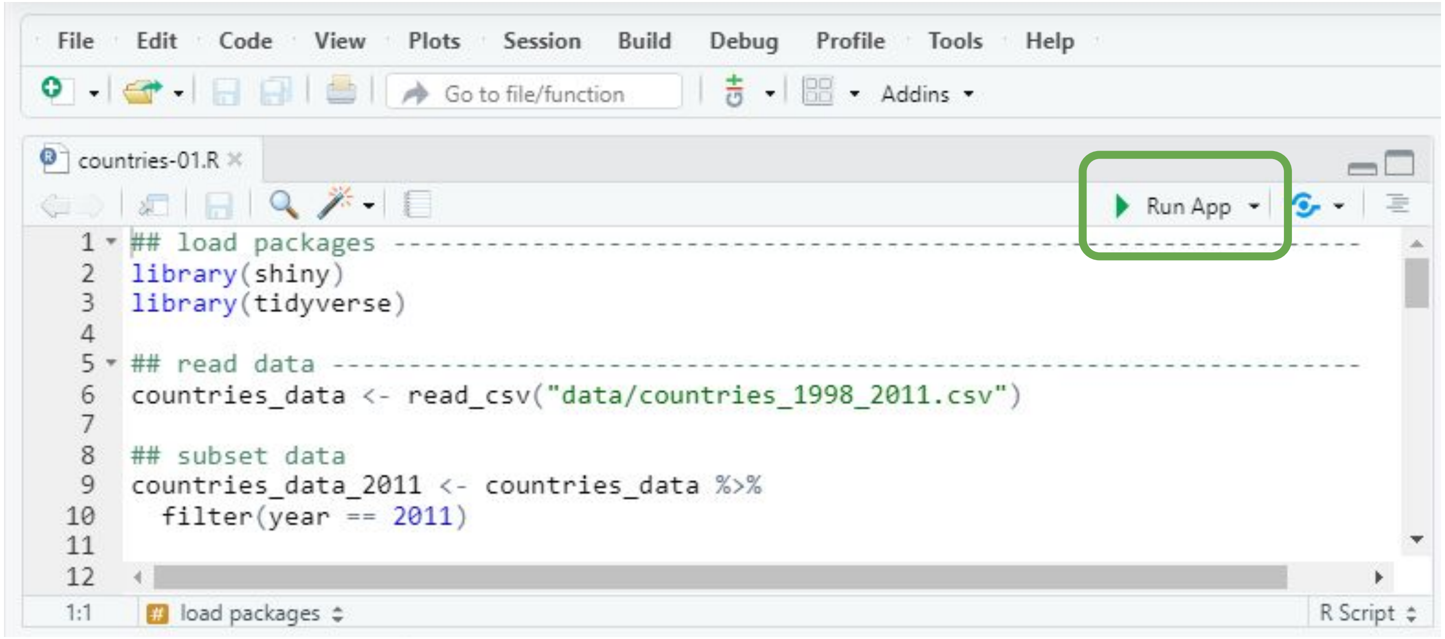
Intro to Shiny

Omayma Said

- Run **countries-01.R**



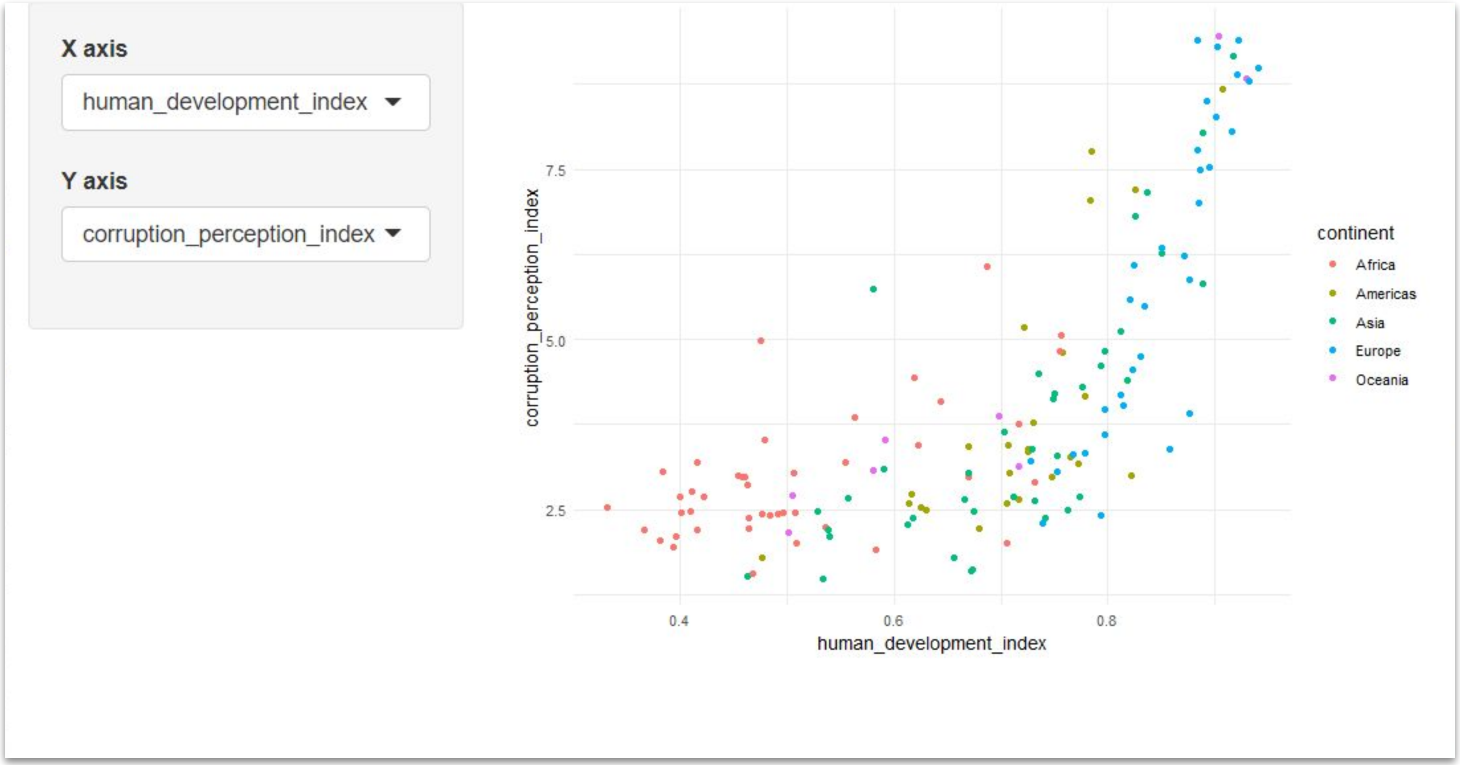
```
## load packages -----------------------------------------------------------
library(shiny)
library(tidyverse)

## read data ---------------------------------------------------------------
countries_data <- read_csv("data/countries_1998_2011.csv")

## subset data
countries_data_2011 <- countries_data %>%
  filter(year == 2011)


```

countries-01.R



X axis

human_development_index ▾

Y axis

corruption_perception_index ▾

SHINY APP TEMPLATE

SHINY APP TEMPLATE

```
library(shiny)

ui <- fluidPage()

server <- function(input, output) {}

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

USER INTERFACE

USER INTERFACE

```r
ui <- fluidPage(

  sidebarLayout(
    ## define inputs in sidebar
    sidebarPanel(
      ## select variable for scatter plot x-axis
      selectInput(inputId = "x_axis", label = "X axis",
                  choices = c("human_development_index", "corruption_perception_index",
                              "population", "life_exp", "gdp_per_capita"),
                  selected = "human_development_index"),

