Shiny - A web framework for R

Bethany Yollin

March 19, 2018

About me

- Lived in the Pacific Northwest for over 20 years
- Attended the University of Washington earning a B.A. in Geography
- Using R for about 7 years
- Data scientist working in the transportation sector
- I like to tell stories with data!

About you

?

The basics

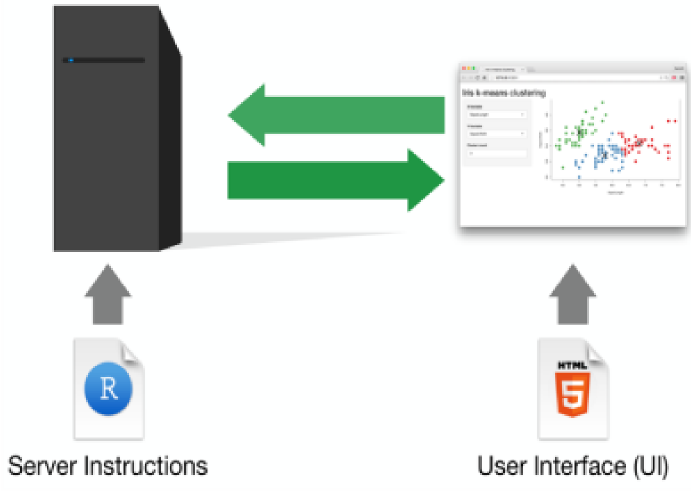
What is Shiny?

Shiny by **RStudio** is a **web framework** for **R**:

- Shiny enables R developers to share their analyses and visualizations on the web
- Shiny consists of two main components:
  - The `shiny` **R package** - allows users to develop Shiny applications in R (no web development experience required, though it may come in handy)
  - **Shiny Server** - allows users to deploy Shiny applications on the web (some Linux systems administration required)
- Shiny is constantly evolving and has a very active open-source community

Shiny application concepts

- A shiny application consists of two parts:
  - the user interface or UI (HTML/CSS)
  - the server logic (R/JavaScript)
- Shiny implements a reactive programming paradigm:
  1. Observing **input** from the user
  2. Performing **computations** based on these inputs
  3. Displaying **outputs** based on these computations



A simple Shiny app

Case-Shiller Home Price Indices

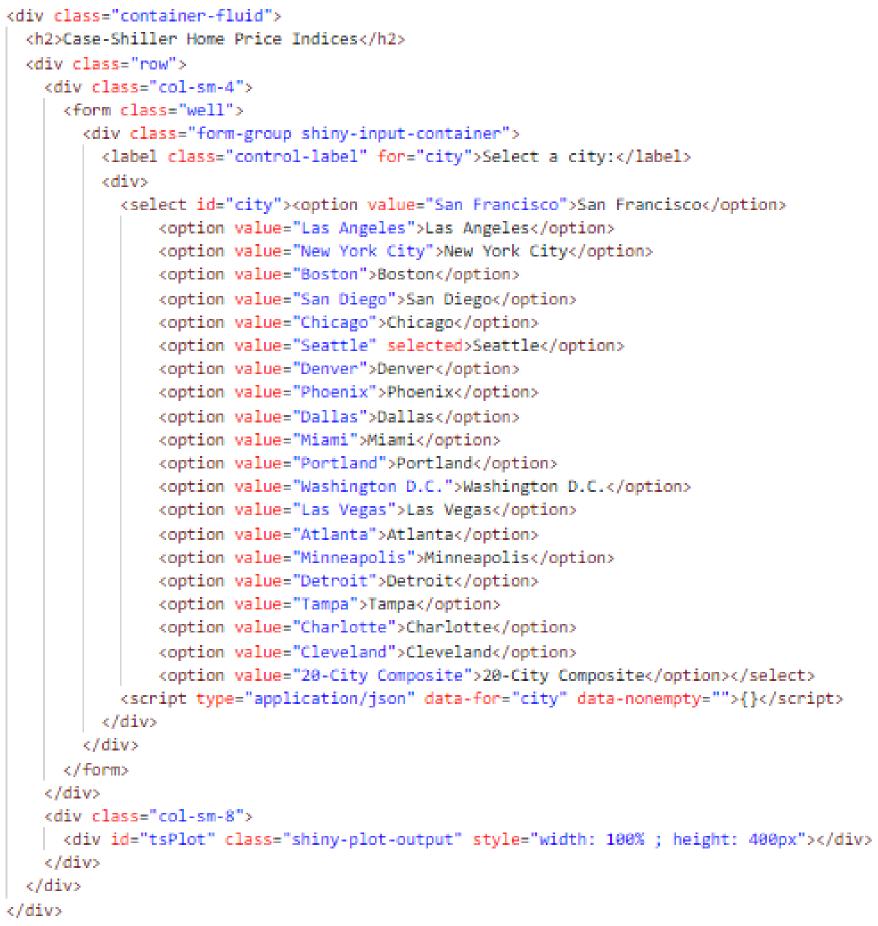
Select a city:

