

Graphic Interactivity with HTML Widgets



ZevRoss
Know Your Data

HTML widgets

Use JavaScript libraries from within R to add interactivity to your graphics.

HTML widgets for different output

- Plots
- Maps
- Tables
- Others (heatmaps, 3D scatter plots, diagrams)

HTML Widgets

HTML widgets gallery.

htmlwidgets for R - gallery

Add Your Widget!htmlwidgets.org

71 registered widgets available to explore

Sort

Github stars

▼

Text Filter

search name, author, descr

▼

Author Filter

▼

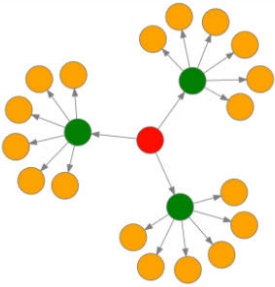
Tag Filter


▼

CRAN Only

☐

Showing 29 of 71



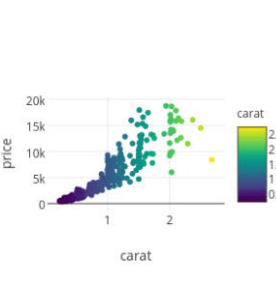
DiagrammeR  499


Easily create graph diagrams using R.

■ **author:** rich-iannone

■ **tags:** visualization, diagram, networks

■ **js libraries:** d3.viz, mermaid



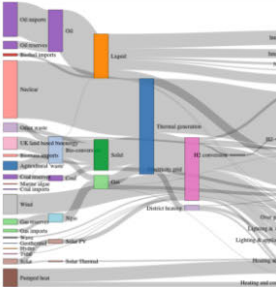
plotly  483


Create interactive web graphics via Plotly's JavaScript graphing library.

■ **author:** cpsievert

■ **tags:** d3, webgl

■ **js libraries:** plotly.js




networkD3  212


A port of Christopher Gandrud's d3Network package to the htmlwidgets framework.

■ **author:** christophergandrud

■ **tags:** visualization, networks

■ **js libraries:** d3



leaflet  203

Leaflet is an open-source JavaScript library for interactive maps. This package makes it easy to create Leaflet maps from R.


■ **author:** rstudio


■ **tags:** visualization, maps

■ **js libraries:** leaflet

id	name	age	grade	test1_score	test2_score
1	Bob	28	C	8.8	9.2
2	Arlley	27	A	9.5	9.1
3	James	26	A	9.6	9.2
4	David	26	C	8.9	9.1

New Haven Temperatures





Show entries

	Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species
1	5.1	3.5	1.4	0.2	setosa
2	4.9	3	1.4	0.2	setosa

Search:

HTML Widgets can easily be included in Shiny apps

- No really, it's pretty easy

We will go through three examples

- `plotly`: easily add interactivity to a ggplot
- `highcharter`: add interactive charts and maps
- `leaflet`: add an interactive map

plotly

Use plotly to add interactivity to a ggplot

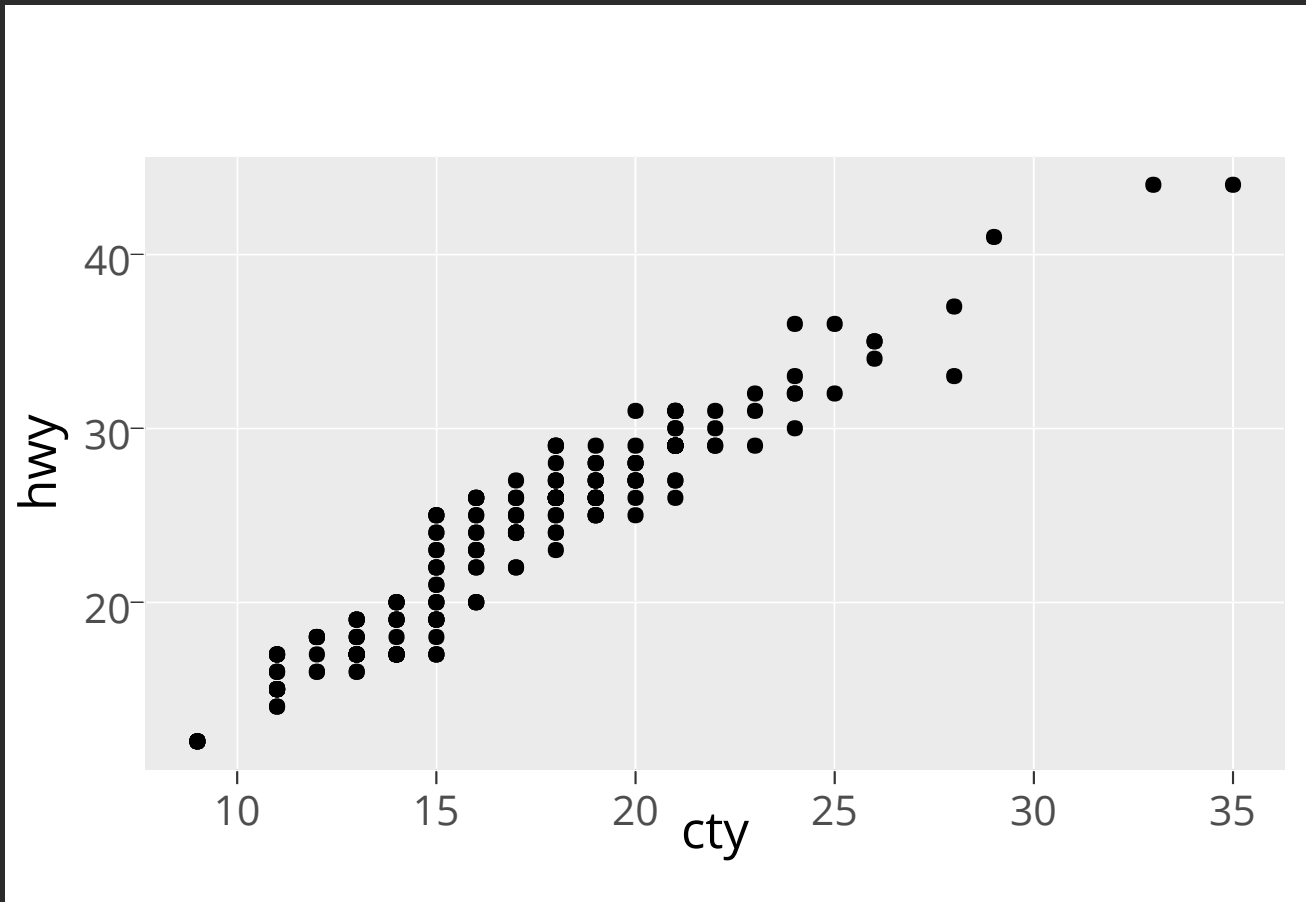
- Package created by Carson Sievert and others
- There is a web site (plot.ly) that allows you to use more than just R (python, MATLAB, Excel)