      ## select variable for scatter plot y-axis
      selectInput(inputId = "y_axis", label = "Y axis",
                  choices = c("human_development_index", "corruption_perception_index",
                              "population", "life_exp", "gdp_per_capita"),
                  selected = "corruption_perception_index")
    ),

    ## Show output in main panel
    mainPanel(
      plotOutput(outputId = "countries_scatter")
    )
  )
)
```

USER INTERFACE

```r
ui <- fluidPage(

  sidebarLayout(
    ## define inputs in sidebar
    sidebarPanel(
      ## select variable for scatter plot x-axis
      selectInput(inputId = "x_axis", label = "X axis",
                  choices = c("human_development_index", "corruption_perception_index",
                              "population", "life_exp", "gdp_per_capita"),
                  selected = "human_development_index"),

      ## select variable for scatter plot y-axis
      selectInput(inputId = "y_axis", label = "Y axis",
                  choices = c("human_development_index", "corruption_perception_index",
                              "population", "life_exp", "gdp_per_capita"),
                  selected = "corruption_perception_index")
    ),

    ## Show output in main panel
    mainPanel(
      plotOutput(outputId = "countries_scatter")
    )
  )
)
```

USER INTERFACE

```r
ui <- fluidPage(

 sidebarLayout(
  ## define inputs in sidebar
  sidebarPanel(
    ## select variable for scatter plot x-axis
    selectInput(inputId = "x_axis", label = "X axis",
                choices = c("human_development_index", "corruption_perception_index",
                            "population", "life_exp", "gdp_per_capita"),
                selected = "human_development_index"),

    ## select variable for scatter plot y-axis
    selectInput(inputId = "y_axis", label = "Y axis",
                choices = c("human_development_index", "corruption_perception_index",
                            "population", "life_exp", "gdp_per_capita"),
                selected = "corruption_perception_index")
  ),

  ## Show output in main panel
  mainPanel(
    plotOutput(outputId = "countries_scatter")
  )
 )
)
```

USER INTERFACE

```r
ui <- fluidPage(

  sidebarLayout(
    ## define inputs in sidebar
    sidebarPanel(
      ## select variable for scatter plot x-axis
      selectInput(inputId = "x_axis", label = "X axis",
                  choices = c("human_development_index", "corruption_perception_index",
                              "population", "life_exp", "gdp_per_capita"),
                  selected = "human_development_index"),

      ## select variable for scatter plot y-axis
      selectInput(inputId = "y_axis", label = "Y axis",
                  choices = c("human_development_index", "corruption_perception_index",
                              "population", "life_exp", "gdp_per_capita"),
                  selected = "corruption_perception_index")
    ),

    ## Show output in main panel
    mainPanel(
      plotOutput(outputId = "countries_scatter")
    )
  )
)
```

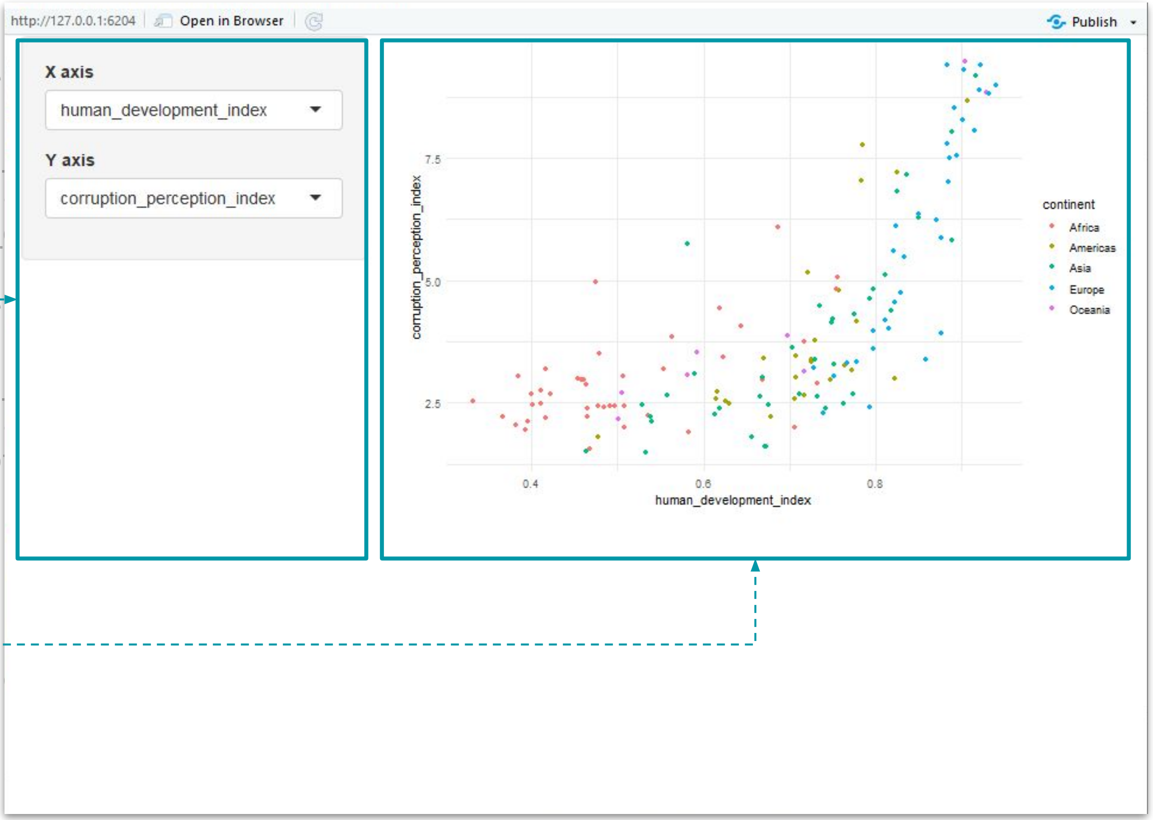
USER INTERFACE

```
ui <- fluidPage(

  sidebarLayout(
    ## define inputs in sidebar
    sidebarPanel(
      ## select variable for scatter
      selectInput(inputId = "x_axis"
                  choices = c("human
                              "popul
                  selected = "human_

      ## select variable for scatter
      selectInput(inputId = "y_axis"
                  choices = c("human
                              "popul
                  selected = "corrup
    ),

    ## Show output in main panel
    mainPanel(
      plotOutput(outputId = "countri
    )
  )
)
```


http://127.0.0.1:6204 Open in Browser Publish

X axis

human_development_index

Y axis

corruption_perception_index

USER INTERFACE

```r
ui <- fluidPage(

  sidebarLayout(
    ## define inputs in sidebar
    sidebarPanel(
      ## select variable for scatter plot x-axis
      selectInput(inputId = "x_axis", label = "X axis",
                  choices = c("human_development_index", "corruption_perception_index",
                              "population", "life_exp", "gdp_per_capita"),
                  selected = "human_development_index"),

      ## select variable for scatter plot y-axis
      selectInput(inputId = "y_axis", label = "Y axis",
                  choices = c("human_development_index", "corruption_perception_index",
                              "population", "life_exp", "gdp_per_capita"),
                  selected = "corruption_perception_index")
    ),