Seattle ▾

The UI

```
ui <- fluidPage(

    titlePanel("Case-Shiller Home Price Indices"),

    sidebarLayout(

        sidebarPanel(

            selectInput(inputId = 'city', label = 'Select a city:', choices = cities,
                        selected = 'Seattle')

        ),

        mainPanel(

            plotOutput(outputId = "tsPlot")

        )
    )
)
```

The UI



```
<div class="container-fluid">
  <h2>Case-Shiller Home Price Indices</h2>
  <div class="row">
    <div class="col-sm-4">
      <form class="well">
        <div class="form-group shiny-input-container">
          <label class="control-label" for="city">Select a city:</label>
          <div>
            <select id="city"><option value="San Francisco">San Francisco</option>
                <option value="Las Angeles">Las Angeles</option>
                <option value="New York City">New York City</option>
                <option value="Boston">Boston</option>
                <option value="San Diego">San Diego</option>
                <option value="Chicago">Chicago</option>
                <option value="Seattle" selected>Seattle</option>
                <option value="Denver">Denver</option>
                <option value="Phoenix">Phoenix</option>
                <option value="Dallas">Dallas</option>
                <option value="Miami">Miami</option>
                <option value="Portland">Portland</option>
                <option value="Washington D.C.">Washington D.C.</option>
                <option value="Las Vegas">Las Vegas</option>
                <option value="Atlanta">Atlanta</option>
                <option value="Minneapolis">Minneapolis</option>
                <option value="Detroit">Detroit</option>
                <option value="Tampa">Tampa</option>
                <option value="Charlotte">Charlotte</option>
                <option value="Cleveland">Cleveland</option>
                <option value="20-City Composite">20-City Composite</option></select>
            <script type="application/json" data-for="city" data-nonempty="">{}</script>
          </div>
        </div>
      </form>
    </div>
    <div class="col-sm-8">
      <div id="tsPlot" class="shiny-plot-output" style="width: 100% ; height: 400px"></div>
    </div>
  </div>
</div>
```

