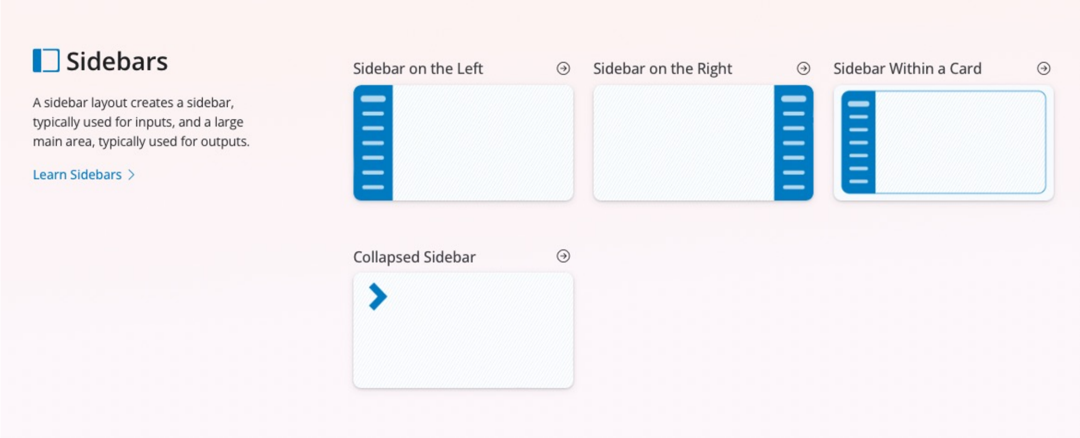
UI Design principles

- Simplicity
- Consistency
- **Visual Hierarchy**
- Feedback and Responsiveness
- Accessibility
- Clarity
- User Control
- Error Prevention and Recovery
- Aesthetics and Visual Appeal
- **Scalability and Adaptability**

- UI Design choices
- Really for family of apps or b/w pages
- **Layout** choice
- Alerts & Reactivity
- Walkthrough/Tooltips
- **Layout design** choices
- Widget choices
- App design
- Aesthetic choices
- **Layout design** choices

Layout Choices

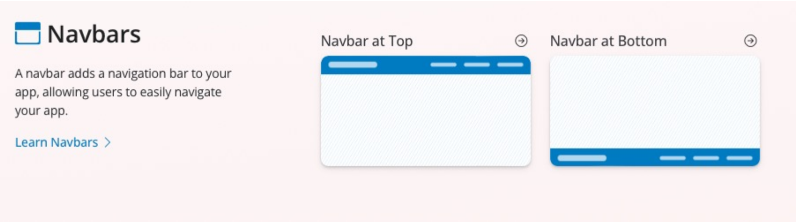
https://shiny.posit.co/r/layouts/



Sidebars

A sidebar layout creates a sidebar, typically used for inputs, and a large main area, typically used for outputs.

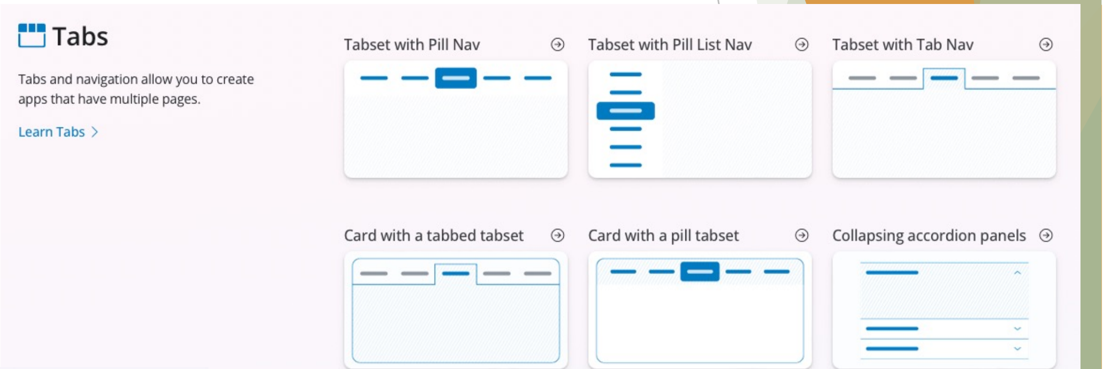
Learn Sidebars >



Navbars

A navbar adds a navigation bar to your app, allowing users to easily navigate your app.

Learn Navbars >

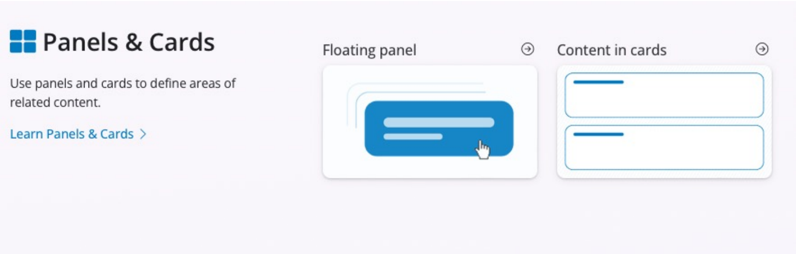


Tabs

Tabs and navigation allow you to create apps that have multiple pages.

Learn Tabs >

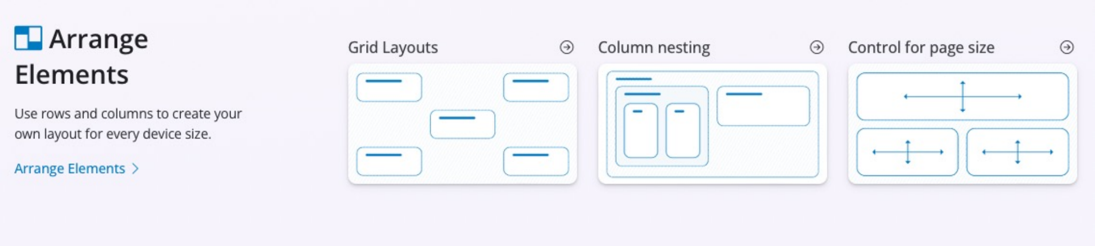
Tabset with Pill Nav
Tabset with Pill List Nav
Tabset with Tab Nav
Card with a tabbed tabset
Card with a pill tabset
Collapsing accordion panels



Panels & Cards

Use panels and cards to define areas of related content.