    ## Show output in main panel
    mainPanel(
      plotOutput(outputId = "countries_scatter")
    )
  )
)
```

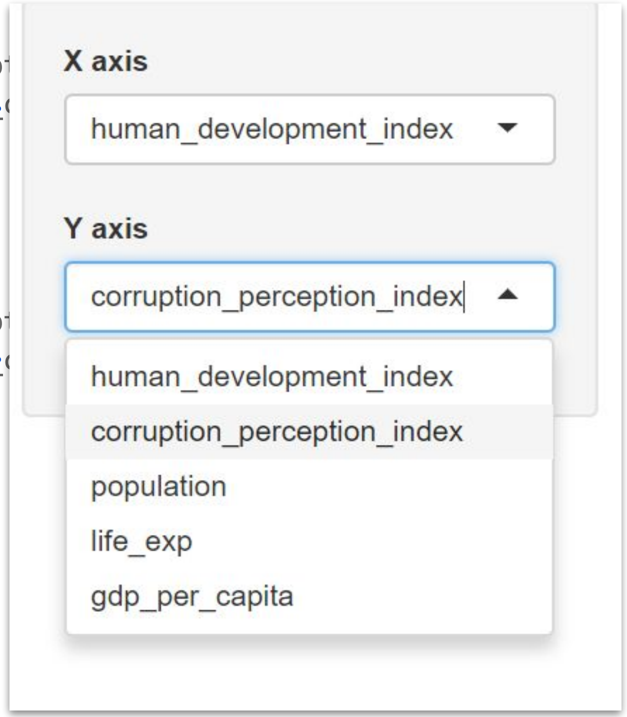
USER INTERFACE

```
ui <- fluidPage(

  sidebarLayout(
    ## define inputs in sidebar
    sidebarPanel(
      ## select variable for scatter plot x-axis
      selectInput(inputId = "x_axis", label = "X axis",
                  choices = c("human_development_index", "corrupt
                              "population", "life_exp", "gdp_per_c
                  selected = "human_development_index"),

      ## select variable for scatter plot y-axis
      selectInput(inputId = "y_axis", label = "Y axis",
                  choices = c("human_development_index", "corrupt
                              "population", "life_exp", "gdp_per_c
                  selected = "corruption_perception_index")
    ),

    ## Show output in main panel
    mainPanel(
      plotOutput(outputId = "countries_scatter")
    )
  )
)
```



X axis

human_development_index

Y axis

corruption_perception_index

USER INTERFACE

```
ui <- fluidPage(

  sidebarLayout(
    ## define inputs in sidebar
    sidebarPanel(
      ## select variable for scatter plot x-axis
      selectInput(inputId = "x_axis", label = "X axis",
                  choices = c("human_development_index", "corruption_perception_index",
                            "population", "life_exp", "gdp_per_capita"),
                  selected = "human_development_index"),

      ## select variable for scatter plot y-axis
      selectInput(inputId = "y_axis", label = "Y axis",
                  choices = c("human_development_index", "corruption_perception_index",
                            "population", "life_exp", "gdp_per_capita"),
                  selected = "corruption_perception_index")
    ),

    ## Show output in main panel
    mainPanel(
      plotOutput(outputId = "countries_scatter")
    )
  )
)
```

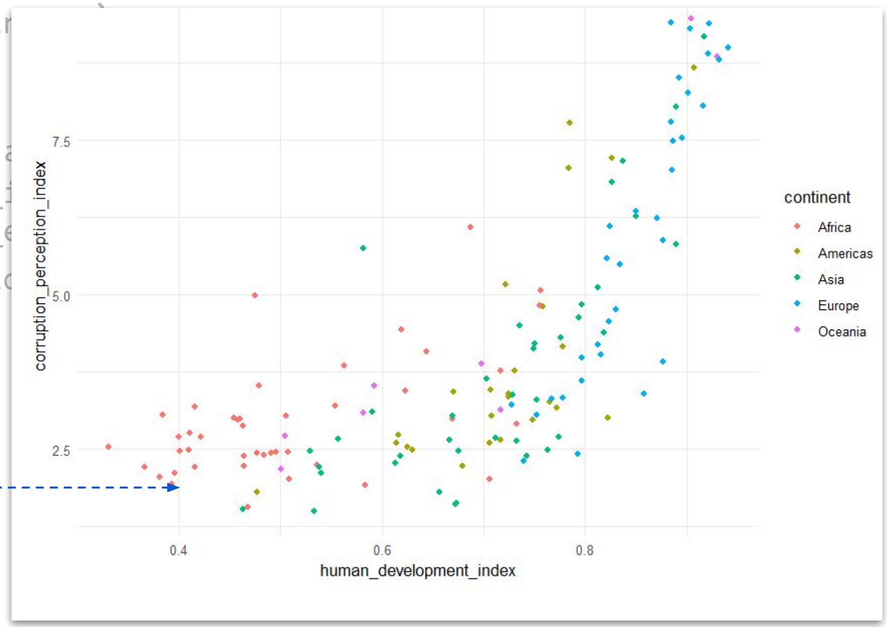
USER INTERFACE

```
ui <- fluidPage(

  sidebarLayout(
    ## define inputs in sidebar
    sidebarPanel(
      ## select variable for scatter plot x-axis
      selectInput(inputId = "x_axis", label = "X axis",
                  choices = c("human_development_index", "corruption_perception_index",
                              "population", "life_exp", "gdp_per_capita"),
                  selected = "human_development_i

      ## select variable for scatter plot y-axis
      selectInput(inputId = "y_axis", label = "Y a
                  choices = c("human_development_i
                              "population", "life_e
                  selected = "corruption_percepti
    ),

    ## Show output in main panel
    mainPanel(
      plotOutput(outputId = "countries_scatter")
    )
  )
)
```



SERVER

SERVER

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

  ## create scatter plot
  output$countries_scatter <- renderPlot({
    ggplot(data = countries_data_2011,
           aes_string(x = input$x_axis, y = input$y_axis,
                      color = "continent"))+
      geom_point()+
      theme_minimal()
  })
}
```

SERVER

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

  ## create scatter plot
  output$countries_scatter <- renderPlot({
    ggplot(data = countries_data_2011,
           aes_string(x = input$x_axis, y = input$y_axis,
                      color = "continent"))+
      geom_point()+
      theme_minimal()
  })
}
```

SERVER

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

  ## create scatter plot
  output$countries_scatter <- renderPlot({
    ggplot(data = countries_data_2011,
           aes_string(x = input$x_axis, y = input$y_axis,
                      color = "continent"))+
      geom_point()+
      theme_minimal()
  })
}
```

