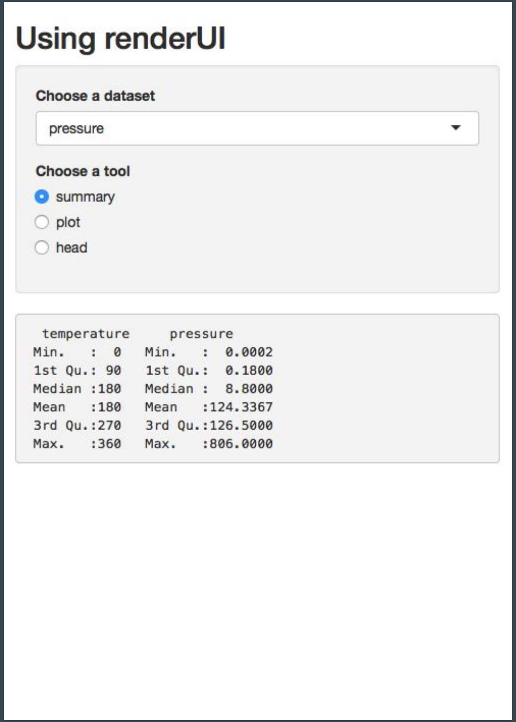
Dynamic Shiny interfaces with `renderUI` and `insertUI` …

Bárbara Borges Ribeiro

rstudio::conf 2017

https://github.com/bborgesr/rstudio-conf2017

What is dynamic UI? (live app)



Using renderUI

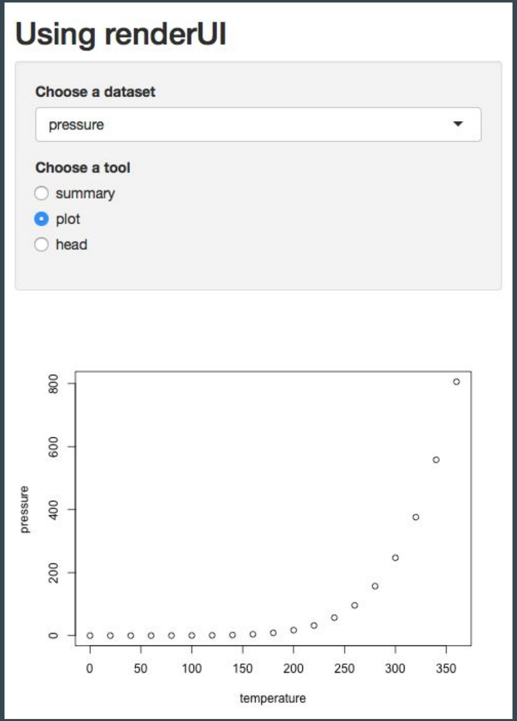
Choose a dataset

pressure

Choose a tool

[✓ summary] plot head

```
 temperature      pressure      
Min.   :  0   Min.   :  0.0002  
1st Qu.: 90   1st Qu.:  0.1800  
Median :180   Median :  8.8000  
Mean   :180   Mean   :124.3367  
3rd Qu.:270   3rd Qu.:126.5000  
Max.   :360   Max.   :806.0000  
```



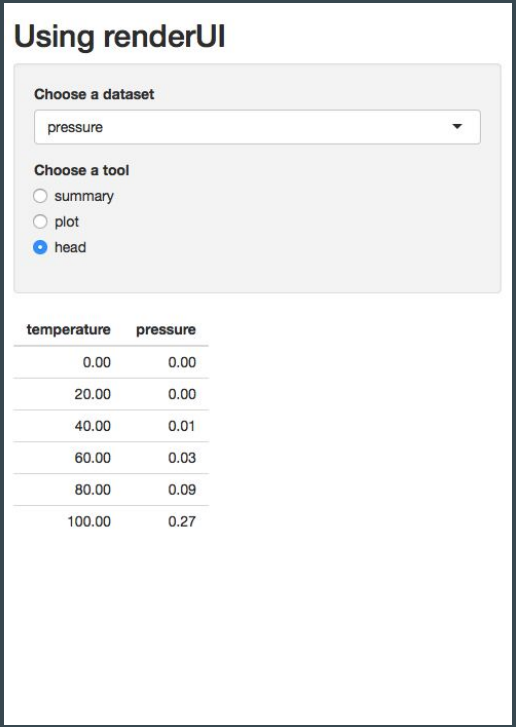
Using renderUI

Choose a dataset

pressure

Choose a tool

summary [✓ plot] head



Using renderUI

Choose a dataset

pressure

Choose a tool

summary plot [✓ head]