Learn Panels & Cards >



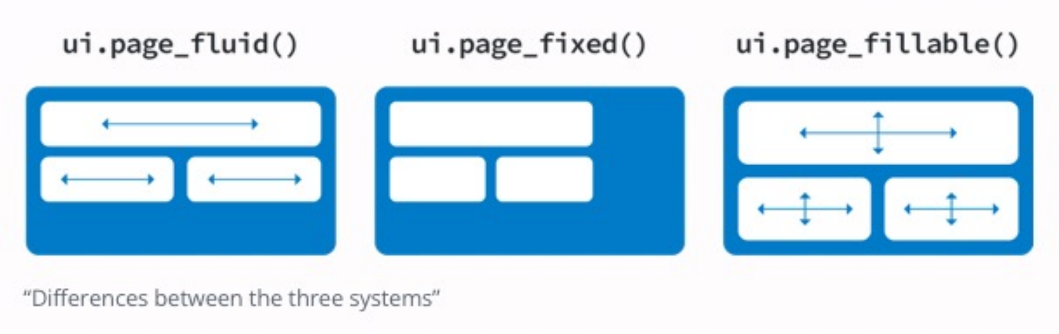
Arrange Elements

Use rows and columns to create your own layout for every device size.

Arrange Elements >

Really there are 3 page types to start

- `page_fillable` : extends page to full page width and height
- `page_fluid` : extends page to full page width
- `page_fixed` : extends page to centered horizontally and width is fixed
  - bad for scalability!!!



"Differences between the three systems"

Really there are 3 page types to start

- page_fillable : extends page to full page width and height
- page_fluid : extends page to full page width
- page_fixed : extends page to centered horizontally and width is fixed
  - bad for scalability!!!

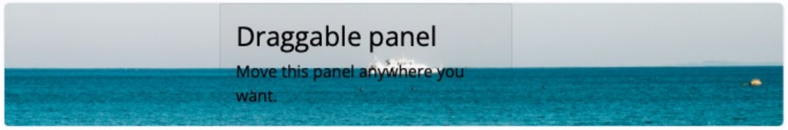
2 additional combination page types

- page_navbar : multiple page_fillable (fillable = TRUE) or page_fixed
- page_sidebar : split page_fillable (fillable = TRUE) or page_fixed

Establishing visual hierarchy

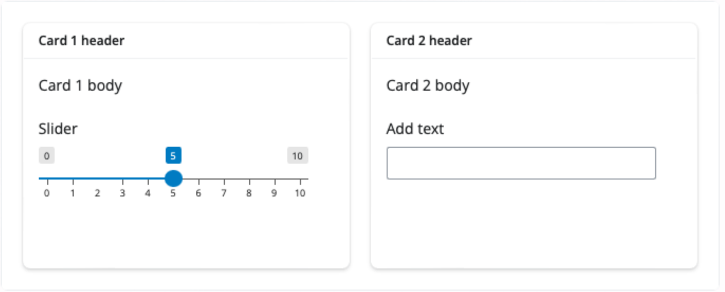
https://shiny.posit.co/r/layouts/panels-cards/

- Panels “float” on top of existing page or container layout
  - absolutePanel : establishes a panel whose position is relative to the parent container or page borders
  - fixedPanel : establishes a panel whose position is fixed. This means it will not move when you scroll on the page
  - Panels can be draggable (draggable = TRUE)
  - Have to specify:
    - exactly two of top, bottom, height
    - exactly two of left, right, width
- Cards are a useful way to subdivide content blocks



Draggable panel

Move this panel anywhere you want.



Card 1 header

Card 1 body

Slider

Card 2 header

Card 2 body

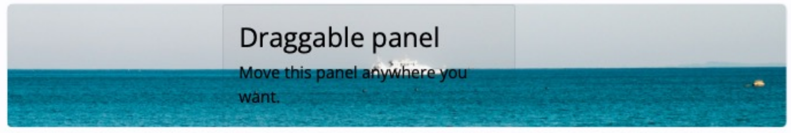
Add text

Establishing visual hierarchy

https://shiny.posit.co/r/layouts/panels-cards/

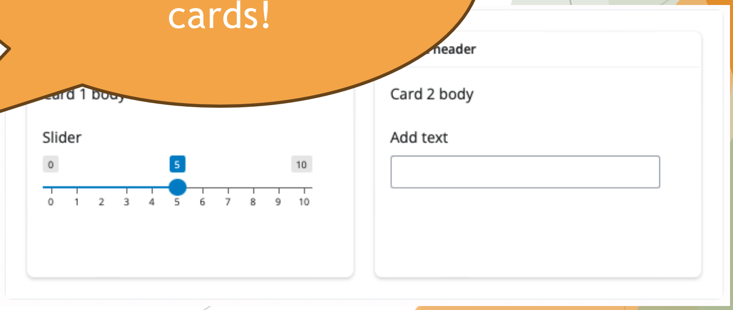
- Panels “float” on top of existing page or container layout
  - absolutePanel : establishes a panel whose position is relative to the parent container or page borders
  - fixedPanel : establishes a panel whose position is fixed. This means it will not move when you scroll on the page
  - Panels can be draggable (draggable = TRUE)
  - Have to specify:
    - exactly two of top, bottom, height
    - exactly two of left, right, width
- Cards are a useful way to subdivide content blocks

If you are a novice to HTML, go with cards!



Draggable panel

Move this panel anywhere you want.



header

ard 1 body

Slider

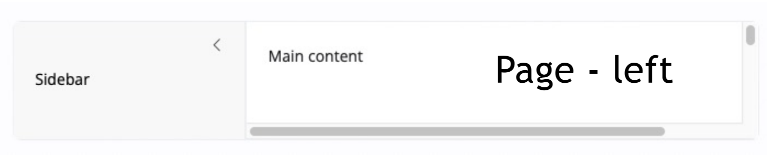
Card 2 body

Add text

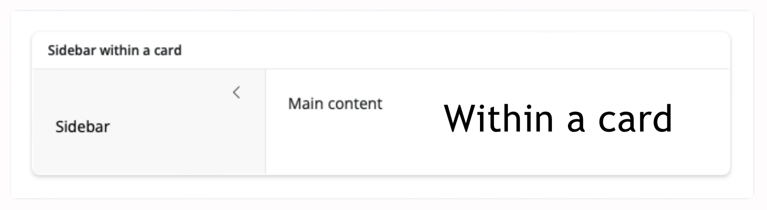
Other layouts are nested within page choice

https://shiny.posit.co/r/layouts/sidebars/

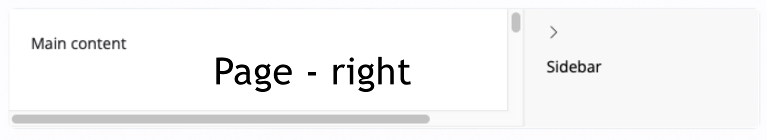
- Sidebar are either on left, right, or ~~collapsed~~, or within a card
  - `page_sidebar` or `sidebar` or `layout_sidebar`