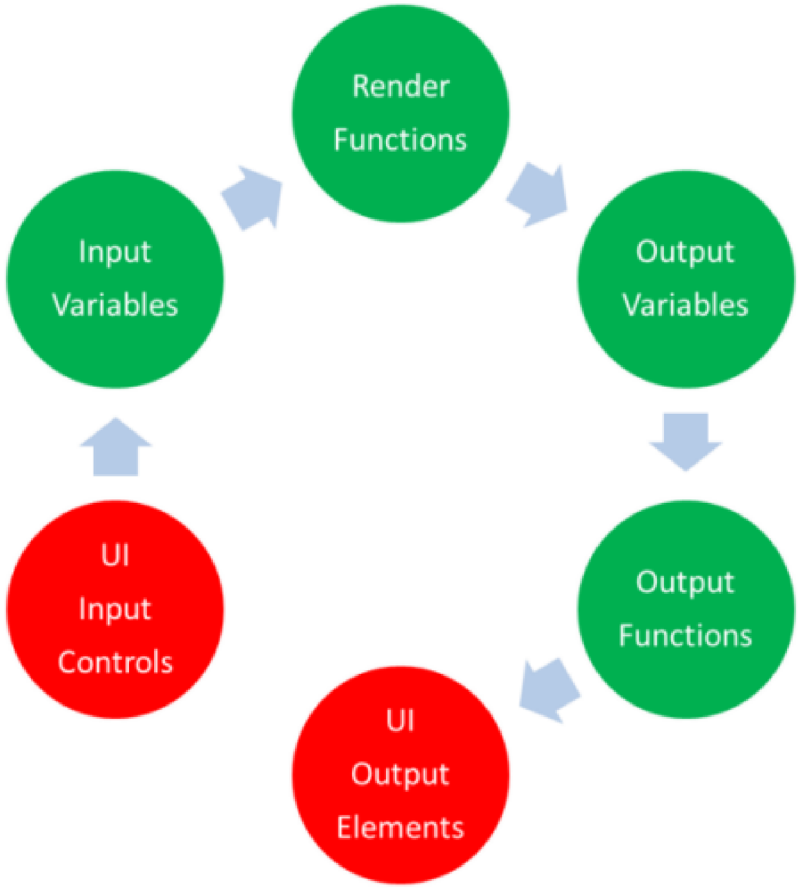
The server

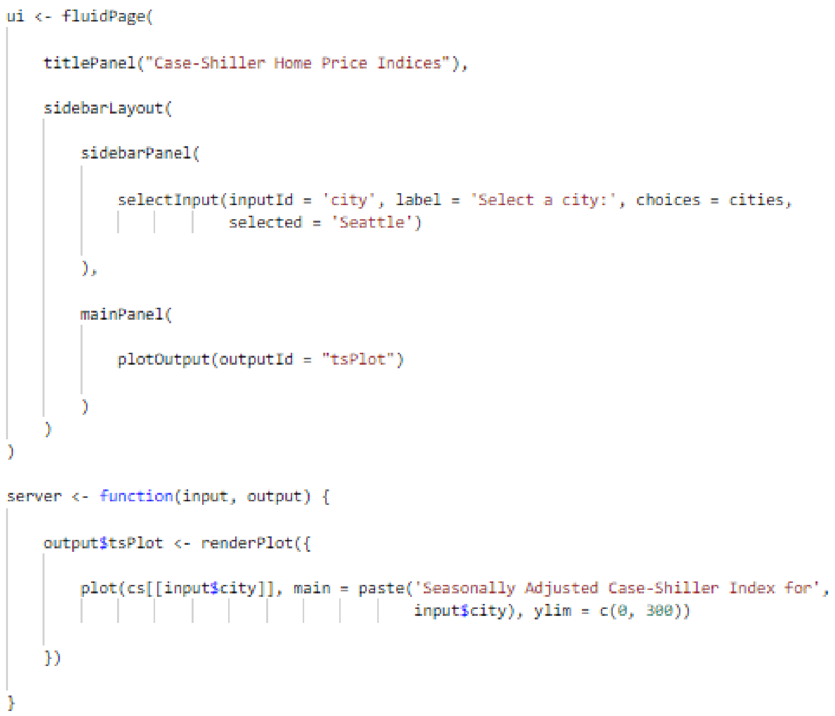
```
server <- function(input, output) {

    output$tsPlot <- renderPlot({

        plot(cs[[input$city]], main = paste('Seasonally Adjusted Case-Shiller Index for',
                                            input$city), ylim = c(0, 300))

    })

}

shinyApp(ui = ui, server = server)
```