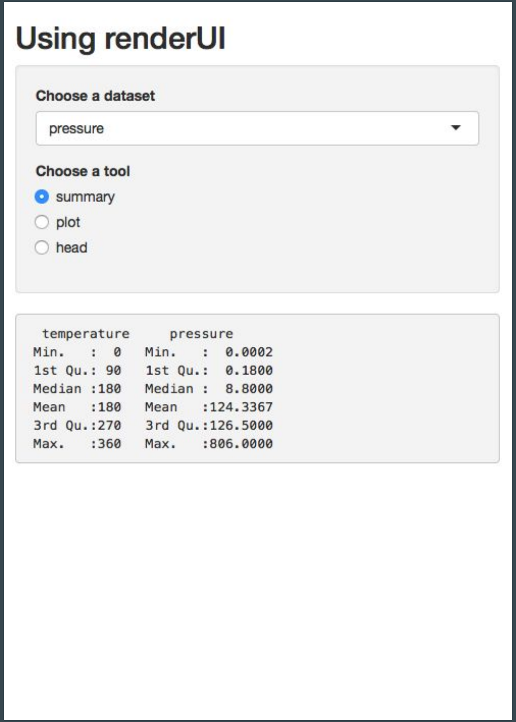
temperature	pressure
0.00	0.00
20.00	0.00
40.00	0.01
60.00	0.03
80.00	0.09
100.00	0.27

https://github.com/bborgesr/rstudio-conf2017

Goals

- Why we may need dynamic UI in a Shiny app
- Our first answer to this: `renderUI`
- Our newest addition: `insertUI` / `removeUI`

renderUI (live app)



Using renderUI

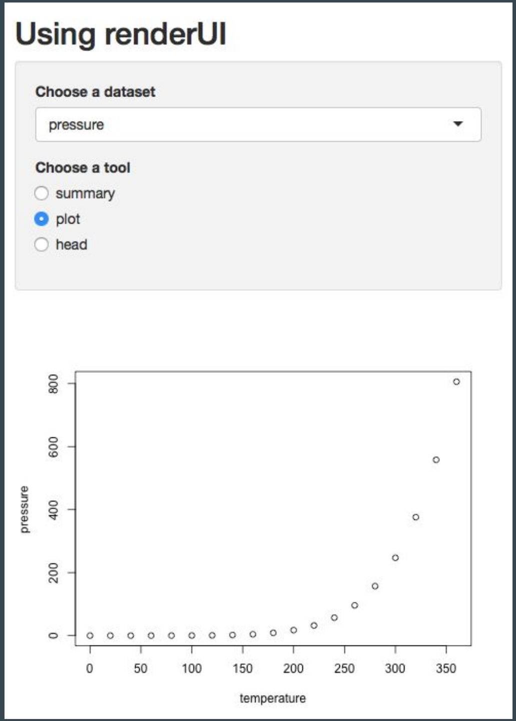
Choose a dataset

pressure

Choose a tool

[✓ summary]
plot
head

```
 temperature     pressure     
Min.   :  0   Min.   :  0.0002  
1st Qu.: 90   1st Qu.:  0.1800  
Median :180   Median :  8.8000  
Mean   :180   Mean   :124.3367  
3rd Qu.:270   3rd Qu.:126.5000  
Max.   :360   Max.   :806.0000  
```