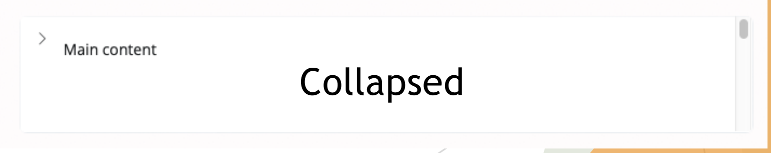
Page - left



Within a card



Page - right

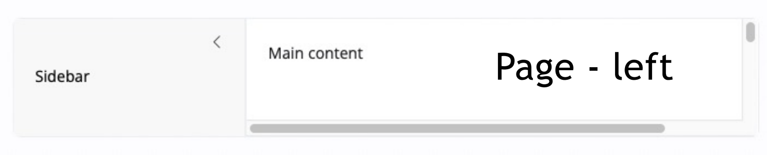


Collapsed

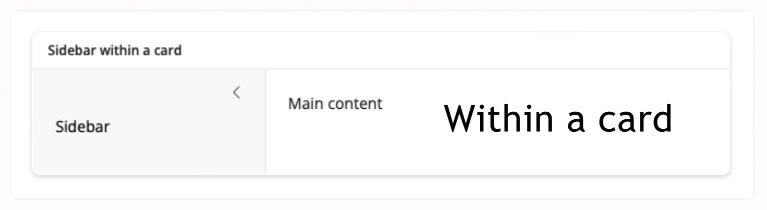
Other layouts are nested within page choice

https://shiny.posit.co/r/layouts/sidebars/

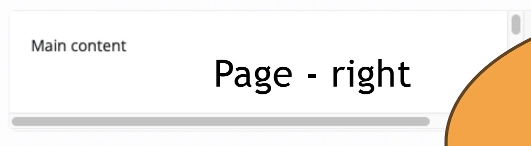
- Sidebar are either on left, right, or ~~collapsed~~, or within a card



Page - left

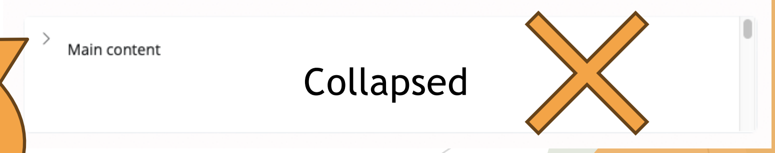


Within a card



Page - right

No person of reasonable age can ever find this collapse >

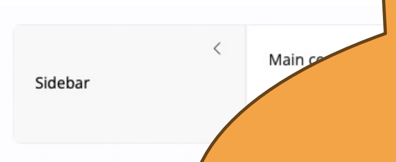


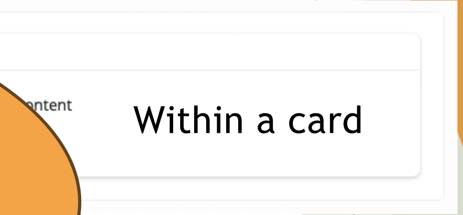
Collapsed

Other layouts are nested within page choice

https://shiny.posit.co/r/layouts/sidebars/

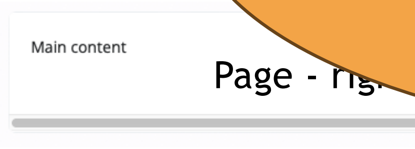
- Sidebar are either on left, right, or ~~collapsed~~, or within a card
  - `page_sidebar` or `sidebar` or `layout_sidebar`



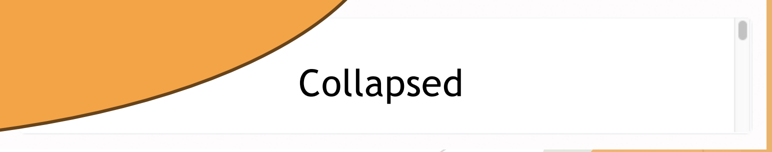


Within a card

This is by far the **fastest** way to setup a simple application. Users are generally familiar with sidebar navigation which helps quickly orient users to a new app without heavy instruction.



Page - rig

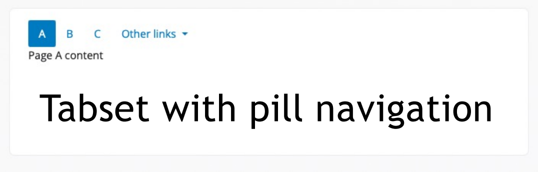


Collapsed

Other layouts are nested within page choice

https://shiny.posit.co/r/layouts/tabs/

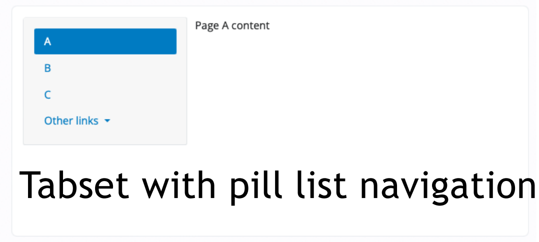
- **Tabs** are equivalent to page_navbar but with more flexibility to move in the page or in cards (page_navbar is either top or bottom)