SERVER

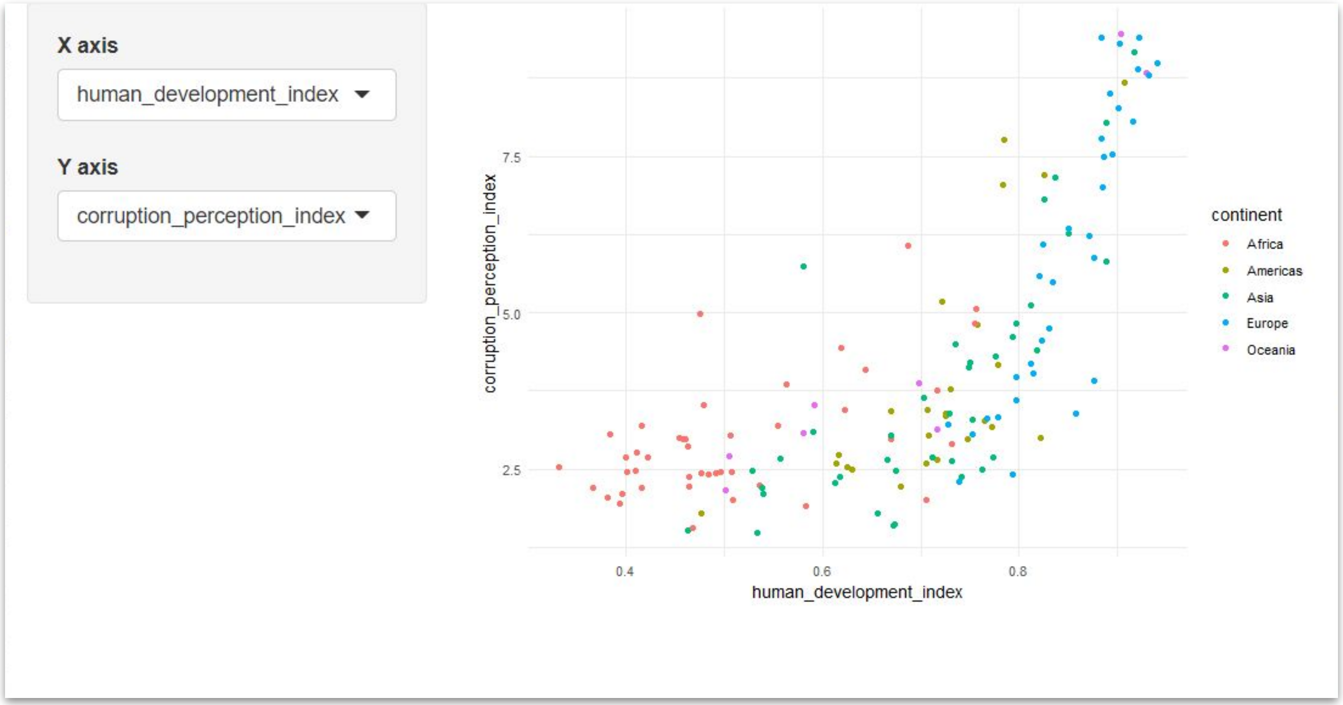
```
server <- function(input, output) {

  ## create scatter plot
  output$countries_scatter <- renderPlot({
    ggplot(data = countries_data_2011,
           aes_string(x = input$x_axis, y = input$y_axis,
                      color = "continent"))+
      geom_point()+
      theme_minimal()
  })
}
```

UI+SERVER

UI+SERVER

```r
# Create the Shiny app object
shinyApp(ui = ui, server = server)
```

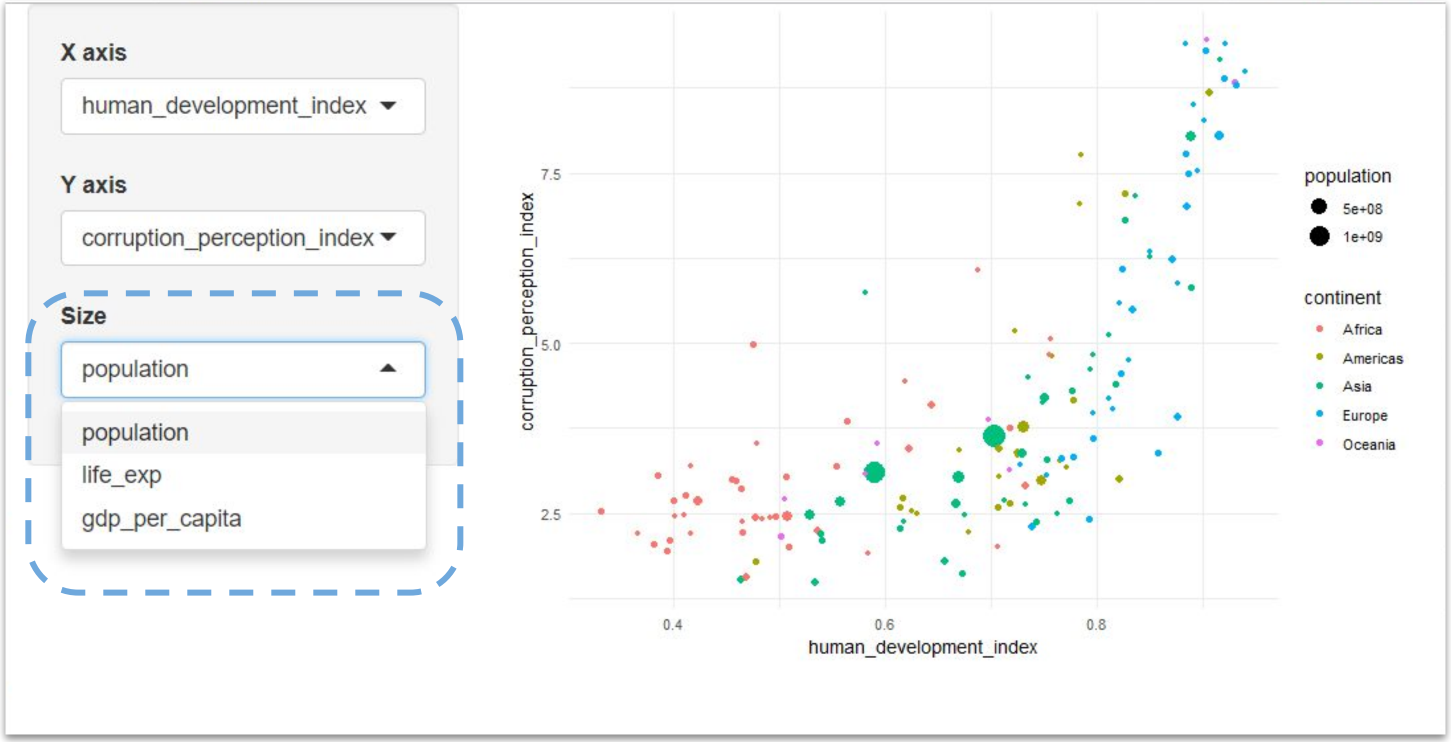

X axis: human_development_index
Y axis: corruption_perception_index

EXERCISE

- Open **countries-02.R**
- Add select variable for point size in the scatter plot with choices "`population`", "`life_exp`", "`gdp_per_capita`".
- Use this variable in the aesthetics of the `ggplot` function as the size argument.

EXERCISE

countries-03.R



X axis

human_development_index

Y axis

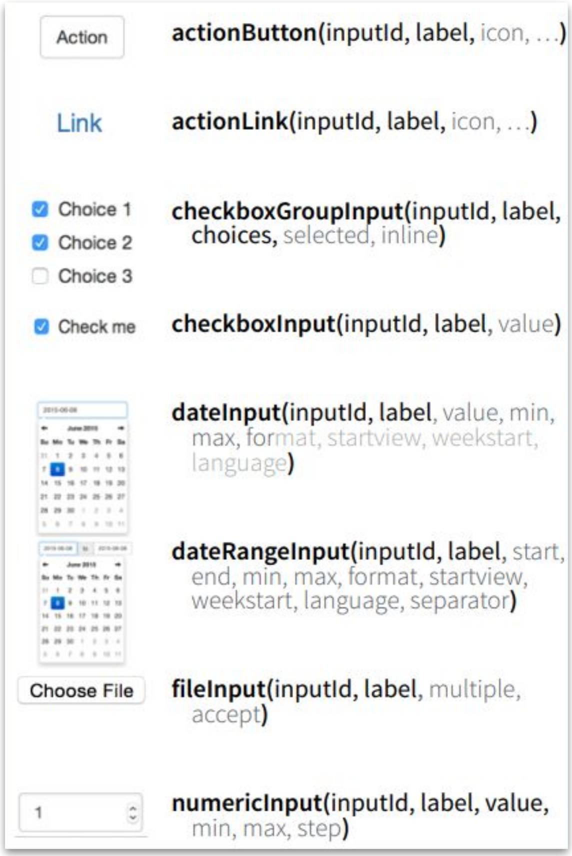
corruption_perception_index

Size

population

population
life_exp
gdp_per_capita

INPUTS



actionButton(inputId, label, icon, …)

actionLink(inputId, label, icon, …)

checkboxGroupInput(inputId, label, choices, selected, inline)