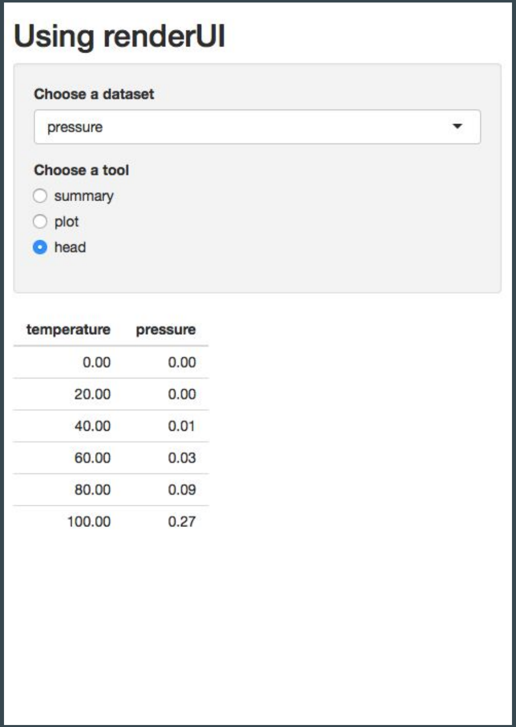
Using renderUI

Choose a dataset

pressure

Choose a tool

summary
[✓ plot]
head



Using renderUI

Choose a dataset

pressure

Choose a tool

summary
plot
[✓ head]

temperature	pressure
0.00	0.00
20.00	0.00
40.00	0.01
60.00	0.03
80.00	0.09
100.00	0.27

https://github.com/bborgesr/rstudio-conf2017

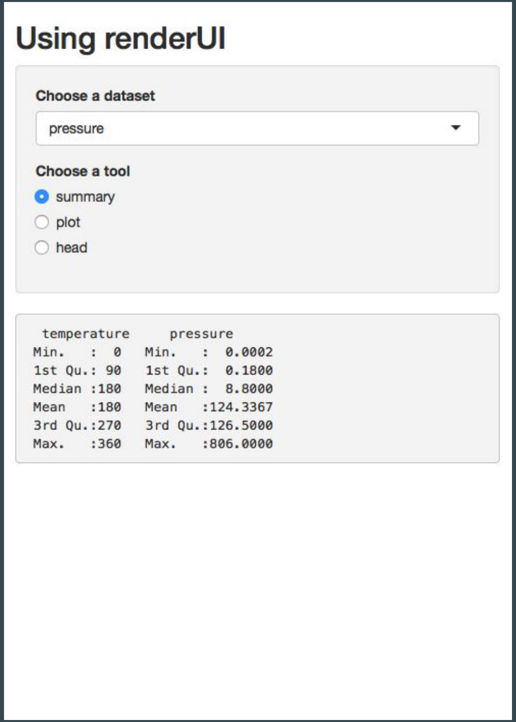
renderUI (live app)

```
# ui
uiOutput(id)


# server
output[[id]] <- renderUI({ ... })

output[[id]] <- renderUI({ tagList(...) })
```

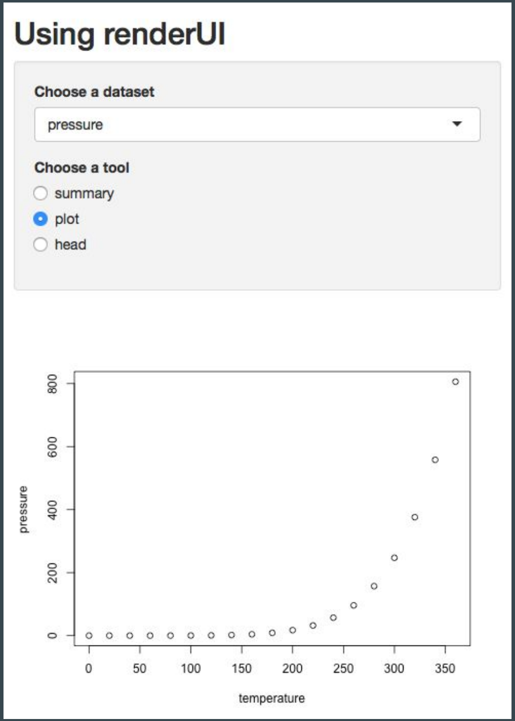
renderUI (live app)