A B C Other links ▾

Page A content

Tabset with pill navigation



A

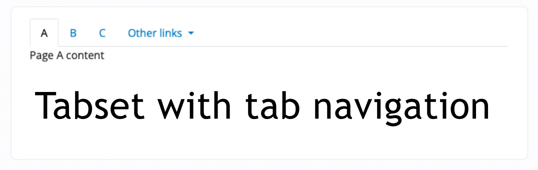
B

C

Other links ▾

Page A content

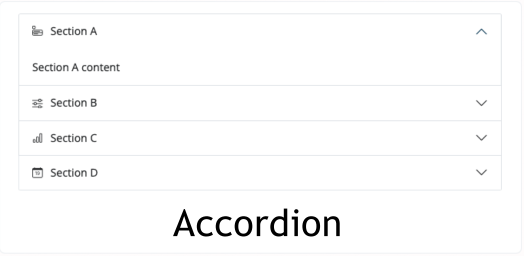
Tabset with pill list navigation



A B C Other links ▾

Page A content

Tabset with tab navigation



Section A

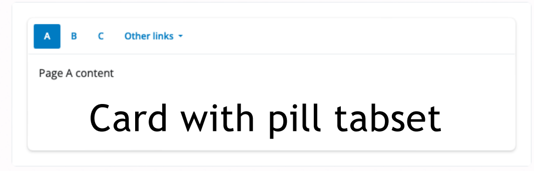
Section A content

Section B

Section C

Section D

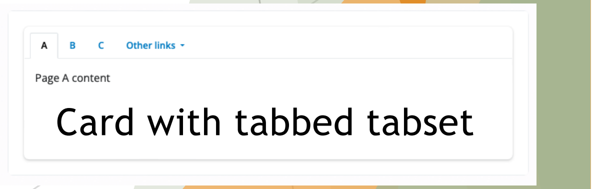
Accordion



A B C Other links ▾

Page A content

Card with pill tabset



A B C Other links ▾

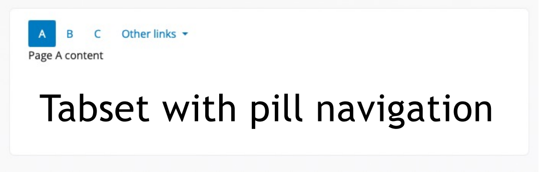
Page A content

Card with tabbed tabset

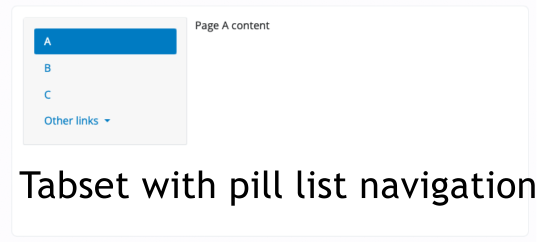
Other layouts are nested within page choice

https://shiny.posit.co/r/layouts/tabs/

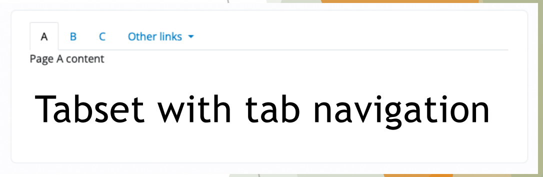
- **Tabs** are equivalent to page_navbar but with more flexibility to move in the page or in cards (page_navbar is either top or bottom)



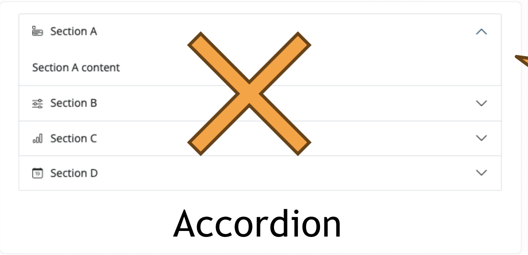
Tabset with pill navigation



Tabset with pill list navigation

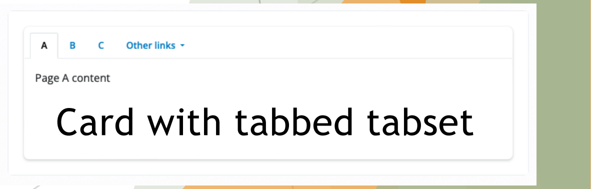


Tabset with tab navigation



Accordion

Most people of reasonable age cannot find this collapse ^

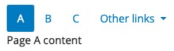


Card with tabbed tabset

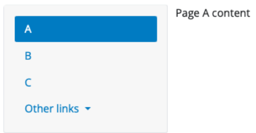
Other layouts are nested within page choice

https://shiny.posit.co/r/layouts/tabs/

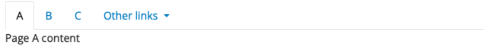
- **Tabs** are equivalent to page_navbar but with more flexibility to move in the page or in cards (page_navbar is either top or bottom)



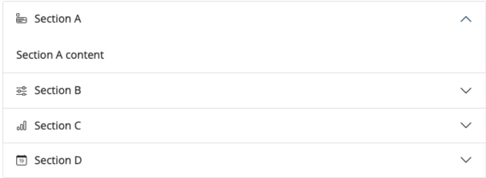
Tabset with pill navigation



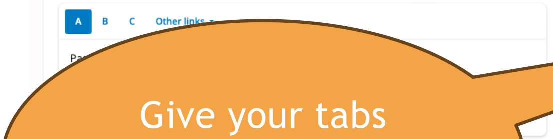
Tabset with pill list navigation



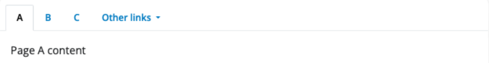
Tabset with tab navigation



Accordion



Give your tabs decent names that are intuitive!

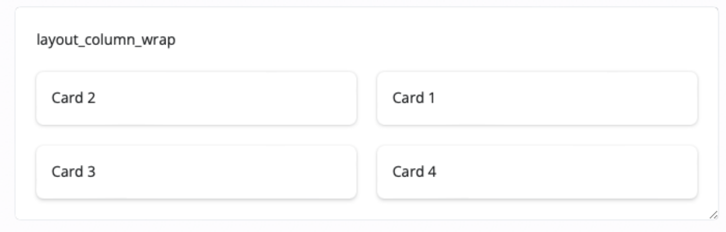


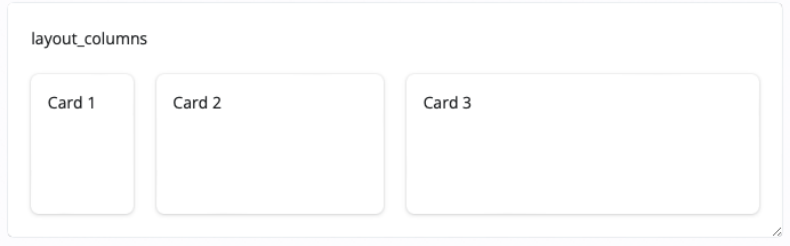
Card with tabbed tabset

How do we position this structure

https://shiny.posit.co/r/layouts/arrange/

- Layout!
  - HTML assumes a fixed page width (set by your browser/device)
  - So we typically divide that width into columns
  - That can be done homogenously or heterogeneously
- layout_column_wrap : sets everything to be the same size
  - takes width as fractions of total row width
- layout_columns : defaults to layout_column_wrap but also for lots of flexibility
  - takes argument col_widths; which must sum to 12 (Bootstrap CSS has 12 width units per row)



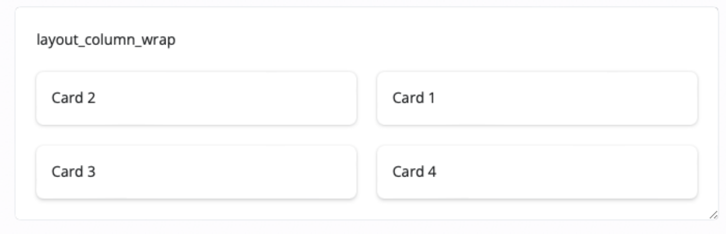


How do we position this structure?

https://shiny.posit.co/r/layouts/arrange/

- Layout!
  - HTML assumes a fixed page width (set by your browser/device)
  - So we typically divide that width into columns
  - That can be done homogenously or heterogeneously
- layout_column_wrap : sets everything to be the same size
  - takes width as fractions of total row width
- layout_columns : defaults to layout_column_wrap but also for lots of flexibility
  - takes argument col_widths; which must sum to 12 (Bootstrap CSS has 12 width units per row)

HTML novices should start here. Leads to very fast prototyping!