R-Ladies

More on reactivity

General reactive workflow for Shiny applications

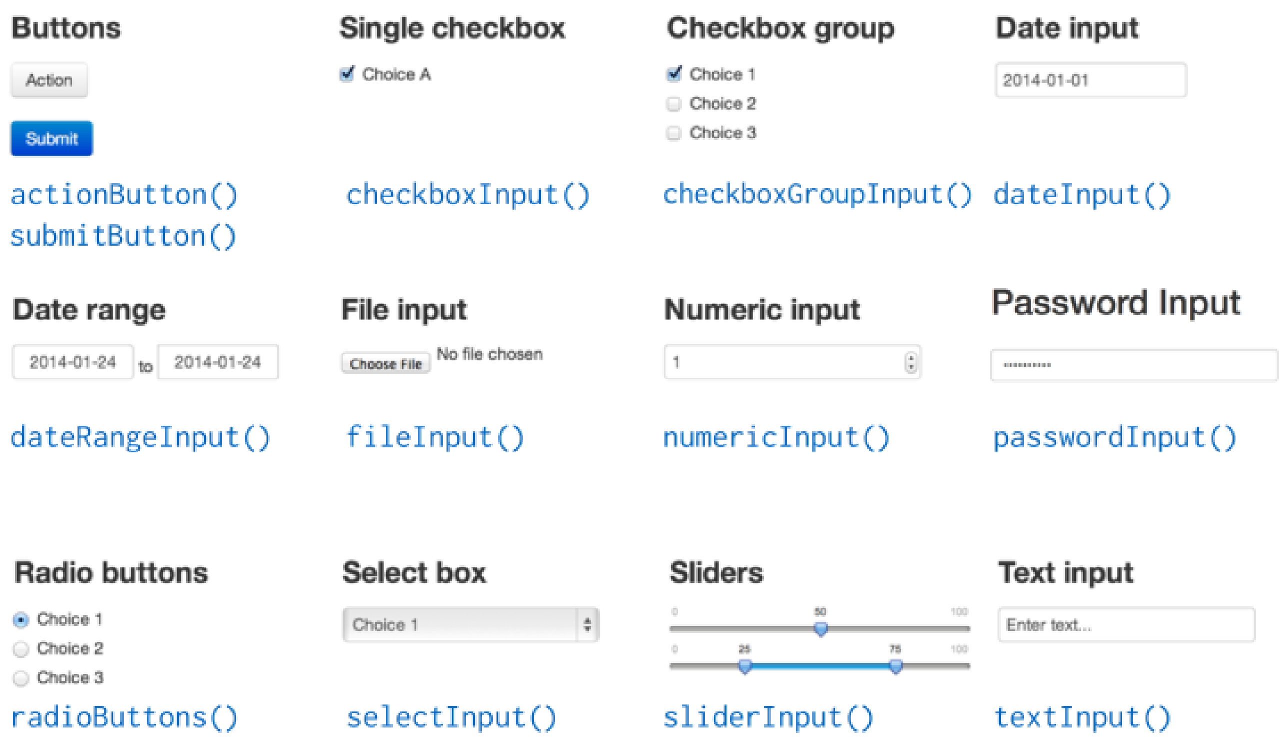




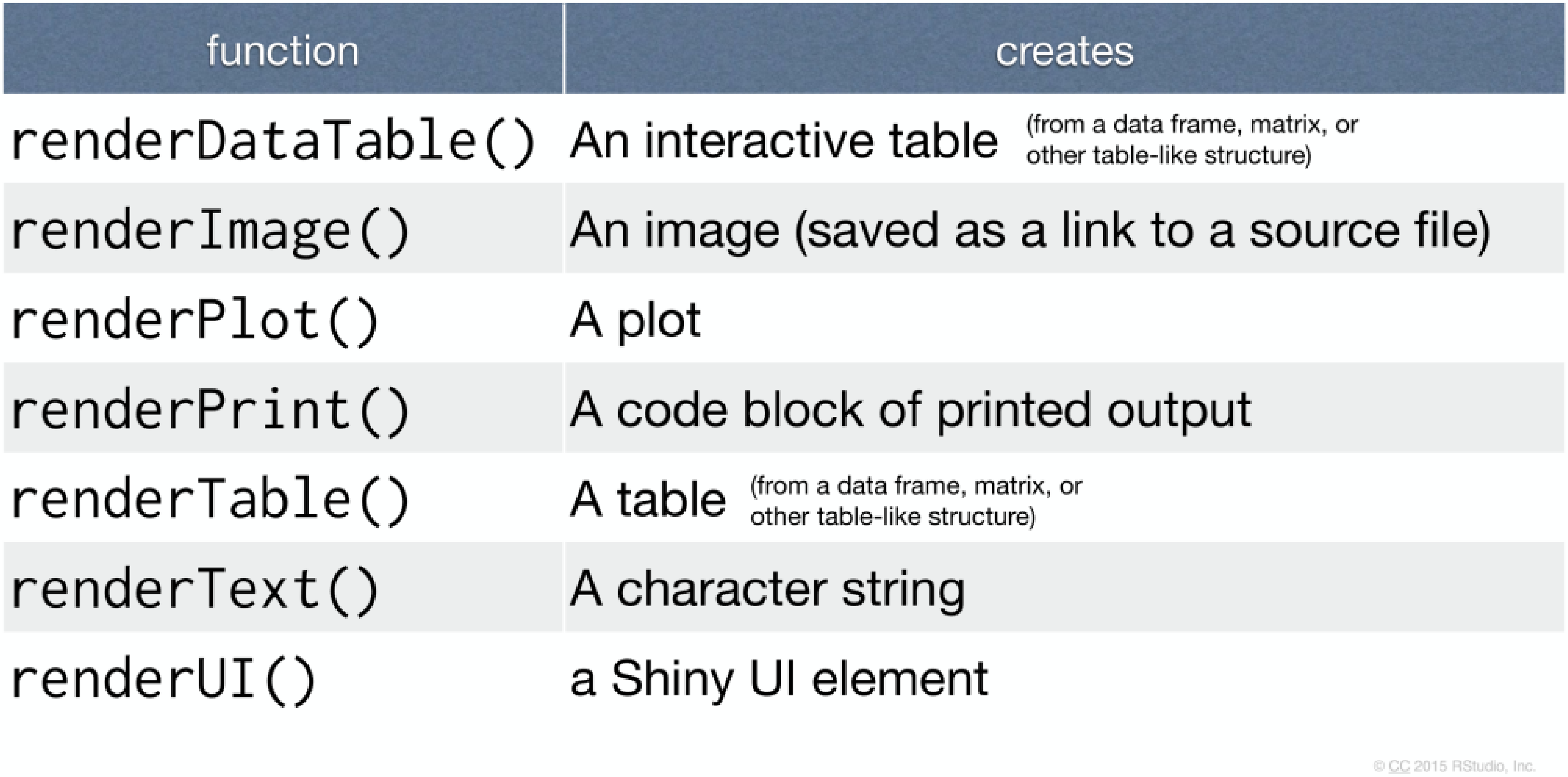
```
ui <- fluidPage(

    titlePanel("Case-Shiller Home Price Indices"),

    sidebarLayout(

        sidebarPanel(

            selectInput(inputId = 'city', label = 'Select a city:', choices = cities,
                        selected = 'Seattle')

        ),

        mainPanel(

            plotOutput(outputId = "tsPlot")

        )
    )
)

server <- function(input, output) {

    output$tsPlot <- renderPlot({

        plot(cs[[input$city]], main = paste('Seasonally Adjusted Case-Shiller Index for',
                                            input$city), ylim = c(0, 300))

    })

}
```