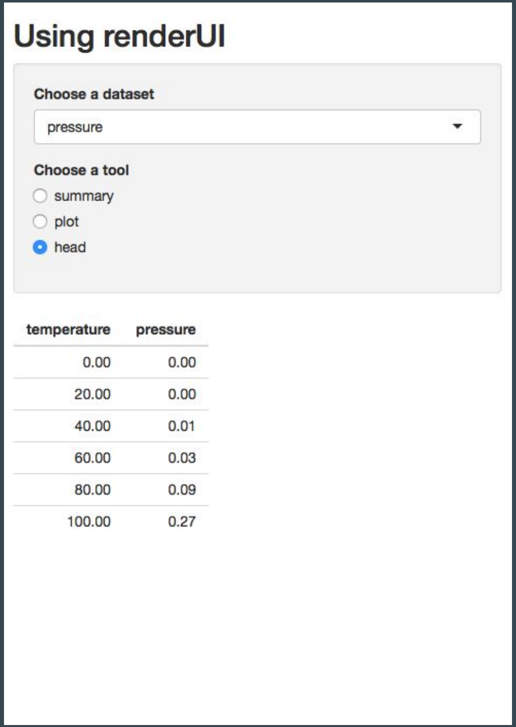
```
shinyApp(
  ui = fluidPage(
    selectInput('data', 'Choose a dataset', c('rock', 'pressure', 'cars')),
    radioButtons('tool', 'Choose a tool', c('summary', 'plot', 'head')),
    uiOutput('result')
  ),

  server = function(input, output, session) {
    dataset <- reactive({ switch(input$data, 'rock' = rock, 'pressure' = pressure, 'cars' = cars) })

    output$result <- renderUI({
      switch(input$tool,
        'summary' = verbatimTextOutput('summary'),
        'plot' = plotOutput('plot'),
        'head' = tableOutput('head'))
    })

    output$summary <- renderPrint({ summary(dataset()) })
    output$plot <- renderPlot({ plot(dataset()) })
    output$head <- renderTable({ head(dataset()) })
  }
)
```

renderUI (live app)



```
 temperature     pressure
Min.   :  0   Min.   :  0.0002
1st Qu.: 90   1st Qu.:  0.1800
Median :180   Median :  8.8000
Mean   :180   Mean   :124.3367
3rd Qu.:270   3rd Qu.:126.5000
Max.   :360   Max.   :806.0000
```

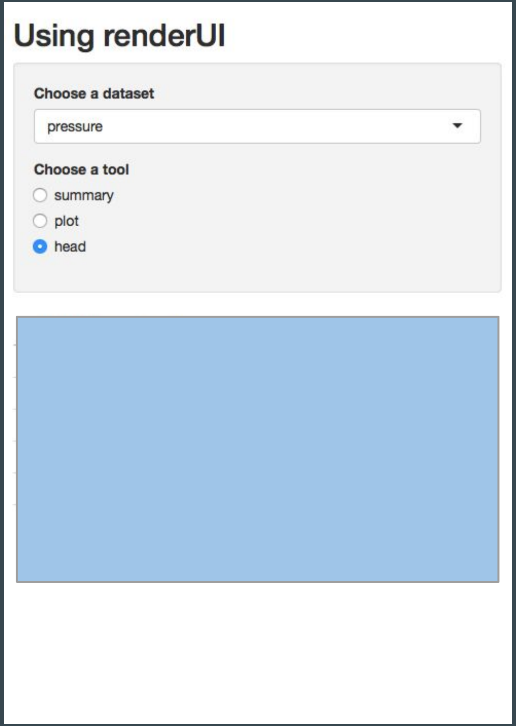




temperature	pressure
0.00	0.00
20.00	0.00
40.00	0.01
60.00	0.03
80.00	0.09
100.00	0.27

https://github.com/bborgesr/rstudio-conf2017

renderUI (live app)

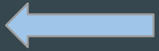

Using renderUI

Choose a dataset

pressure

Choose a tool

- summary
- plot
- [✓ head]



predefined `uiOutput` slot

depending on user input, can become

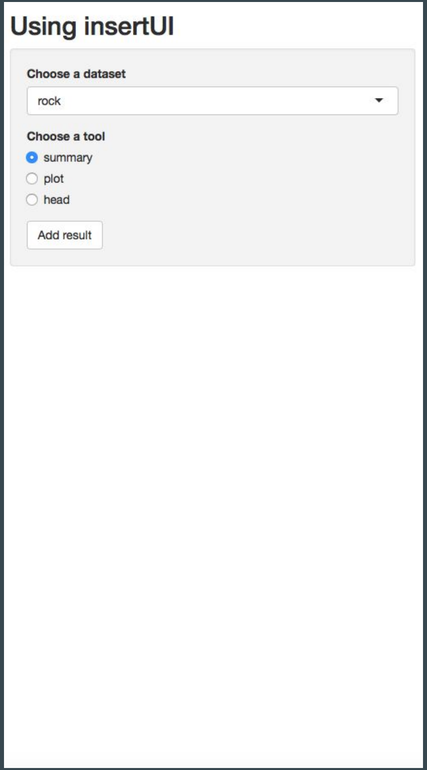
- `verbatimTextOutput`
- `plotOutput`
- `tableOutput`

Is this always sufficient?