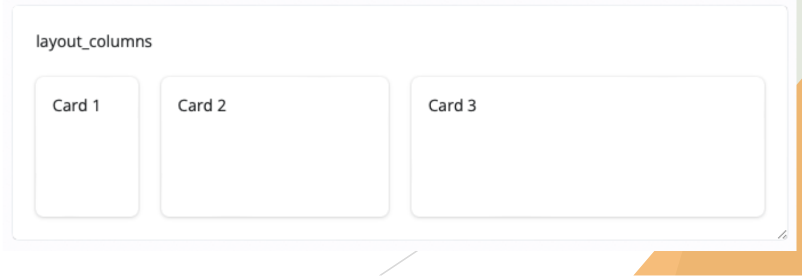
layout_column_wrap

Card 2

Card 1

Card 3

Card 4



layout_columns

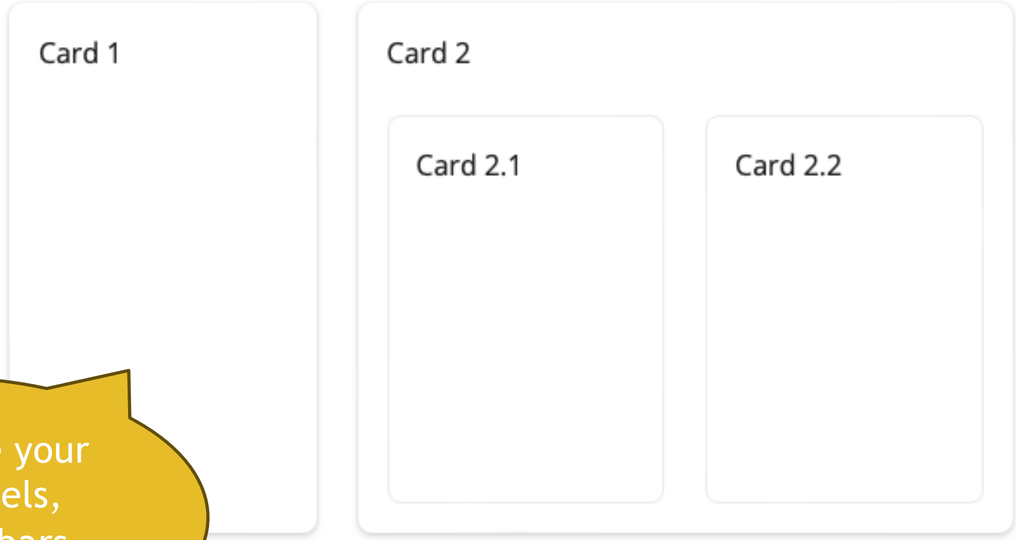
Card 1

Card 2

Card 3

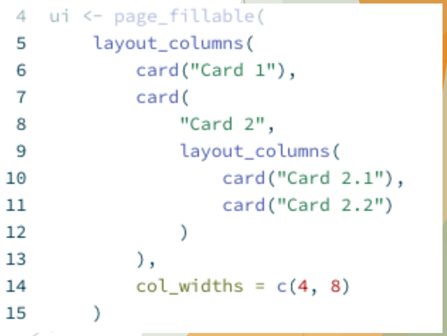
How do we position this structure?

https://shiny.posit.co/r/layouts/arrange/



Always give your page, panels, cards, sidebars, and widgets **titles**

layouts can be used to nest panels/cards



```
ui <- page_fillable(
    layout_columns(
        card("Card 1"),
        card(
            "Card 2",
            layout_columns(
                card("Card 2.1"),
                card("Card 2.2")
            )
        ),
        col_widths = c(4, 8)
    )
```