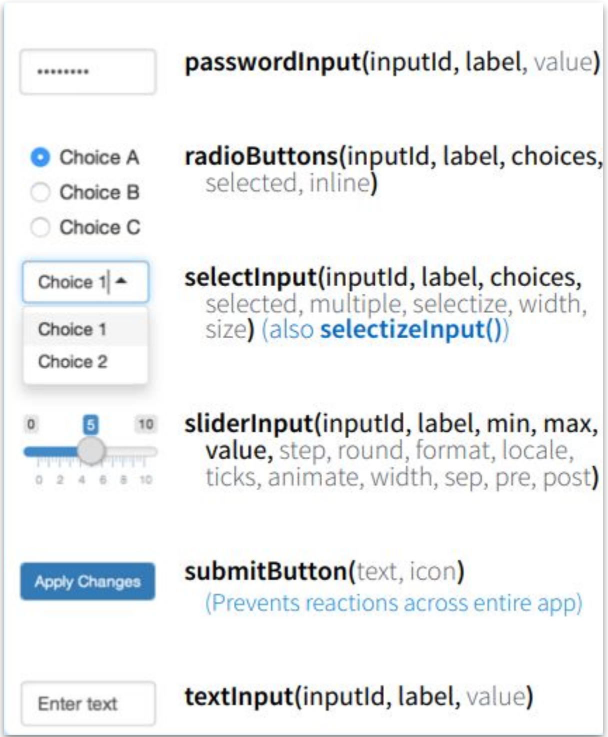
checkboxInput(inputId, label, value)

dateInput(inputId, label, value, min, max, format, startview, weekstart, language)

dateRangeInput(inputId, label, start, end, min, max, format, startview, weekstart, language, separator)

fileInput(inputId, label, multiple, accept)

numericInput(inputId, label, value, min, max, step)



passwordInput(inputId, label, value)

radioButtons(inputId, label, choices, selected, inline)

selectInput(inputId, label, choices, selected, multiple, selectize, width, size) (also **selectizeInput()**)

sliderInput(inputId, label, min, max, value, step, round, format, locale, ticks, animate, width, sep, pre, post)

submitButton(text, icon)
(Prevents reactions across entire app)

textInput(inputId, label, value)

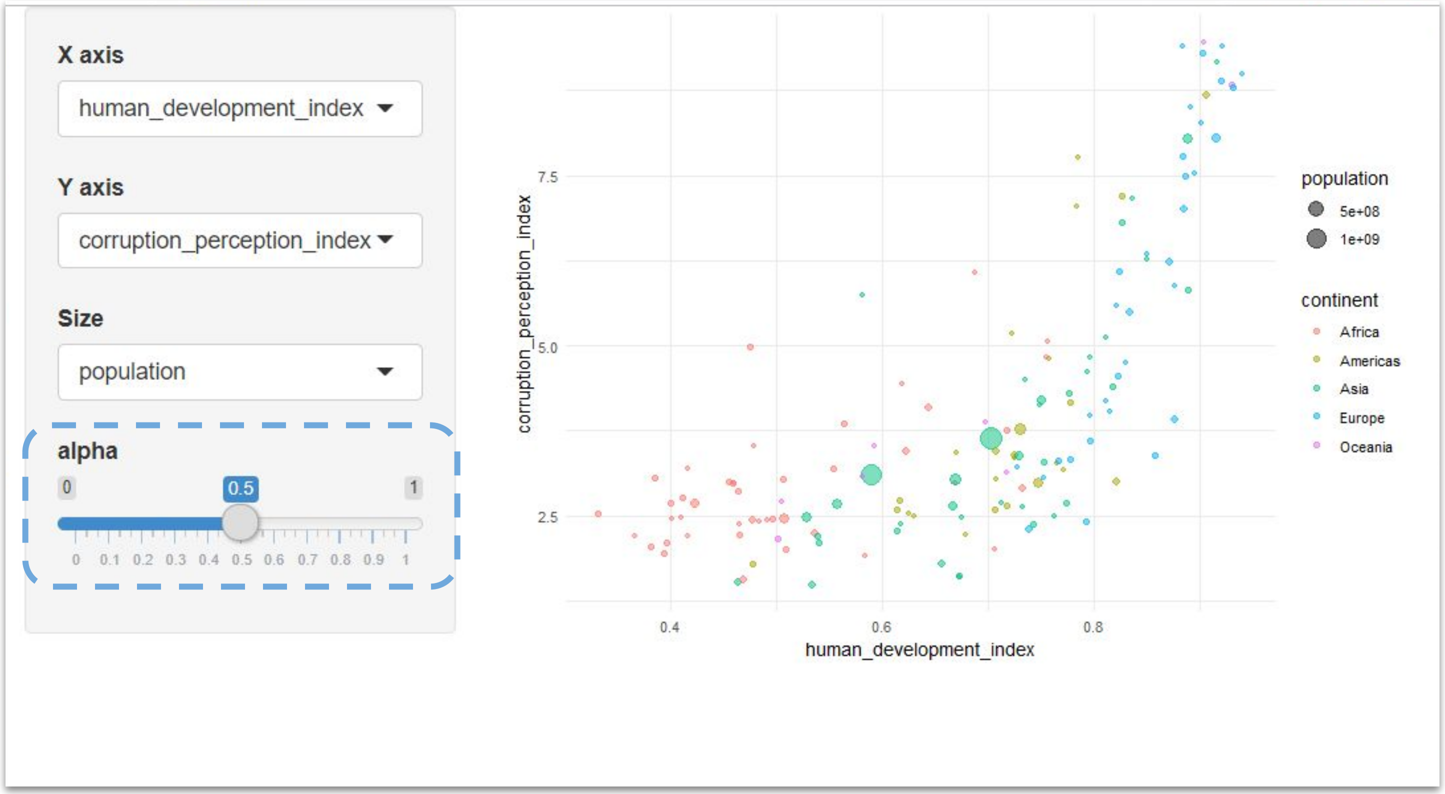
https://shiny.rstudio.com/images/shiny-cheatsheet.pdf

EXERCISE

- Open **countries-03.R**
- Add a `SliderInput` with range [0-1].
- Pass this variable to the `alpha` argument in the `geom_point` function.