The R package is designed to "shoehorn" interactivity into a `ggplot`

- Super-easy if you know `ggplot2`
- Plots are often nice enough but not perfect.

To interactify a ggplot outside of Shiny

```
library(plotly)
p <- ggplot(mpg, aes(cty, hwy)) + geom_point()
ggplotly(p)
```



Including an interactive ggplot in Shiny

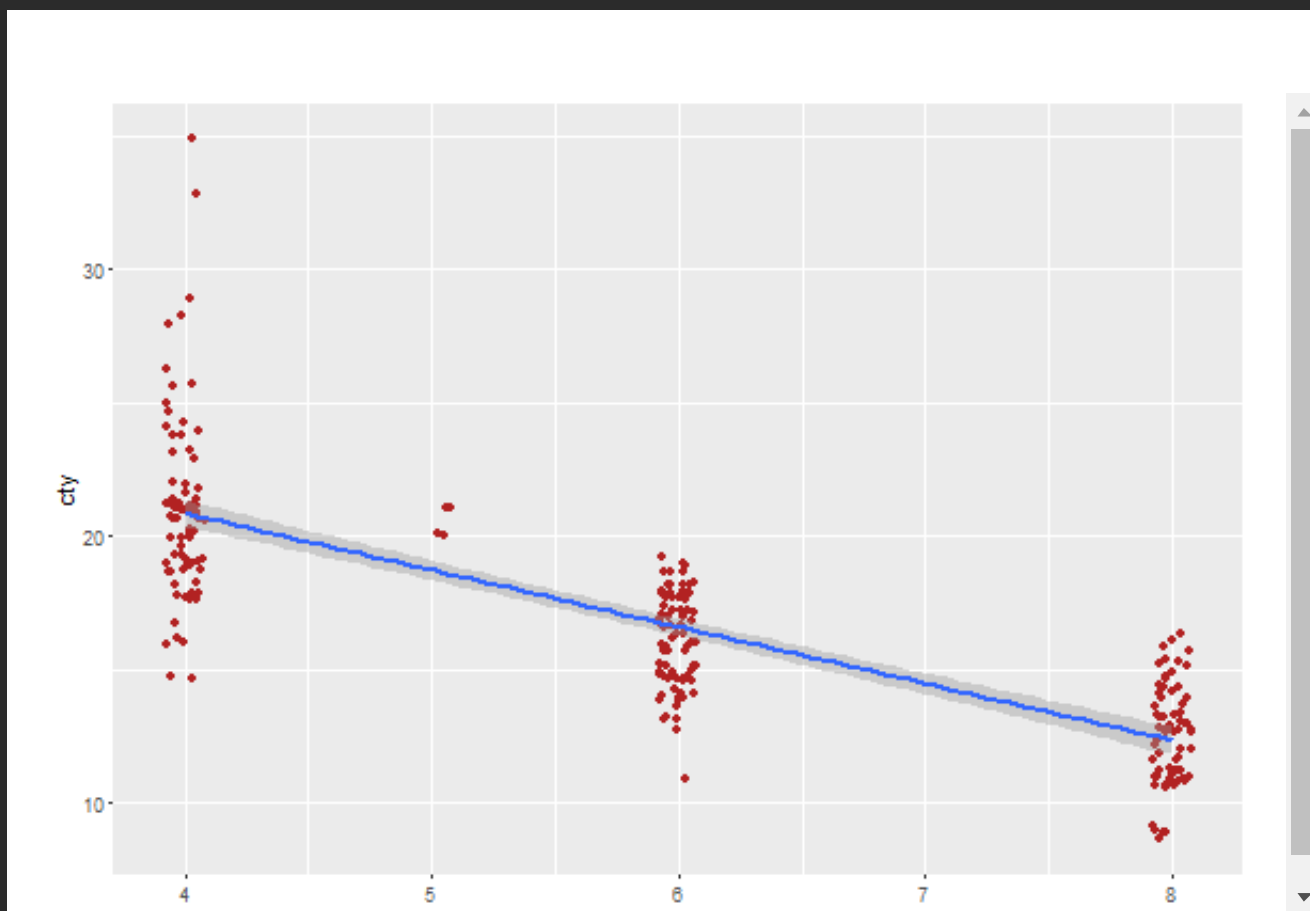
Non-interactive ggplot in Shiny (code)

```
library(ggplot2)
ui <- basicPage(
  plotOutput("myplot")
)
server <- function(input, output, session){

  output$myplot <- renderPlot({