R-Ladies

Input elements

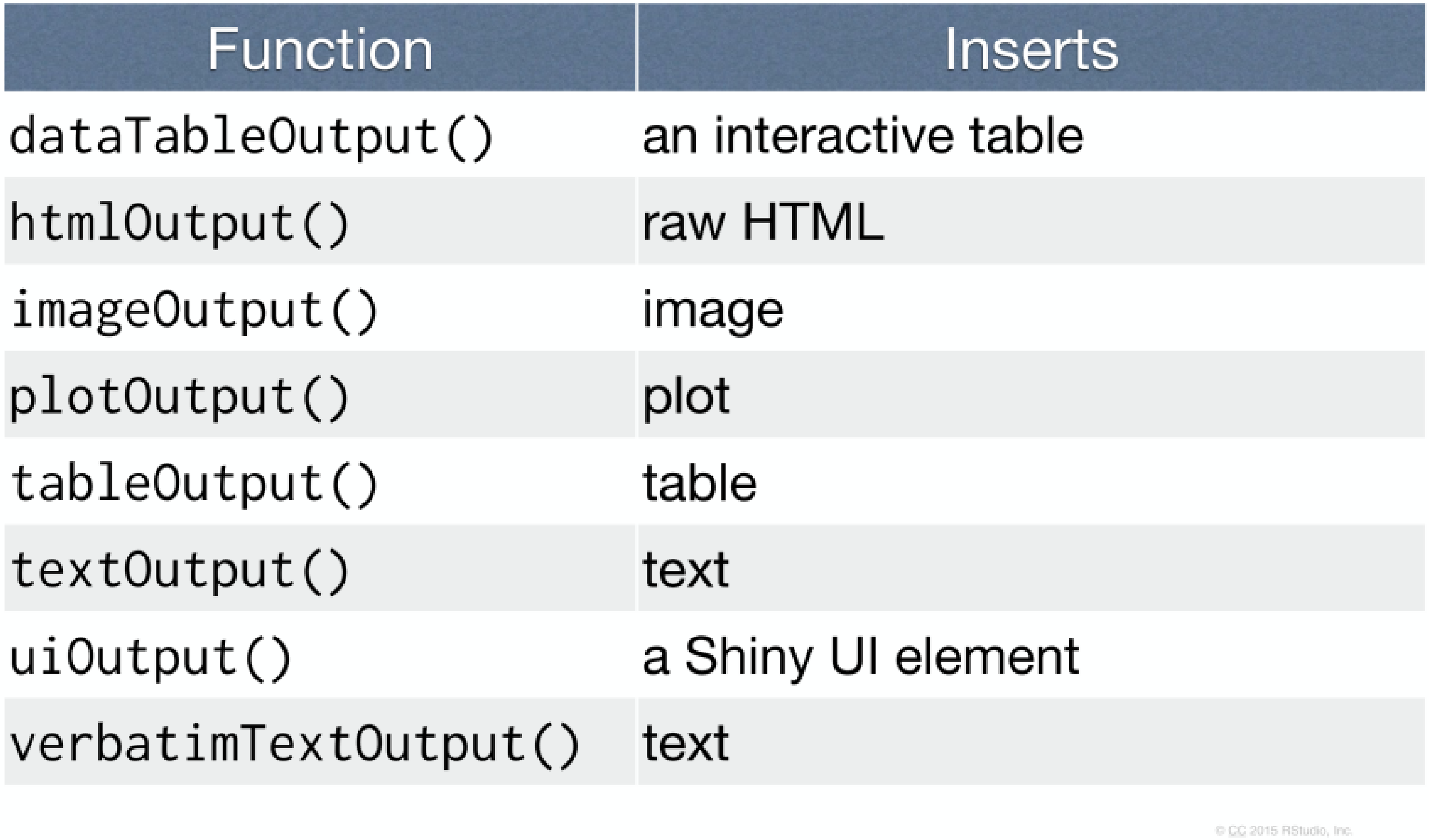


Render functions



function	creates
renderDataTable()	An interactive table (from a data frame, matrix, or other table-like structure)
renderImage()	An image (saved as a link to a source file)
renderPlot()	A plot
renderPrint()	A code block of printed output
renderTable()	A table (from a data frame, matrix, or other table-like structure)
renderText()	A character string
renderUI()	a Shiny UI element