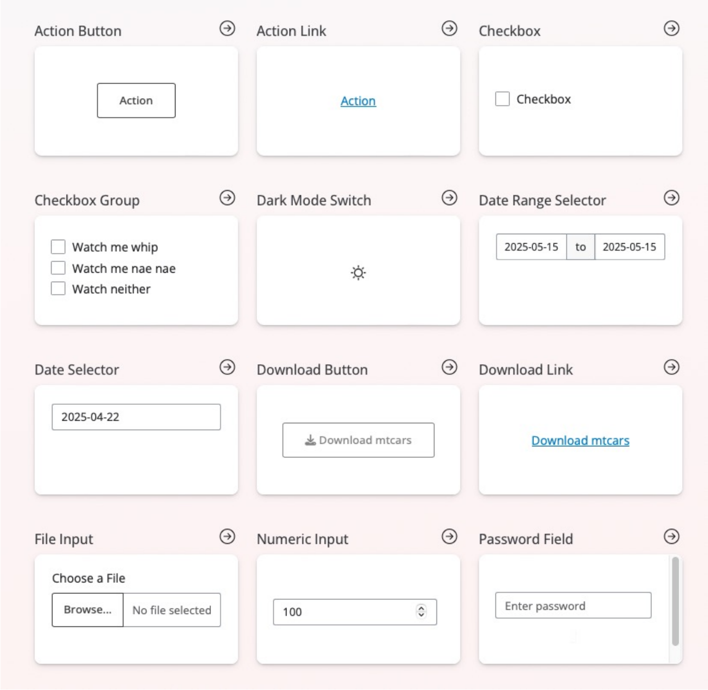
10 minute break

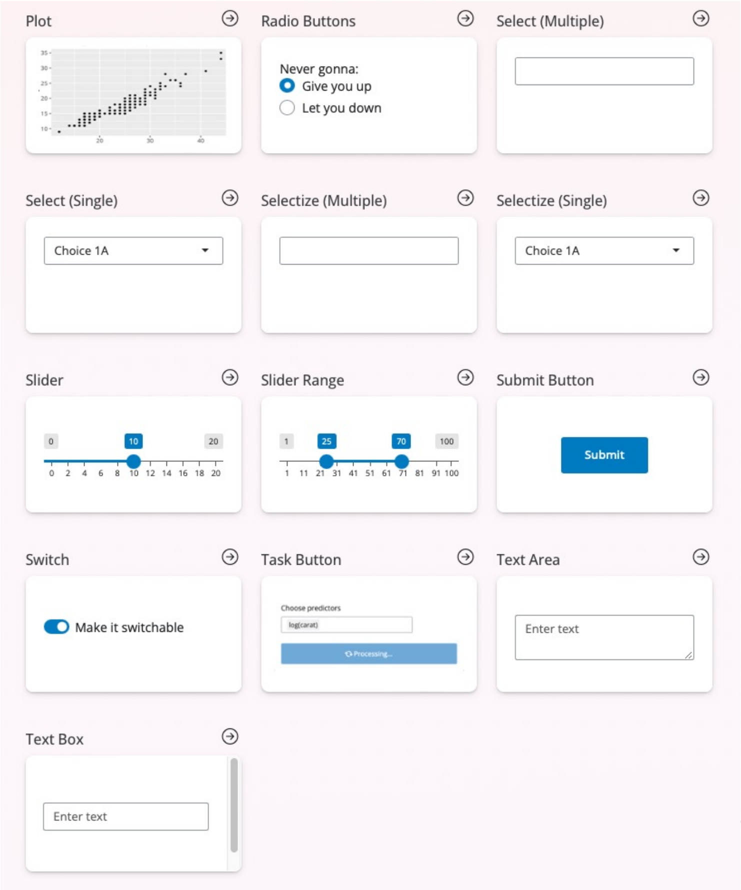
UI Design principles

- **Simplicity**
- Consistency
- Visual Hierarchy
- Feedback and Responsiveness
- Accessibility
- Clarity
- **User Control**
- Error Prevention and Recovery
- **Aesthetics and Visual Appeal**
- Scalability and Adaptability

- **UI Design choices**
- ~~Really for family of apps or b/w pages~~
- Layout choice
- ~~Alerts &~~ Reactivity >> tomorrow!
- ~~Walkthrough/Tooltips~~
- Layout design choices
- **Widget choices**
- App design >> this afternoon!
- **Aesthetic choices**
- Layout design choices

Widgets!

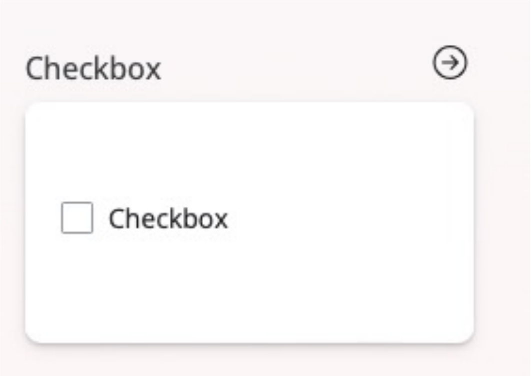


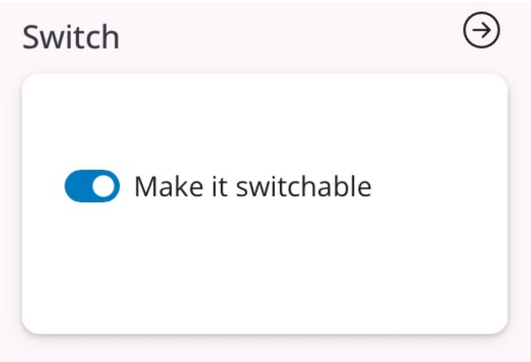


https://github.com/nanxstats/awesome-shiny-extensions

Inputs to R – TRUE/FALSE

- The output of these **widgets** will produce TRUE/FALSE outputs.
- These are useful for:
  - turning features on/off
  - activating behaviors in a simulation

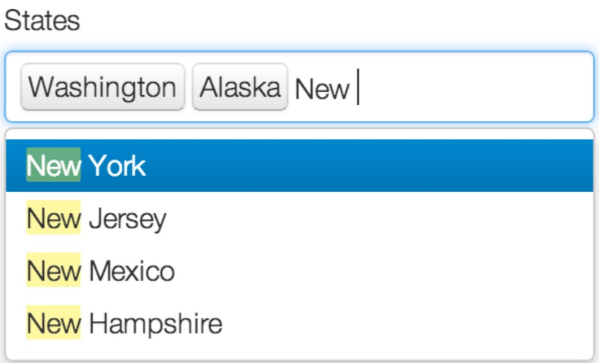




Input to R – Categorical values

- The output of these widgets will be one character or factor value
- If you want to default to no choice you need:

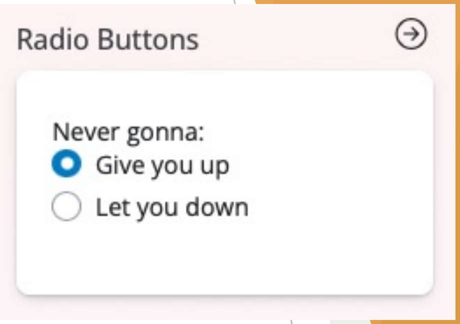
  -
    ```
    selectInput('in', 'Options',
    c(Choose=’’, ‘Choice 1’ = 1,
    ‘Choice 2’ = 2))
    ```
- Selectize is built on selectize.js and is more flexible as it allows users to search



States

Washington Alaska New

New York
New Jersey
New Mexico
New Hampshire

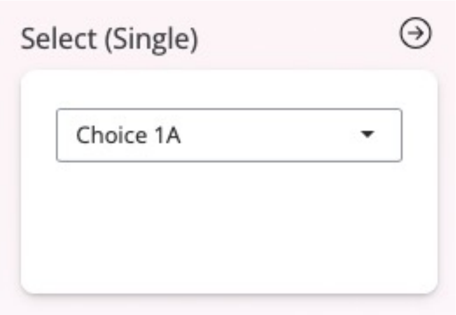


Radio Buttons

Never gonna:

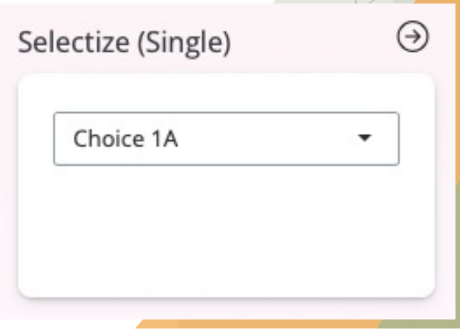
● Give you up

○ Let you down



Select (Single)

Choice 1A



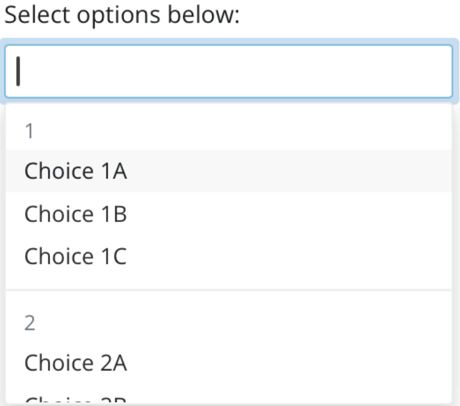
Selectize (Single)

Choice 1A

Input to R – Categorical values

- The output of these widgets will be one character or factor value
- Setting a default blank is same as single
- Checkbox Group is useful for a few choices
- Selectize is far more useful here especially as the choices get lengthy

You can also group your choices (confusingly this does not impact what you can choose)


Select options below:

1

Choice 1A

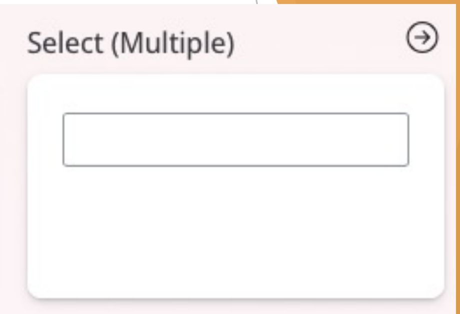
Choice 1B

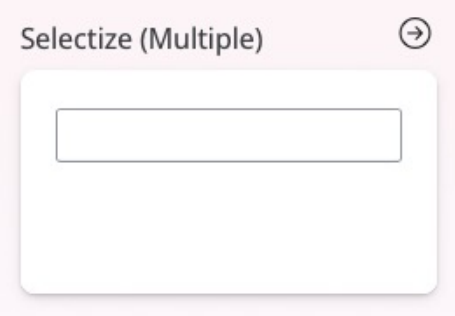
Choice 1C

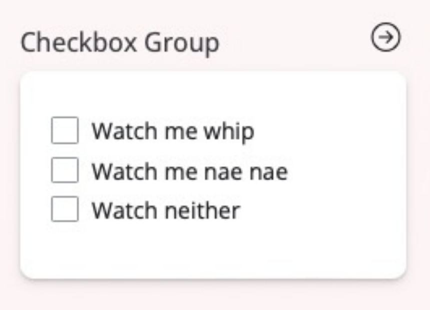
2

Choice 2A

Choice 2B


Select (Multiple)


Selectize (Multiple)

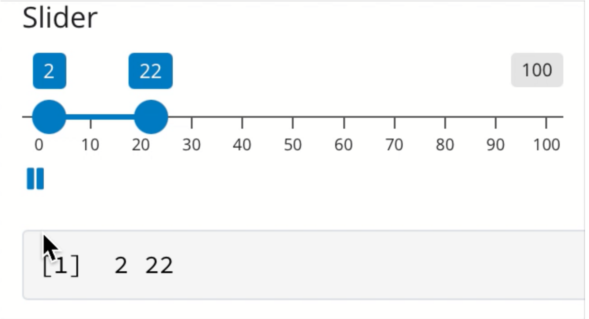

Checkbox Group

- □ Watch me whip
- □ Watch me nae nae
- □ Watch neither

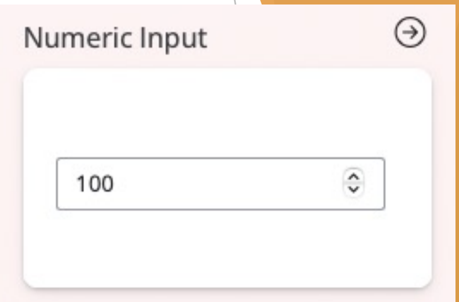
Input to R - Continuous values

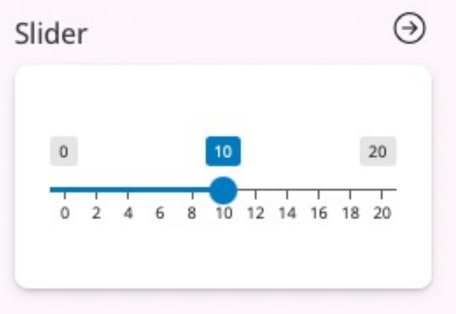
- The output of these widgets will be a continuous value (or two in the case of slider range)
- Almost never do you want to do `numericInput`
  - Users can supply any value and cause errors

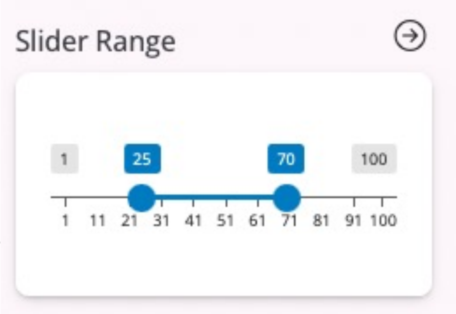
- Slider is most useful but choose your bounds (min, max) carefully (i.e. test them)
- Slider range is useful for setting up sequences
- Can also set up animations to show users a presentation