    ggplot(mpg, aes(cyl, cty)) +
      geom_jitter(aes(text = paste("make:", manufacturer)),
width=0.2, color="firebrick") +
      stat_smooth(method="lm")
  })
}

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

Non-interactive ggplot in Shiny (app)



Taking bets, how many lines of code to make plot interactive?

Just add "ly" in two places

Just add "ly" in two places

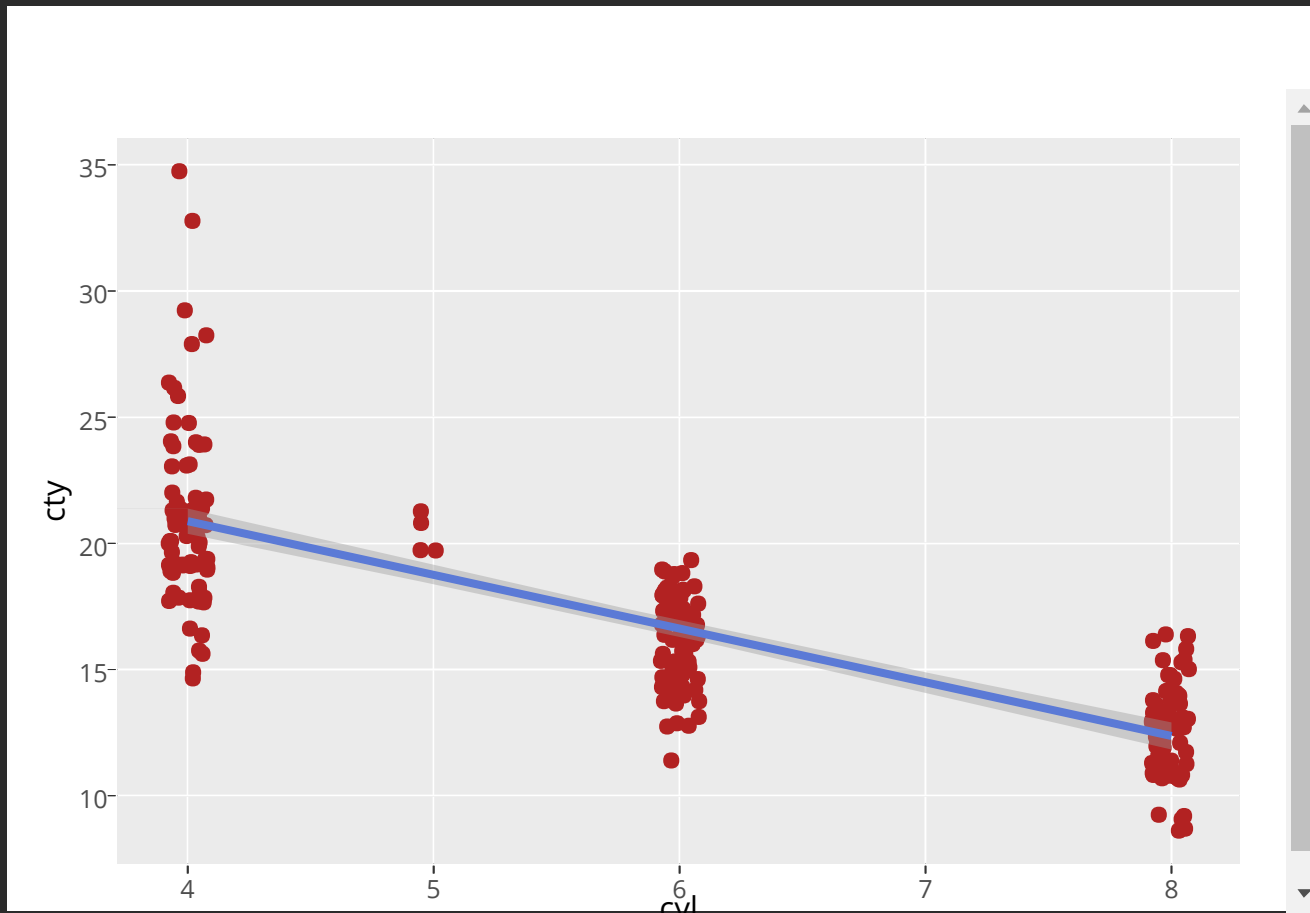
```
library(plotly)
ui <- basicPage(
  plotlyOutput("myplot") # add "ly"
)
server <- function(input, output, session){