- You need to have a `uiOutput` slot for each element beforehand
- If you want to add more elements, you have to re-render the whole slot

So, `renderUI` can become clunky, especially when you want to add independent UI elements, instead of just replacing one by the other over and over again.

insertUI (live app)



Using insertUI

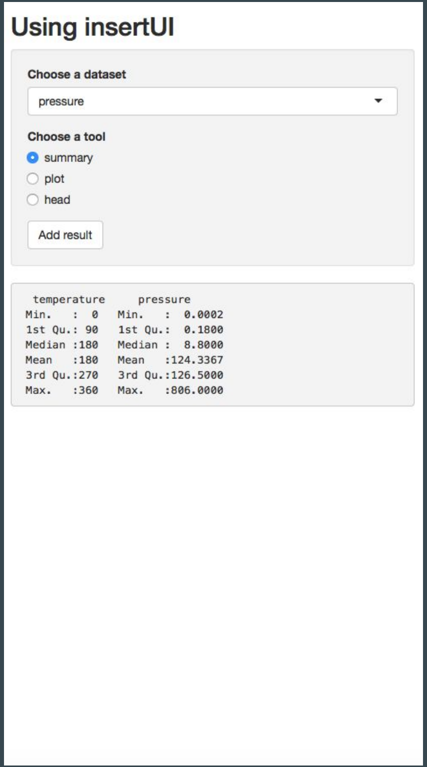
Choose a dataset

rock

Choose a tool

[✓ summary]
plot
head

Add result



Using insertUI

Choose a dataset

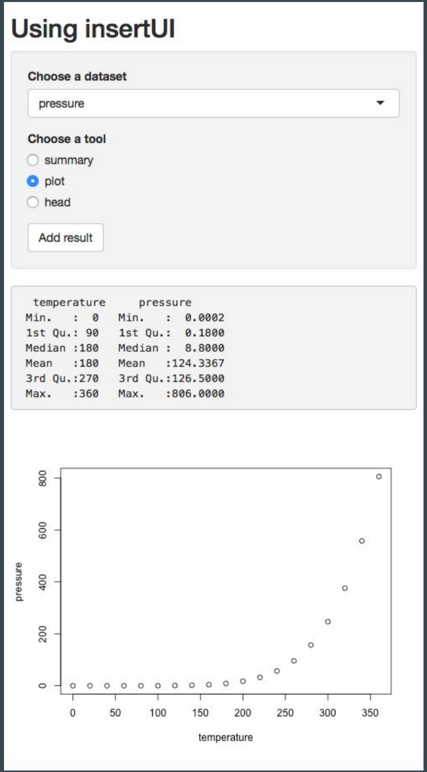
pressure

Choose a tool

[✓ summary]
plot
head

Add result

```
  temperature      pressure
 Min.   :  0   Min.   :  0.0002
 1st Qu.: 90   1st Qu.:  0.1800
 Median :180   Median :  8.8000
 Mean   :180   Mean   :124.3367
 3rd Qu.:270   3rd Qu.:126.5000
 Max.   :360   Max.   :806.0000
```



Using insertUI

Choose a dataset

pressure

Choose a tool

summary
[✓ plot]
head

Add result

```
  temperature      pressure
 Min.   :  0   Min.   :  0.0002
 1st Qu.: 90   1st Qu.:  0.1800
 Median :180   Median :  8.8000
 Mean   :180   Mean   :124.3367
 3rd Qu.:270   3rd Qu.:126.5000
 Max.   :360   Max.   :806.0000
```

https://github.com/bborgesr/rstudio-conf2017

insertUI (live app)

```
# ui
actionButton("add", "Add UI")
div(id = "placeholder")    # we want to add new content after
this


# server
observeEvent(input$add, {
  insertUI(selector = "#placeholder",
           where = "afterEnd",
           ui = tagList(...))
})
```