```
[1]  2 22
```

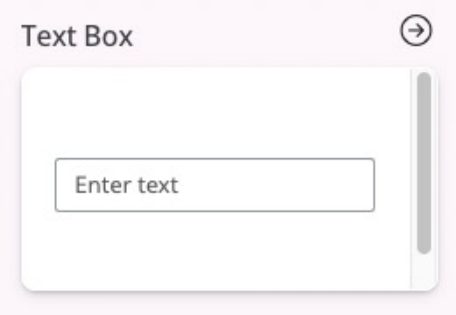


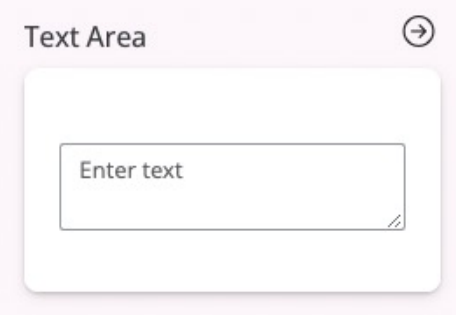




Input to R – Text values

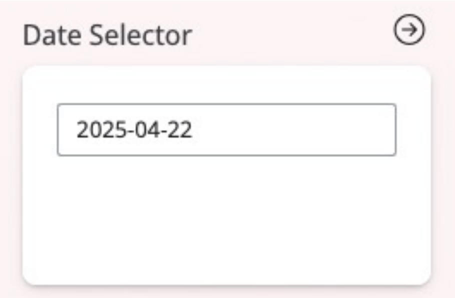
- The output of these widgets will be a character string
- These are useful for having users define labels on plots or adding text to a report
- Haven’t seen too much use for them





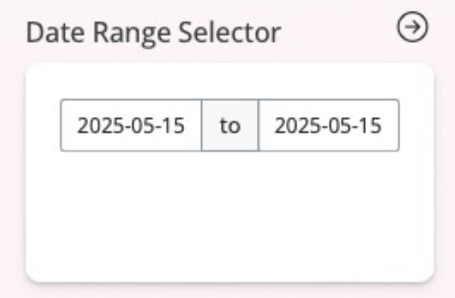
Input to R - Dates

- The output of these widgets will be a Date class object in R
- You should set bounds (min, max)
- lots of options for customizing visuals



Date Selector

2025-04-22

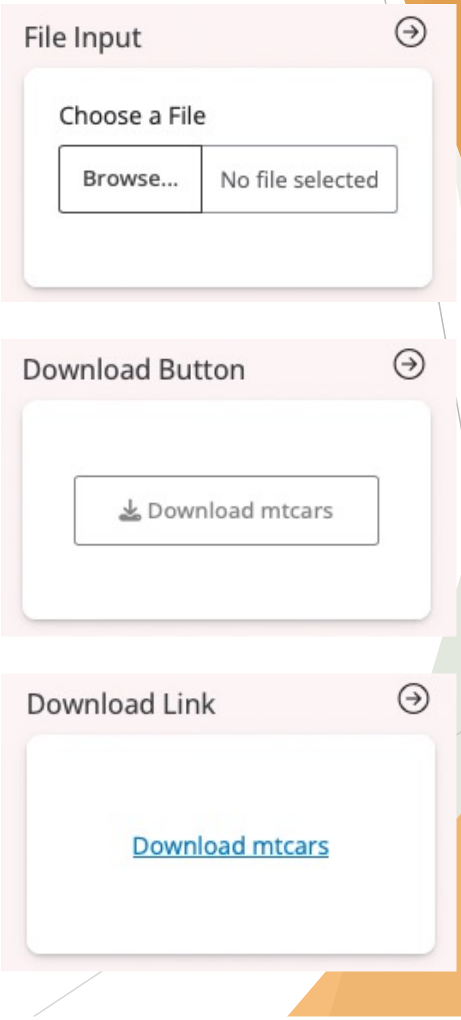


Date Range Selector

2025-05-15 to 2025-05-15

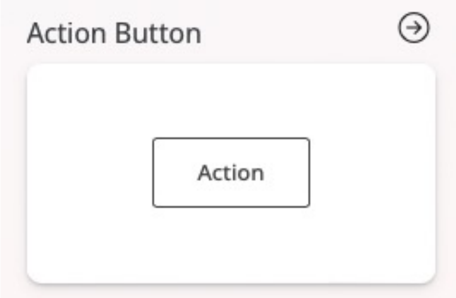
User input/output data

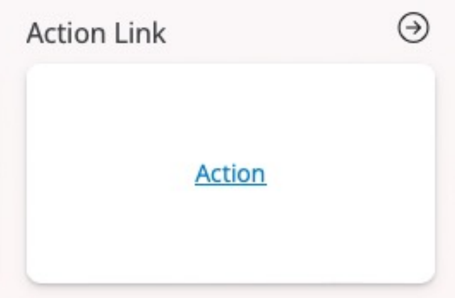
- File Input allows users to upload data
  - Super common for users to screw this up and generate errors
  - Typically needs detailed instructors and error handling (most novices struggle with this)
- Download button = Download link
  - Allows users to download something
  - You have to setup what this download is
  - You could have a Rmd doc be generated on download

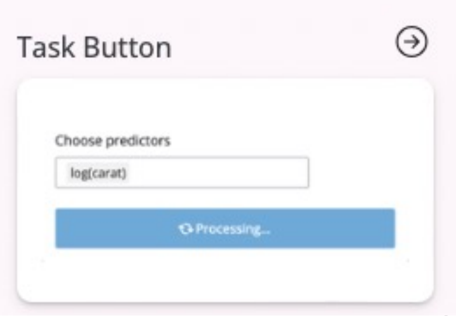


User controlled reactivity

- Action Button = Action Link, just visually differs
- Actions can:
  - Command – do something when action
  - Delay reactions – link another input to action
  - Switch – Just use a switch
  - Clear – set values to some default
  - Tabs are a hidden action button that can be used to trigger the four behaviors above
- Task are delay reactions for long tasks (like fitting a big model)
- Submit are delay reactions for ALL inputs

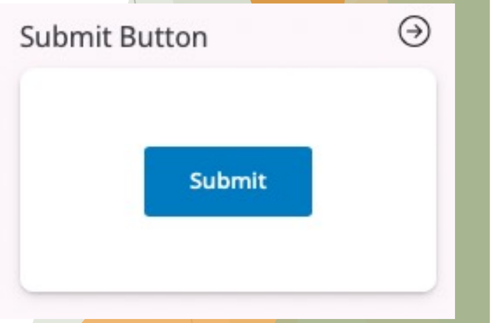






Choose predictors

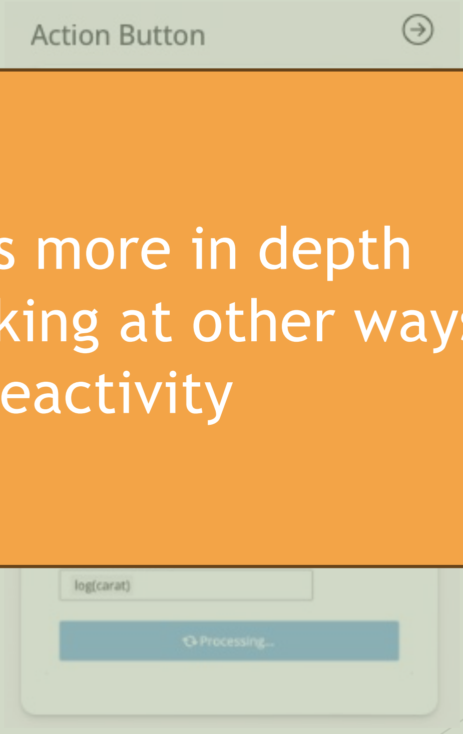
log(carat)

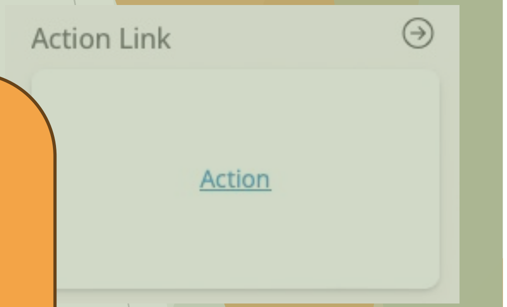


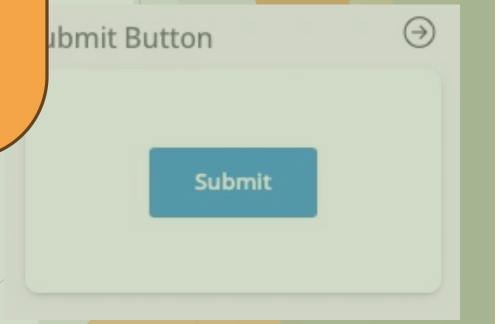
User controlled reactivity

- Action Button = Action Link, just visually differs
- Actions can:
  - Command - c
  - Delay reactio
  - Switch - Just
  - Clear - set va
  - Tabs are a hi used to trigg
- Task are delay reactions for long tasks (like fitting a big model)
- Submit are delay reactions for ALL inputs

We will cover this more in depth tomorrow when looking at other ways to set up reactivity







5 minute break