  output$myplot <- renderPlotly({ # add "ly"
    ggplot(mpg, aes(cyl, cty)) +
      geom_jitter(aes(text = paste("make:", manufacturer)),
width=0.2, color="firebrick") +
      stat_smooth(method="lm")
  })
}

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

Just add "ly" in two places

Not perfect. But often easy and pretty good beats hard and perfect.



highcharter

highcharter

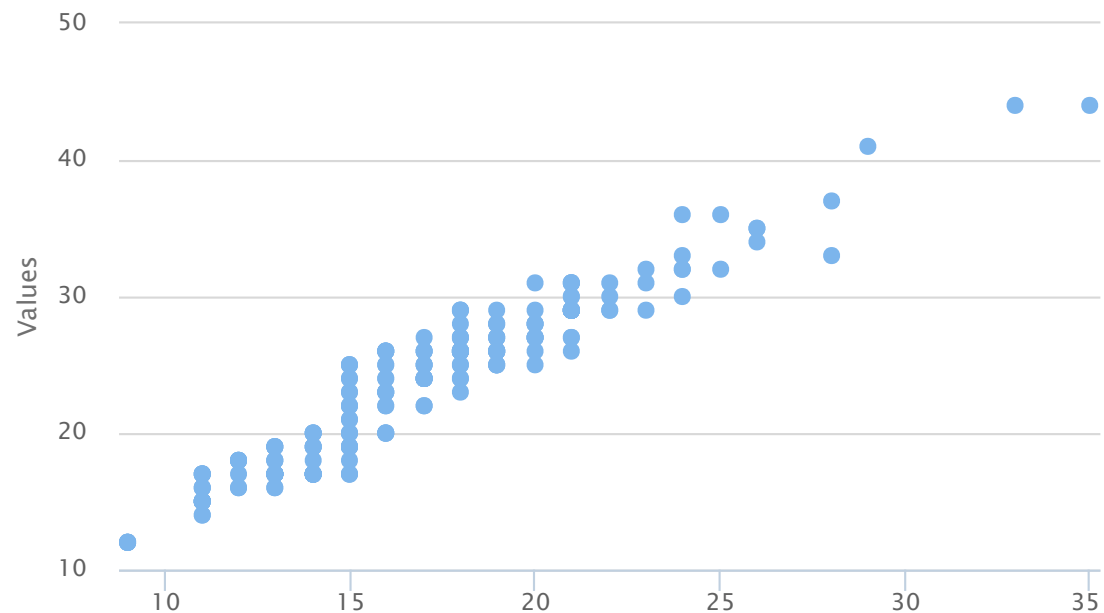
- Created by Joshua Kunst
- Makes use of the highcharts JS library
- Can make plots or maps

**highcharter generally makes
nicer plots than plotly**

But they can be harder to put together. Good detail at the [highcharter site](#).

Using highcharter can be fairly simple

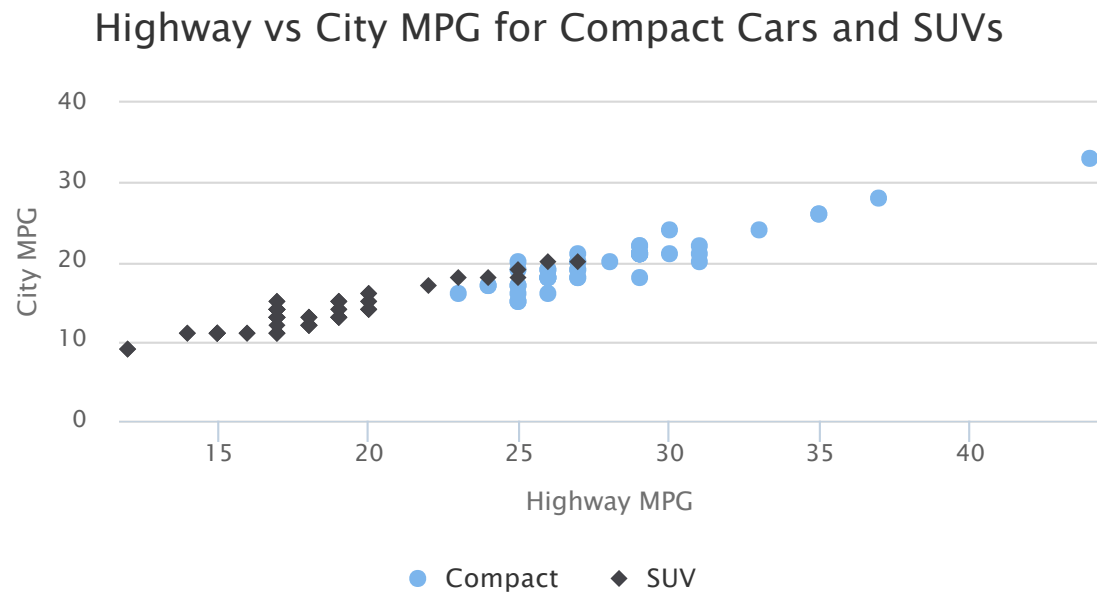
```
library(highcharter)
highchart() %>%
  hc_add_series_scatter(x=mpg$cty, mpg$hwy)
```