countries-04.R

EXERCISE



X axis

human_development_index

Y axis

corruption_perception_index

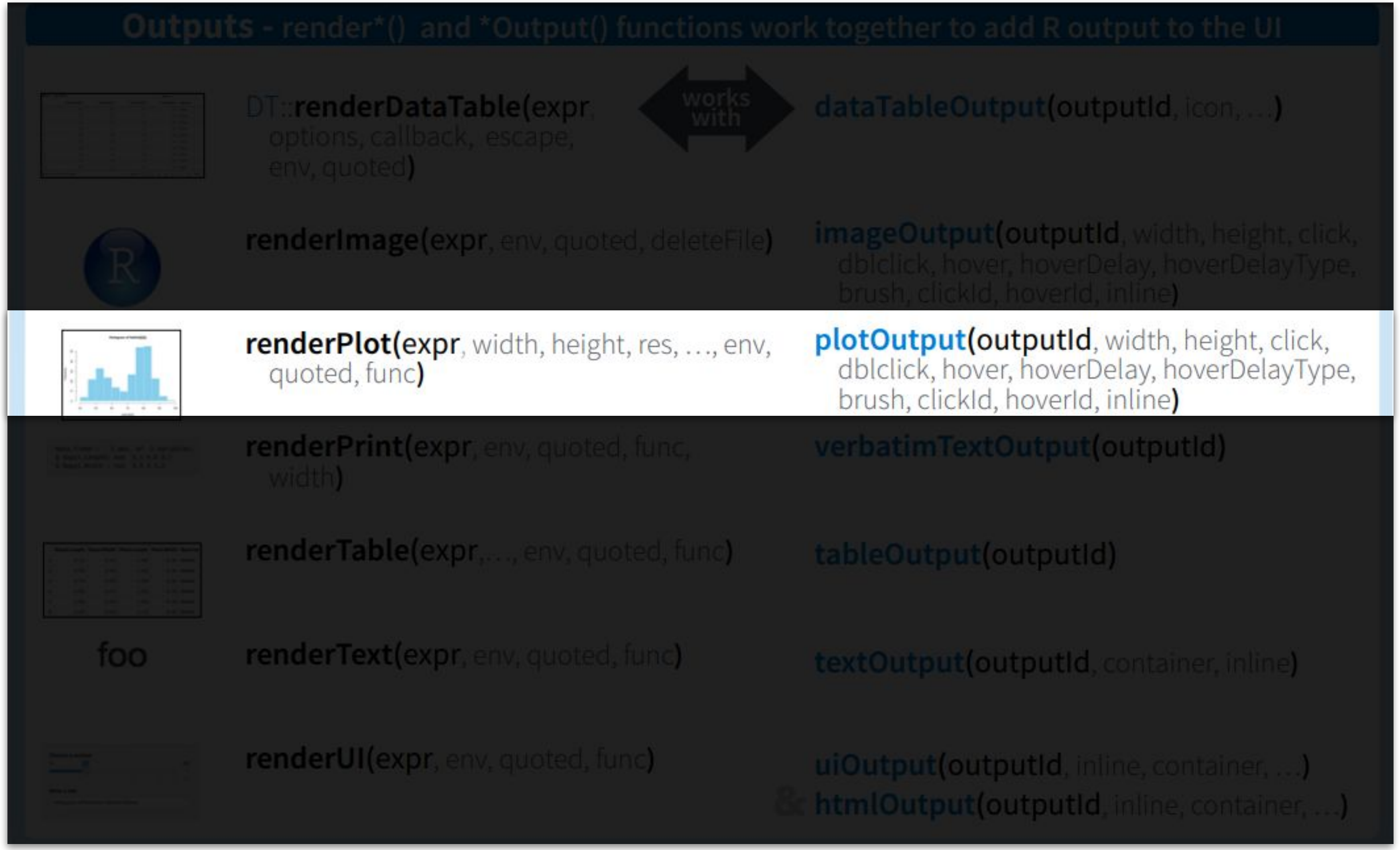
Size

population

alpha

0 0.5 1

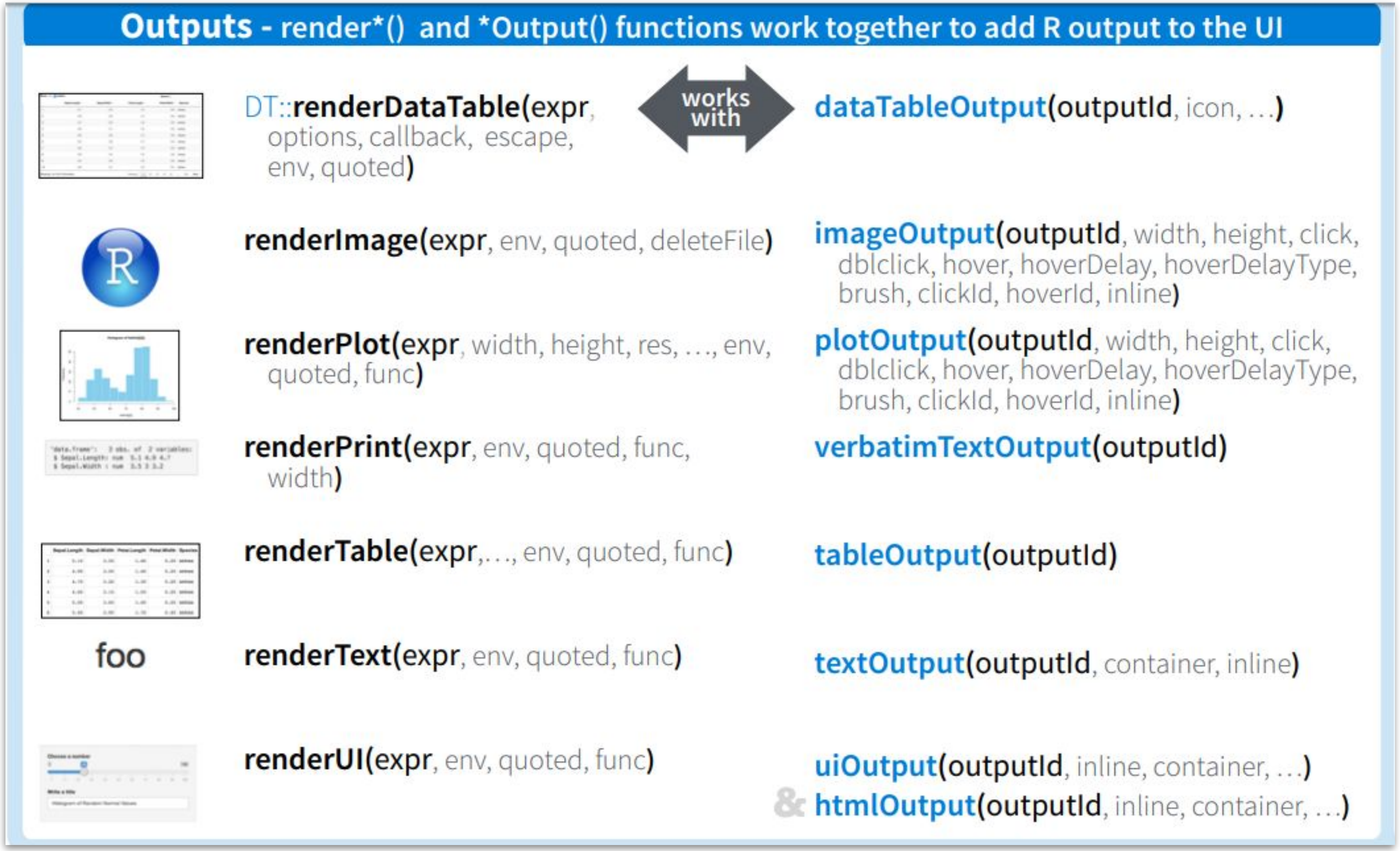
OUTPUTS



Outputs - render*() and *Output() functions work together to add R output to the UI

	works with	
DT::**renderDataTable(expr**, options, callback, escape, env, quoted**)**		**dataTableOutput(outputId**, icon, ...**)**
renderImage(expr, env, quoted, deleteFile**)**		**imageOutput(outputId**, width, height, click, dblclick, hover, hoverDelay, hoverDelayType, brush, clickId, hoverId, inline**)**
renderPlot(expr, width, height, res, ..., env, quoted, func**)**		**plotOutput(outputId**, width, height, click, dblclick, hover, hoverDelay, hoverDelayType, brush, clickId, hoverId, inline**)**
renderPrint(expr, env, quoted, func, width**)**		**verbatimTextOutput(outputId)**
renderTable(expr,..., env, quoted, func**)**		**tableOutput(outputId)**
foo **renderText(expr**, env, quoted, func**)**		**textOutput(outputId**, container, inline**)**
renderUI(expr, env, quoted, func**)**		**uiOutput(outputId**, inline, container, ...**)** & **htmlOutput(outputId**, inline, container, ...**)**