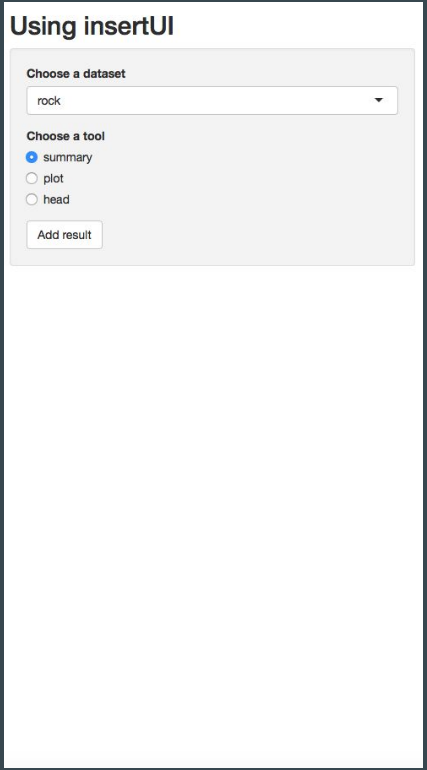
removeUI

```
# ui
actionButton("rm", "Remove UI")
div(id = "foo")     # we want to remove this


# server
observeEvent(input$rm, {
  removeUI(selector = "#foo")
})
```

insertUI (live app)

```
shinyApp(
  ui = fluidPage(
    selectInput('data', 'Choose a dataset', c('rock', 'pressure', 'cars')),
    radioButtons('tool', 'Choose a tool', c('summary', 'plot', 'head')),
    actionButton('add', 'Add result'),
    div(id = 'placeholder')
  ),
  server = function(input, output, session) {
    dataset <- reactive({ switch(input$data, 'rock' = rock, 'pressure' = pressure, 'cars' = cars) })
    observeEvent(input$add, {
      id <- paste0(input$tool, input$add)
      insertUI('#placeholder',
        ui = switch(input$tool,
          'summary' = verbatimTextOutput(id),
          'plot' = plotOutput(id),
          'head' = tableOutput(id))
      )
      output[[id]] <-
        if (input$tool == 'summary') renderPrint({ summary(isolate(dataset())) })
        else if (input$tool == 'plot') renderPlot({ plot(isolate(dataset())) })
        else if (input$tool == 'head') renderTable({ head(isolate(dataset())) })
    })
  }
)
```

insertUI (live app)