Using highcharter gets complex fast

```
library(highcharter)
compact <- mpg[mpg$class == "compact",]
suv <- mpg[mpg$class == "suv",]
highchart() %>%
  hc_add_series_scatter(name = "Compact", x = compact$hwy,
y= compact$cty, showInLegend = TRUE) %>%
  hc_add_series_scatter(name = "SUV", x = suv$hwy, y=
suv$cty, showInLegend = TRUE) %>%
  hc_tooltip(headerFormat = "{series.name}
",
              pointFormat = "{point.x} (Highway MPG)
{point.y} (City MPG)") %>%
  hc_title(text = "Highway vs City MPG for Compact Cars
and SUVs") %>%
  hc_xAxis(title = list(text = "Highway MPG")) %>%
  hc_yAxis(title = list(text = "City MPG"))
```


But highcharter makes nice and useful interactive graphics!



highcharter is Shiny-ready

- `renderHighchart()` for the server
- `highchartOutput()` for the UI

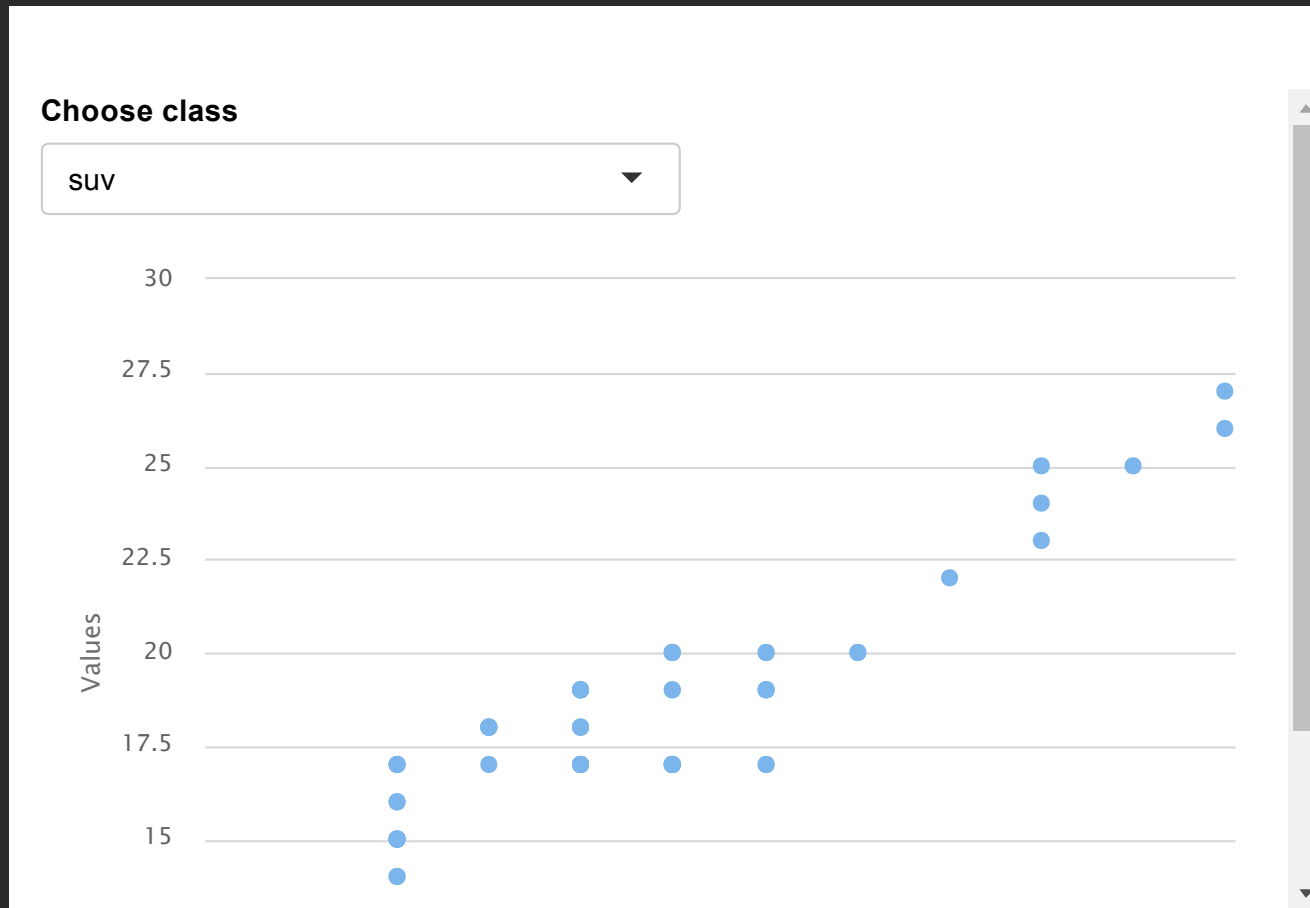
A highchart in Shiny (code)

```
library(highcharter)
library(ggplot2) # needed for mpg dataset only
ui <- basicPage(
  selectInput("class", "Choose class", c("suv",
    "compact"), selected = "suv"),
  highchartOutput("myplot") # add highchart
)
server <- function(input, output, session){