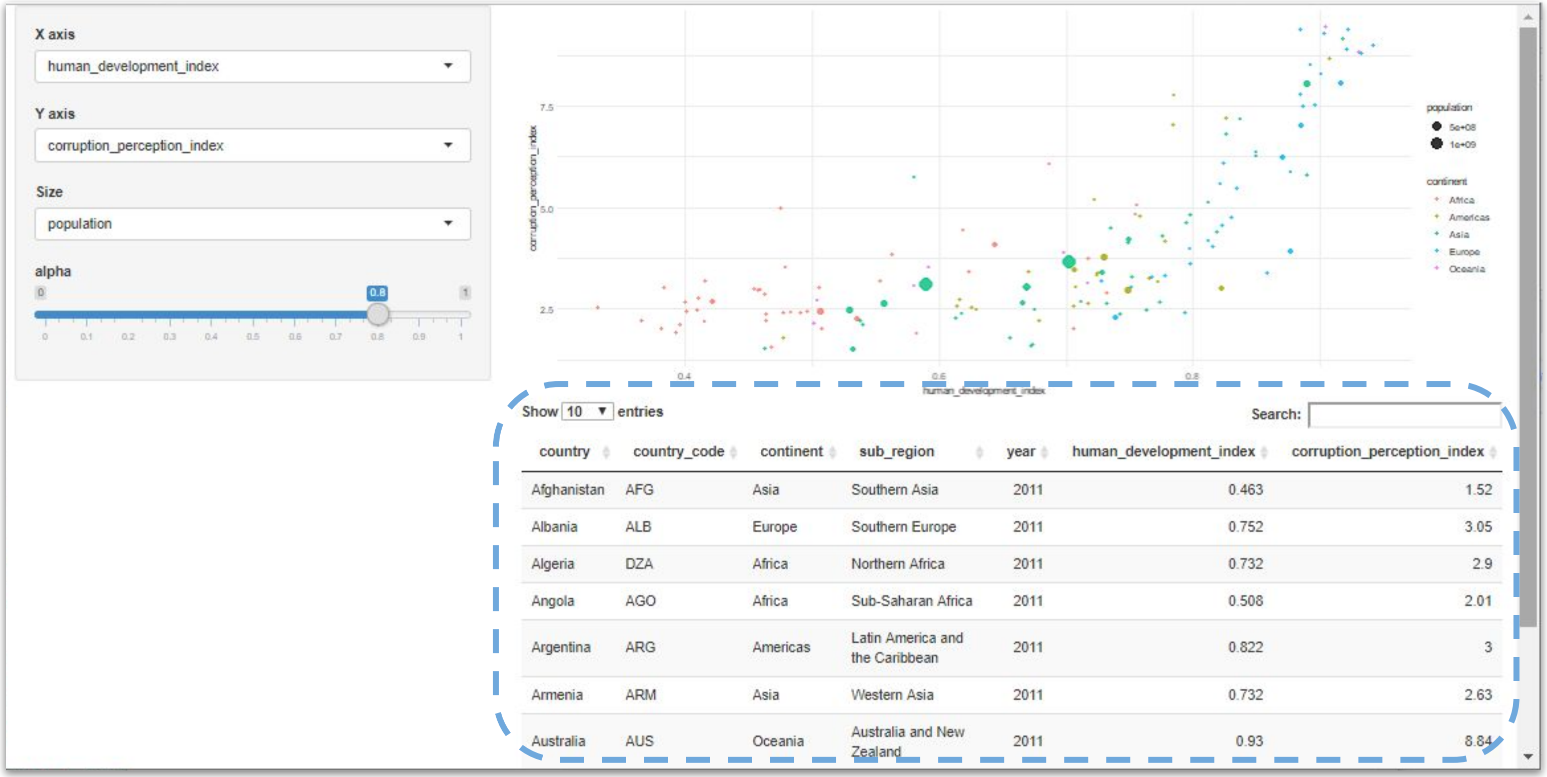
OUTPUTS



Outputs - render*() and *Output() functions work together to add R output to the UI

	works with	
DT::**renderDataTable(expr**, options, callback, escape, env, quoted**)**		**dataTableOutput(outputId**, icon, ...**)**
renderImage(expr, env, quoted, deleteFile**)**		**imageOutput(outputId**, width, height, click, dblclick, hover, hoverDelay, hoverDelayType, brush, clickId, hoverId, inline**)**
renderPlot(expr, width, height, res, ..., env, quoted, func**)**		**plotOutput(outputId**, width, height, click, dblclick, hover, hoverDelay, hoverDelayType, brush, clickId, hoverId, inline**)**
renderPrint(expr, env, quoted, func, width**)**		**verbatimTextOutput(outputId)**
renderTable(expr,..., env, quoted, func**)**		**tableOutput(outputId)**
foo **renderText(expr**, env, quoted, func**)**		**textOutput(outputId**, container, inline**)**
renderUI(expr, env, quoted, func**)**		**uiOutput(outputId**, inline, container, ...**)** & **htmlOutput(outputId**, inline, container, ...**)**

How would you add this table?



X axis: human_development_index

Y axis: corruption_perception_index

Size: population

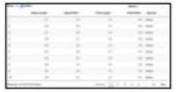
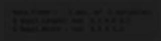
alpha: 0.8

Show 10 entries

Search:

country	country_code	continent	sub_region	year	human_development_index	corruption_perception_index
Afghanistan	AFG	Asia	Southern Asia	2011	0.463	1.52
Albania	ALB	Europe	Southern Europe	2011	0.752	3.05
Algeria	DZA	Africa	Northern Africa	2011	0.732	2.9
Angola	AGO	Africa	Sub-Saharan Africa	2011	0.508	2.01
Argentina	ARG	Americas	Latin America and the Caribbean	2011	0.822	3
Armenia	ARM	Asia	Western Asia	2011	0.732	2.63
Australia	AUS	Oceania	Australia and New Zealand	2011	0.93	8.84

Outputs - render*() and *Output() functions work together to add R output to the UI