Using insertUI

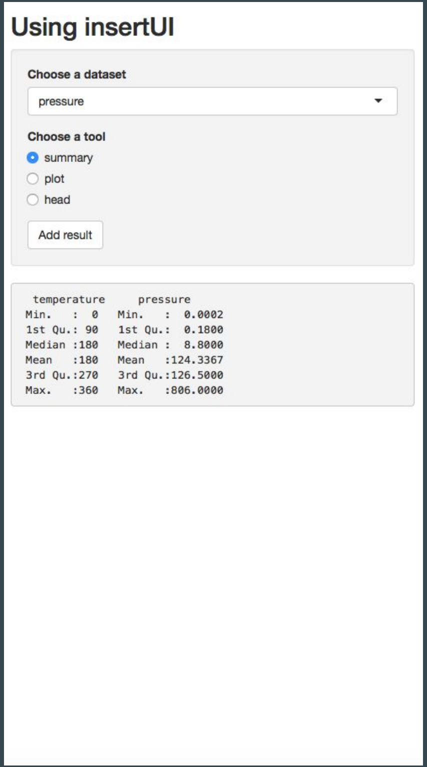
Choose a dataset

rock

Choose a tool

[✓ summary]
plot
head

Add result



Using insertUI

Choose a dataset

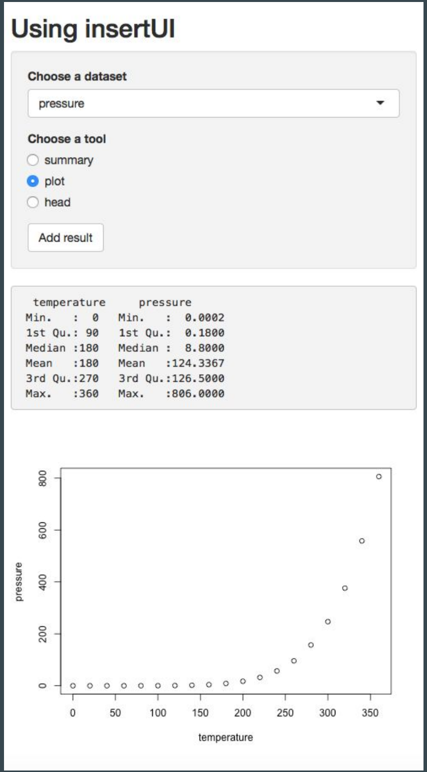
pressure

Choose a tool

[✓ summary]
plot
head

Add result

```
  temperature     pressure     
 Min.   :  0   Min.   :  0.0002  
 1st Qu.: 90   1st Qu.:  0.1800  
 Median :180   Median :  8.8000  
 Mean   :180   Mean   :124.3367  
 3rd Qu.:270   3rd Qu.:126.5000  
 Max.   :360   Max.   :806.0000  
```



Using insertUI

Choose a dataset

pressure

Choose a tool

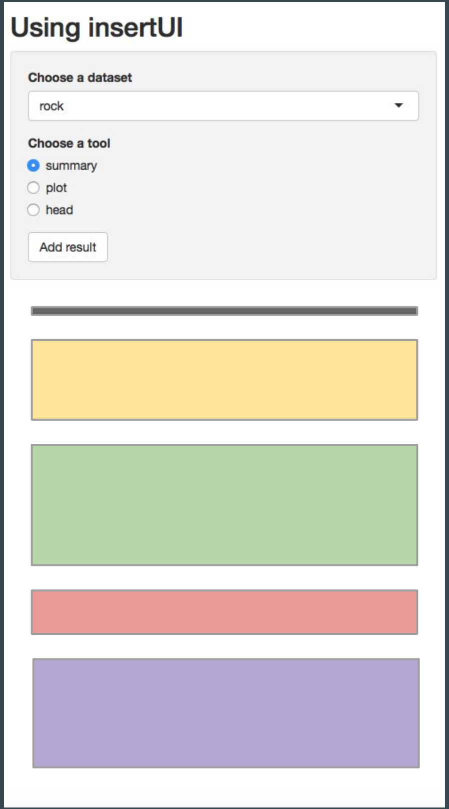
summary
[✓ plot]
head

Add result

```
  temperature     pressure     
 Min.   :  0   Min.   :  0.0002  
 1st Qu.: 90   1st Qu.:  0.1800  
 Median :180   Median :  8.8000  
 Mean   :180   Mean   :124.3367  
 3rd Qu.:270   3rd Qu.:126.5000  
 Max.   :360   Max.   :806.0000  
```

https://github.com/bborgesr/rstudio-conf2017

insertUI (live app) & renderUI (live app)

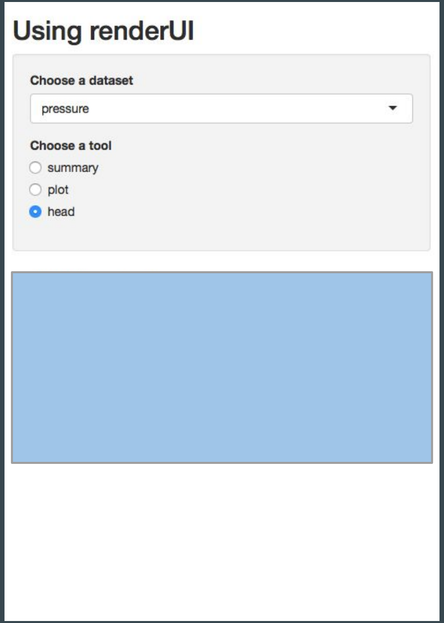


placeholder div

0, 1 or more UI elements (independent from each other)

* in direct response to a button click;

* depending on user input, each can be:
  - verbatimTextOutput
  - plotOutput
  - tableOutput



predefined uiOutput slot

depending on user input, can become:
- verbatimTextOutput
- plotOutput
- tableOutput

`insertUI` (live app) & `renderUI` (live app)

- `renderUI` feels “Shiny-like”
- `renderUI` is safer, less likely to get you in trouble