  dat <- reactive({mpg[mpg$class == input$class,]})
  output$myplot <- renderHighchart({ # add highchart
    dat <- dat()
    highchart() %>%
      hc_add_series_scatter(x=dat$cty, dat$hwy)
  })
}

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

A highchart in Shiny (app)



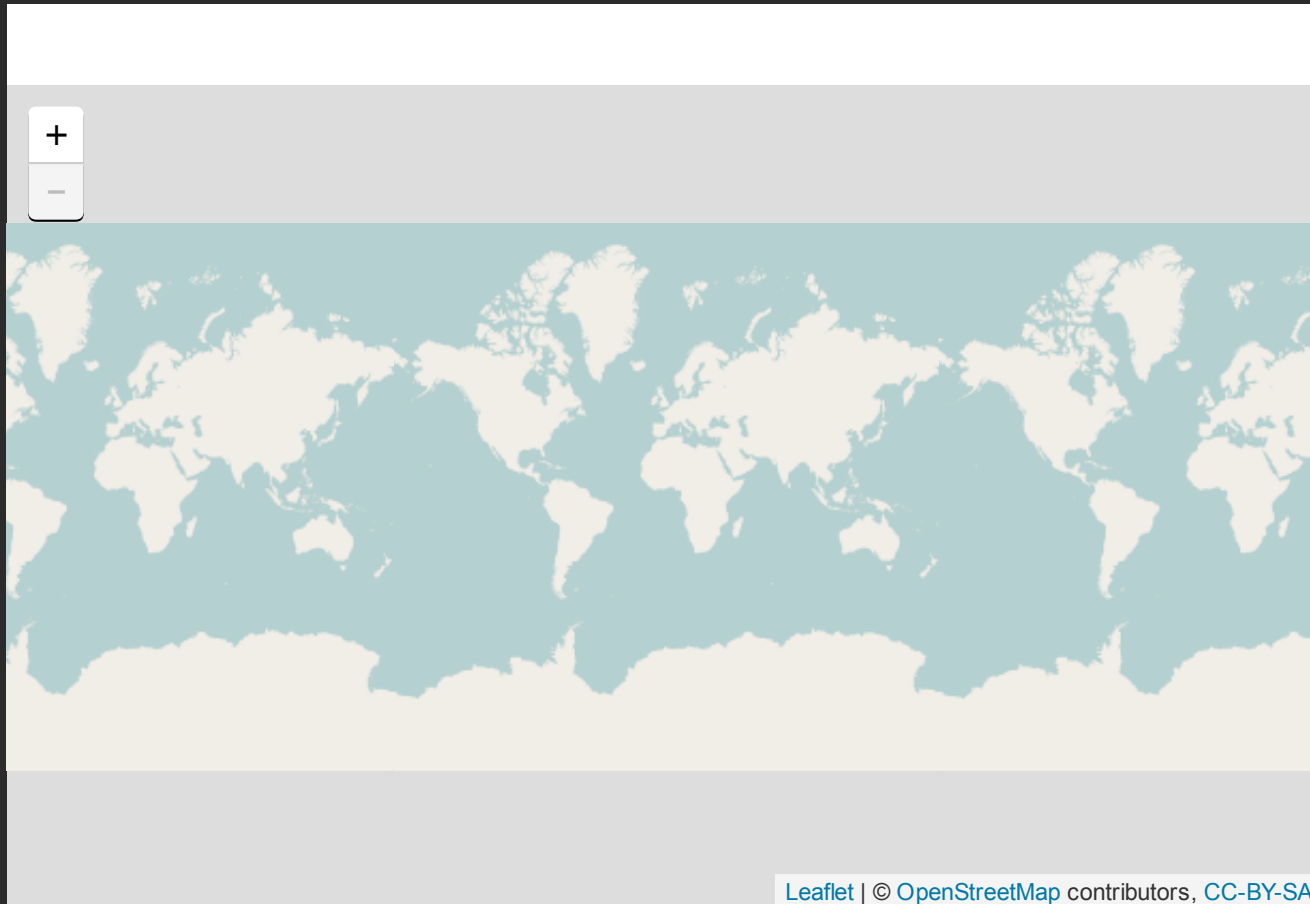
leaflet

leaflet

- Created by the RStudio team
- Uses the popular leaflet JS package
- Can include points, lines or polygons
- Can include baselayers

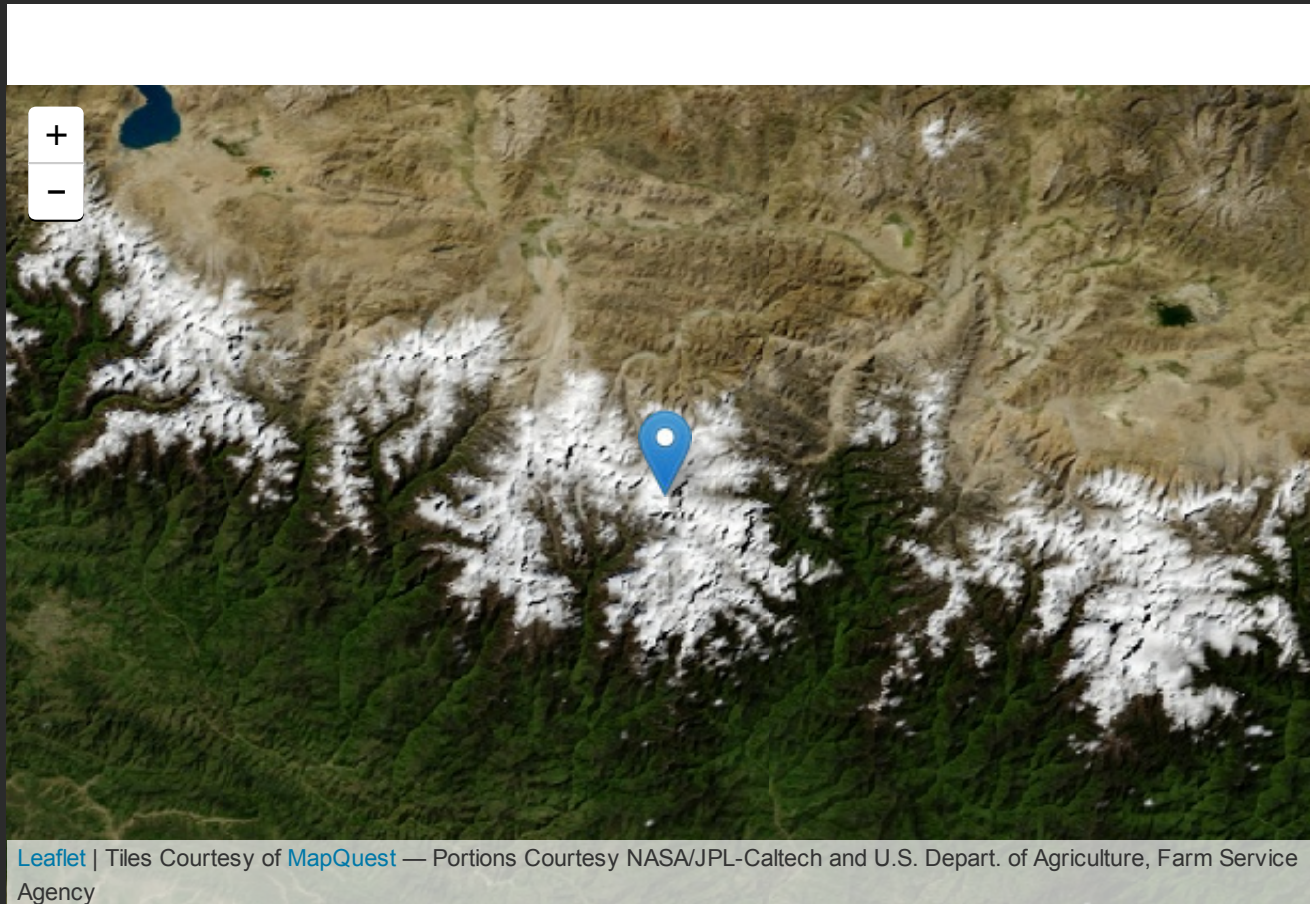
A simple leaflet map