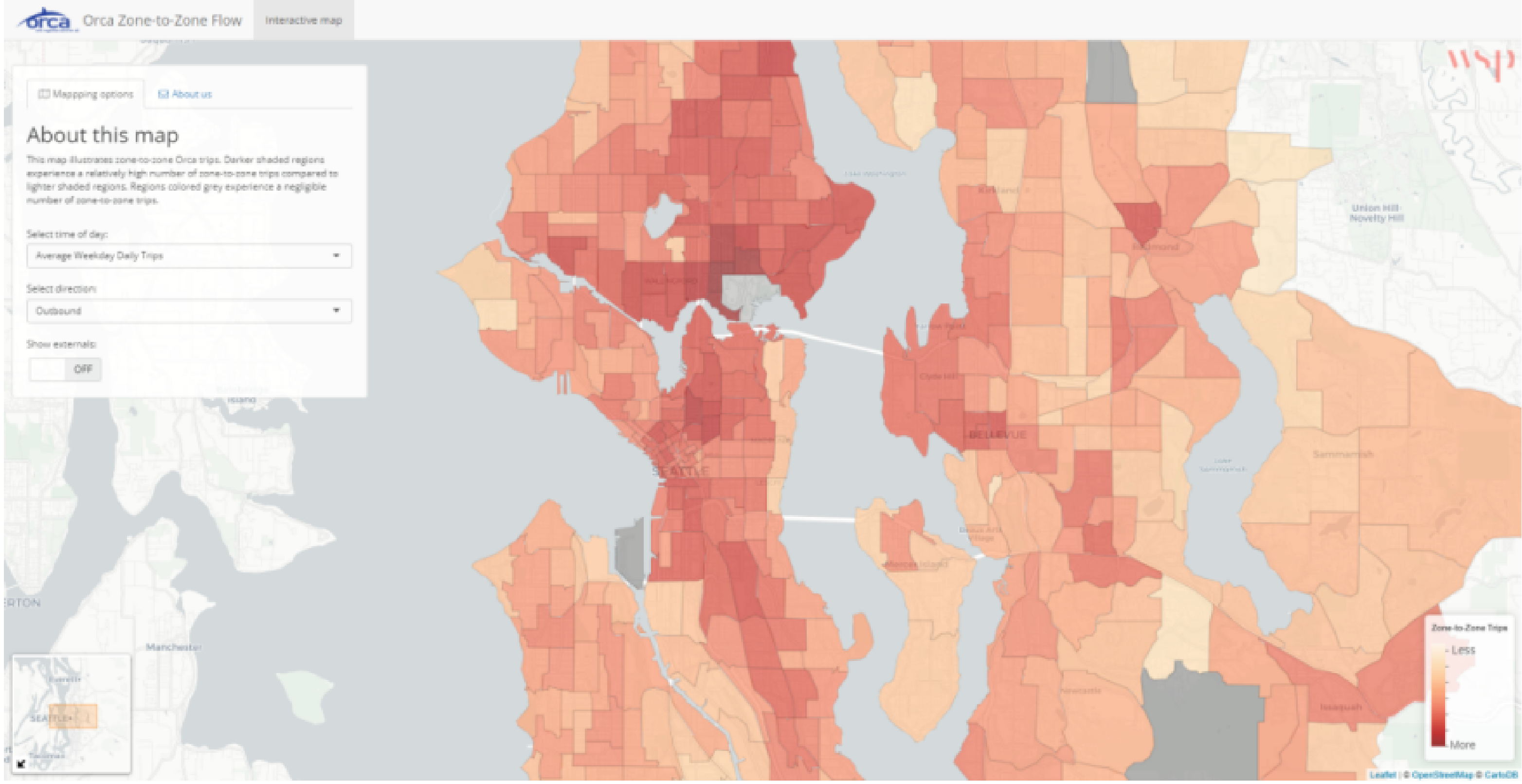
Output functions



Function	Inserts
`dataTableOutput()`	an interactive table
`htmlOutput()`	raw HTML
`imageOutput()`	image
`plotOutput()`	plot
`tableOutput()`	table
`textOutput()`	text
`uiOutput()`	a Shiny UI element
`verbatimTextOutput()`	text

More advanced applications

Shiny + leaflet



About this map

This map illustrates zone-to-zone Orca trips. Darker shaded regions experience a relatively high number of zone-to-zone trips compared to lighter shaded regions. Regions colored grey experience a negligible number of zone-to-zone trips.

Select time of day:

Average Weekday Daily Trips

Select direction:

Outbound

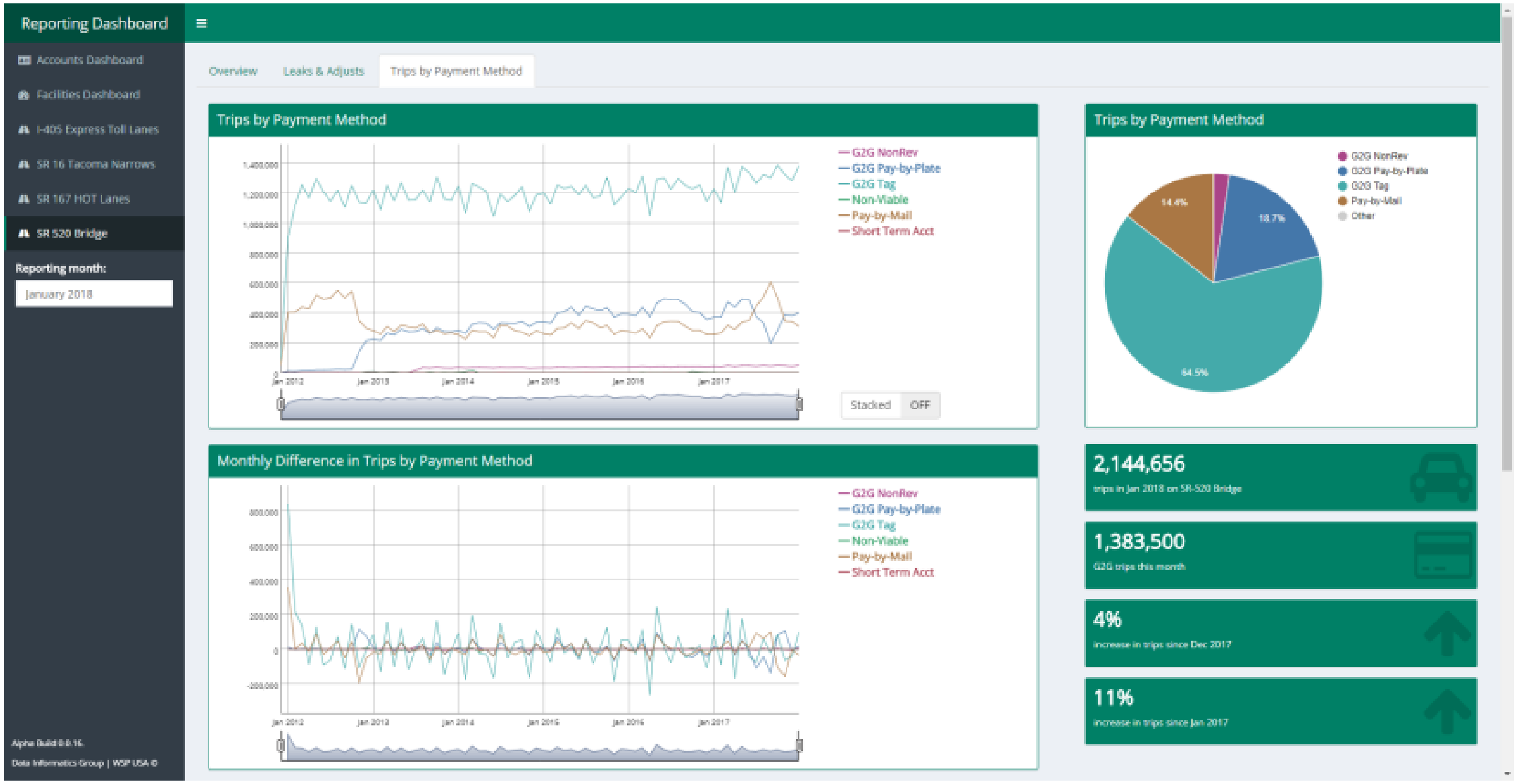
Show externals:

OFF

https://byollin.shinyapps.io/orca/ (https://byollin.shinyapps.io/orca/)

R-Ladies

Shiny + shinydashboard



Reporting month: January 2018

2,144,656
trips in Jan 2018 on SR-520 Bridge

1,383,500
G2G trips this month

4%
increase in trips since Dec 2017

11%
increase in trips since Jan 2017

Alpha Build 0.0.16.
Data Informatics Group | WSP USA ©

http://127.0.0.1:3838/reporting/ (http://127.0.0.1:3838/reporting/)

Wrap up

Add-on packages for Shiny

- `shinyjs` - extends the Shiny package and provides utilities for calling custom JavaScript bindings
- `shinythemes` - adds Bootswatch CSS themes to Shiny applications
- `shinydashboard` - wraps the Admin LTE library for creating dashboard interfaces
- `leaflet` - wraps the leaflet JavaScript library for interactive, mobile-friendly web-mapping
- `DT` - interactive tables
- `dygraphs` - interactive time-series charts
- `googleVis` - interface to the Google Charts API
- `crosstalk` - enables cross-widget interaction
- `plumbr` - create web APIs that call R
- `promises` - enables asynchronous evaluation R code
- Much more! Explore some here: http://gallery.htmlwidgets.org/ (http://gallery.htmlwidgets.org/)

Resources for getting started with Shiny

- https://shiny.rstudio.com/tutorial/ (https://shiny.rstudio.com/tutorial/)
- https://shiny.rstudio.com/articles/ (https://shiny.rstudio.com/articles/)
- https://shiny.rstudio.com/gallery/ (https://shiny.rstudio.com/gallery/)

Questions?