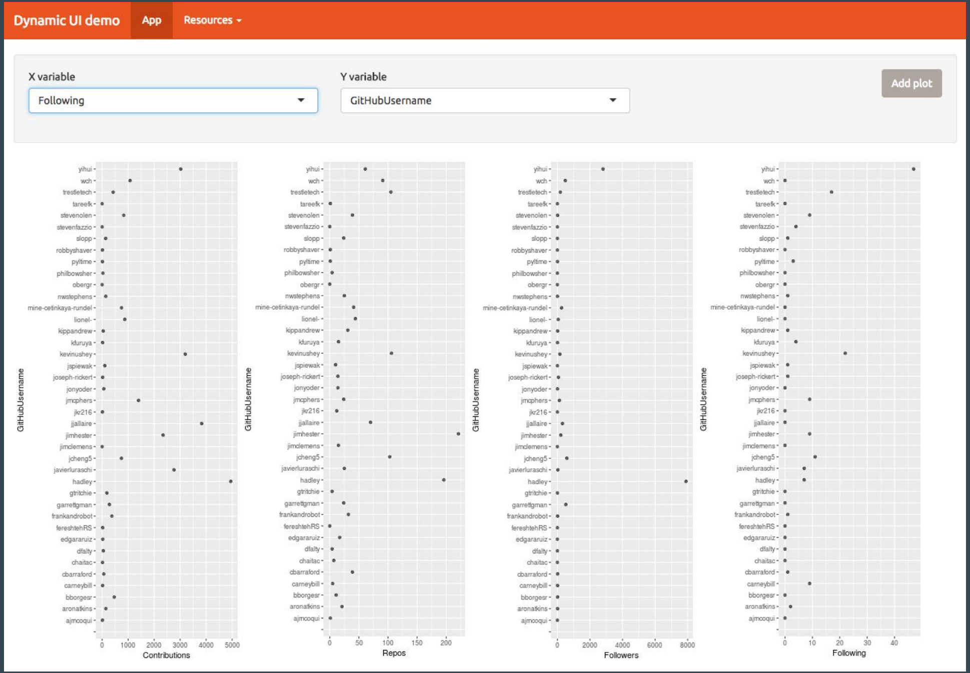
- `renderUI` becomes unwieldy for adding more than one thing
- `insertUI` is more flexible (you add anything!)

- `insertUI` requires you to create a more sophisticated mental model
- `insertUI` is harder to debug
- `insertUI` can result in longer code
- `insertUI` is not trivial to bookmark

Bookmarking with `renderUI` (live app)

```r
enableBookmarking(store = "url")
shinyApp(
  ui = function(req) {
    fluidPage(
      selectInput('data', 'Choose a dataset', c('rock', 'pressure', 'cars')),
      radioButtons('tool', 'Choose a tool', c('summary', 'plot', 'head')),
      uiOutput('result'),
      bookmarkButton()
    )
  },

  server = function(input, output, session) {
    dataset <- reactive({ switch(input$data, 'rock' = rock, 'pressure' = pressure, 'cars' = cars) })

    output$result <- renderUI({
      switch(input$tool,
        'summary' = verbatimTextOutput('summary'),
        'plot' = plotOutput('plot'),
        'head' = tableOutput('head'))
    })

    output$summary <- renderPrint({ summary(dataset()) })
    output$plot <- renderPlot({ plot(dataset()) })
    output$head <- renderTable({ head(dataset()) })
  }
)
```

Awe and amaze with `insertUI` (live app)



X variable: Following

Y variable: GitHubUsername

https://github.com/bborgesr/rstudio-conf2017

Takeaways and resources

- Dynamic UI when you need to create UI in response to the user's choices

- `renderUI` when this is enough for you

- `insertUI` when you need more flexibility

Github repo with link to these slides and all the apps demoed:
https://github.com/bborgesr/rstudio-conf2017

Shiny website article about dynamic UI (lots of example apps!):
http://shiny.rstudio.com/articles/dynamic-ui.html

Documentation for `renderUI` and `insertUI`:
http://shiny.rstudio.com/reference/shiny/latest/renderUI.html
http://shiny.rstudio.com/reference/shiny/latest/insertUI.html

Email: barbara@rstudio.com