```
library(leaflet)  
leaflet() %>% addTiles()
```



A little more complex

```
leaflet() %>% addProviderTiles("MapQuestOpen.Aerial") %>%  
setView(86.92, 27.99, zoom=8) %>% addMarkers(86.92, 27.99)
```



Including a leaflet map in Shiny

- `renderLeaflet()` for the server
- `leafletOutput()` for the UI

The user interface

```
library(leaflet)
ui <- basicPage(
  sliderInput("magnitude", "Magnitude:", min = 4, max =
6.4,
              value = c(5, 6), step=0.2),
  leafletOutput("mymap") # add leaflet
)
```

The server

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

  dat <- reactive({
    minmag <- input$magnitude[1]
    maxmag <- input$magnitude[2]
    quakes[quakes$mag>=minmag & quakes$mag<=maxmag,]
  })

  output$mymap <- renderLeaflet({ # add leaflet
    leaflet(data = dat()) %>% addTiles() %>%
      addMarkers(~long, ~lat, popup = ~as.character(mag))
  })
}
```

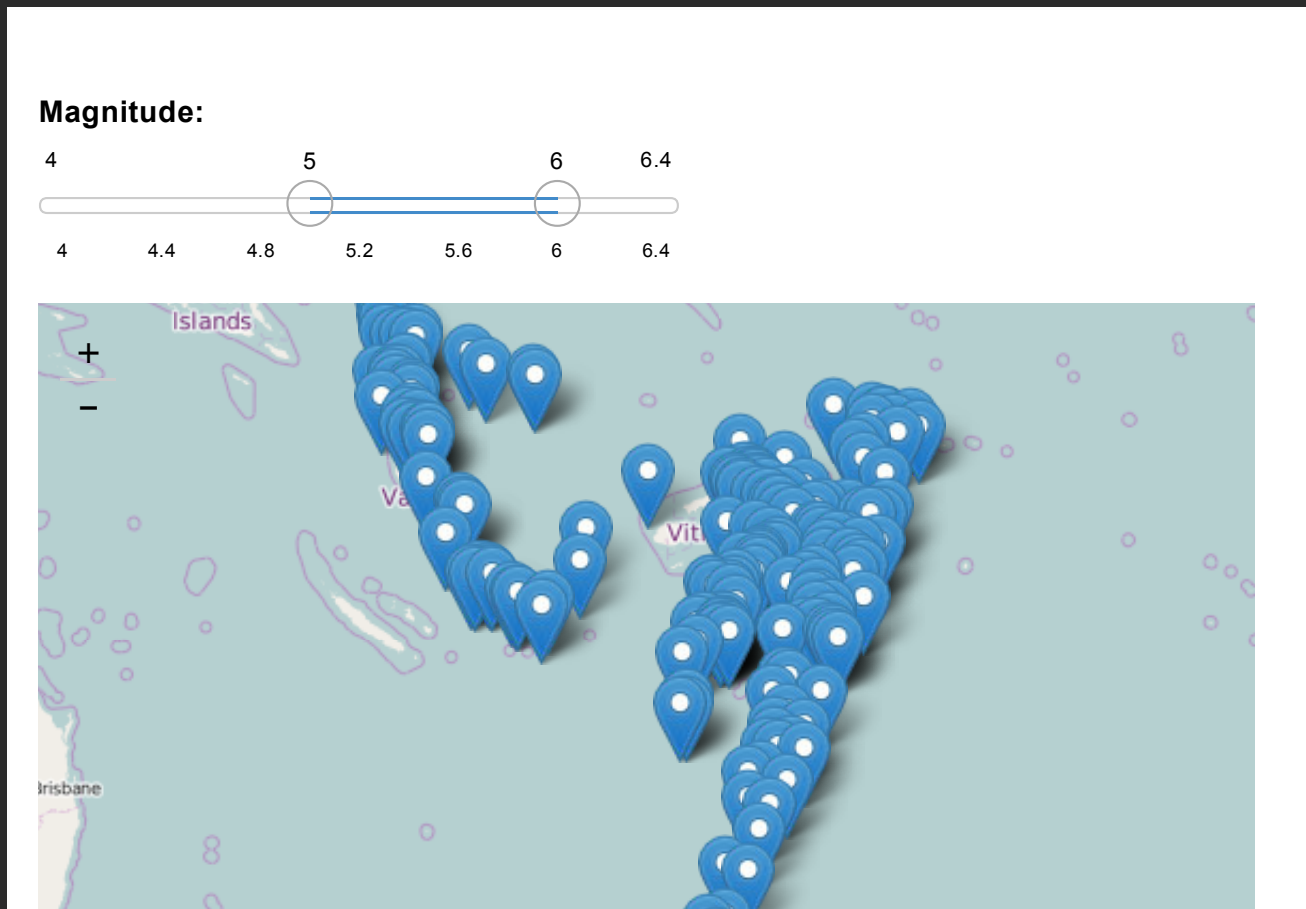
Including a leaflet map in Shiny (code)

```
library(leaflet)
ui <- basicPage(
  sliderInput("magnitude", "Magnitude:",
             min = 4, max = 6.4, value = c(5, 6),
             step=0.2),
  leafletOutput("mymap") # add leaflet
)
server <- function(input, output, session){

  dat <- reactive({
    minmag <- input$magnitude[1]
    maxmag <- input$magnitude[2]
    quakes[quakes$mag>=minmag & quakes$mag<=maxmag,]
  })
  output$mymap <- renderLeaflet({ # add leaflet
    leaflet(data = dat()) %>% addTiles() %>%
      addMarkers(~long, ~lat, popup = ~as.character(mag))
  })
}

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


Including a leaflet map in Shiny (app)



exercise 3 (html widgets)