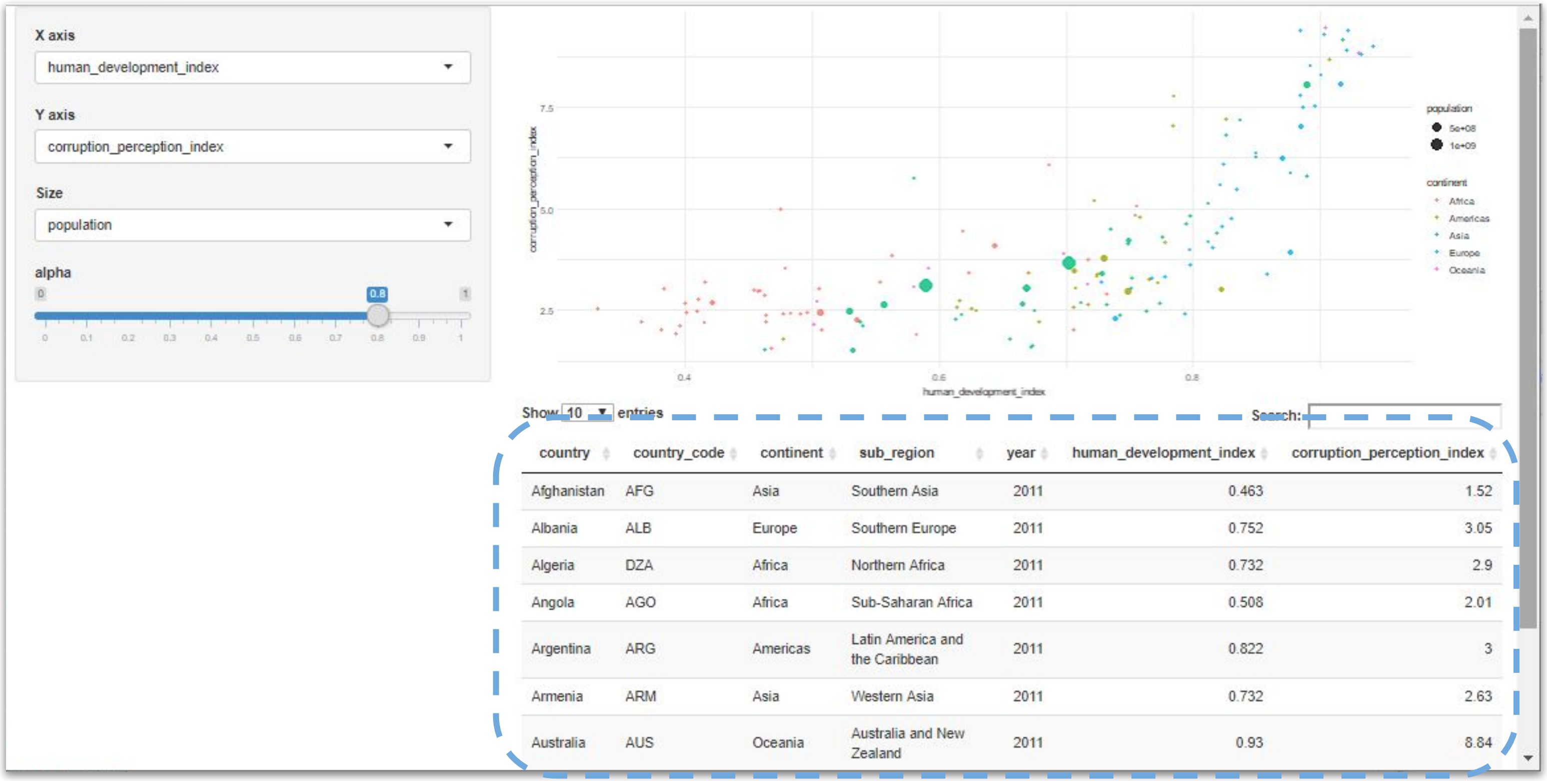
	DT::renderDataTable(expr, options, callback, escape, env, quoted)	works with	dataTableOutput(outputId, icon, ...)
	renderImage(expr, env, quoted, deleteFile)		imageOutput(outputId, width, height, click, dblclick, hover, hoverDelay, hoverDelayType, brush, clickId, hoverId, inline)
	renderPlot(expr, width, height, res, ..., env, quoted, func)		plotOutput(outputId, width, height, click, dblclick, hover, hoverDelay, hoverDelayType, brush, clickId, hoverId, inline)
	renderPrint(expr, env, quoted, func, width)		verbatimTextOutput(outputId)
	renderTable(expr,..., env, quoted, func)		tableOutput(outputId)
foo	renderText(expr, env, quoted, func)		textOutput(outputId, container, inline)
	renderUI(expr, env, quoted, func)		uiOutput(outputId, inline, container, ...) & htmlOutput(outputId, inline, container, ...)

EXERCISE

- Open **countries-04.R**
- Use `DT::renderDataTable` to create an a table showing the first 7 columns from `countries_data_2011`.
- Add `DT::dataTableOutput` inside `mainPanel()`.

countries-05.R

EXERCISE



X axis

human_development_index

Y axis

corruption_perception_index

Size

population

alpha

0 — 0.8 — 1

Show 10 entries

Search:

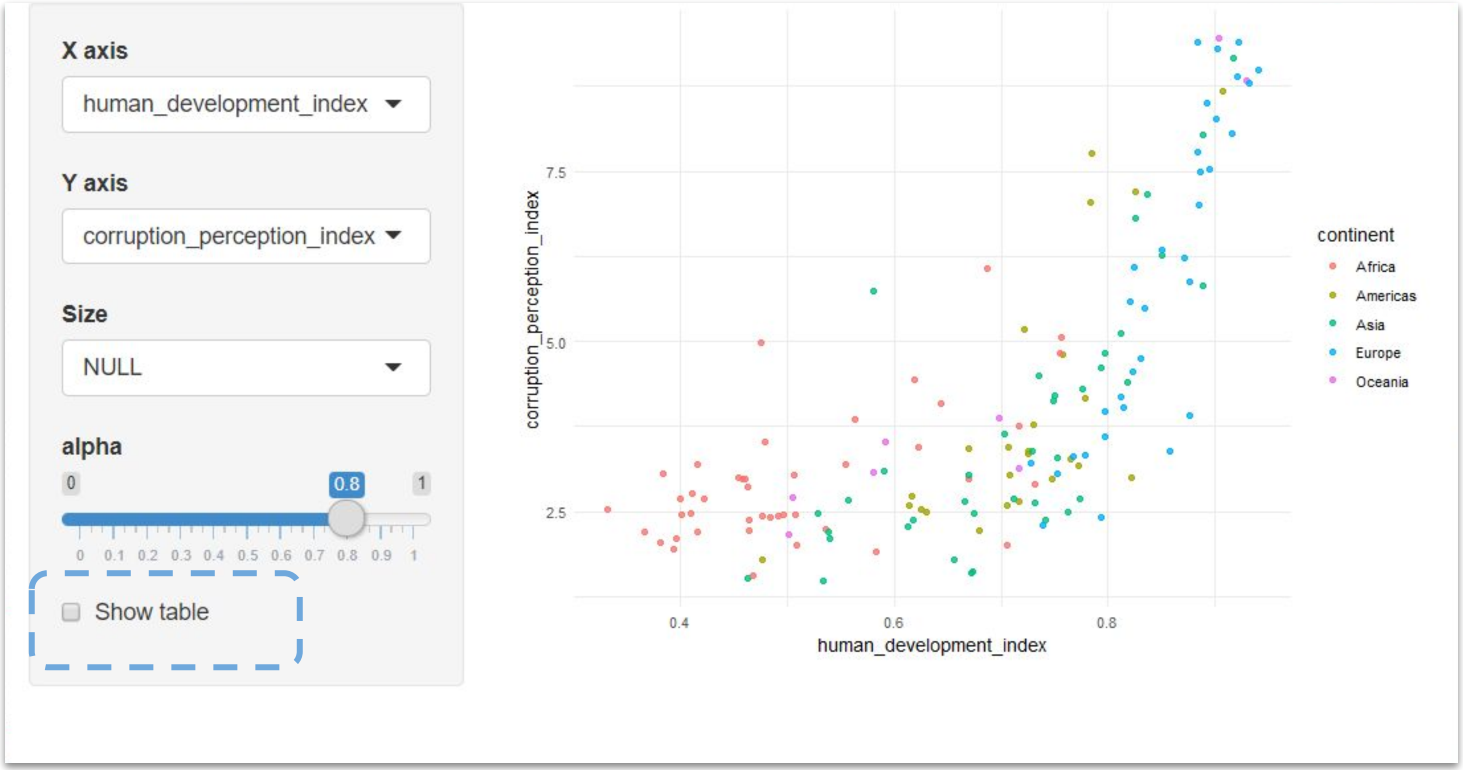
country	country_code	continent	sub_region	year	human_development_index	corruption_perception_index
Afghanistan	AFG	Asia	Southern Asia	2011	0.463	1.52
Albania	ALB	Europe	Southern Europe	2011	0.752	3.05
Algeria	DZA	Africa	Northern Africa	2011	0.732	2.9
Angola	AGO	Africa	Sub-Saharan Africa	2011	0.508	2.01
Argentina	ARG	Americas	Latin America and the Caribbean	2011	0.822	3
Armenia	ARM	Asia	Western Asia	2011	0.732	2.63
Australia	AUS	Oceania	Australia and New Zealand	2011	0.93	8.84

EXERCISE

- Open **countries-05.R**
- Add a `checkboxInput` in the `sidebarPanel()`.
- Use the value of the checkbox inside `DT::renderDataTable()` to show/hide the data table.

countries-06.R

EXERCISE



X axis

human_development_index

Y axis

corruption_perception_index

Size

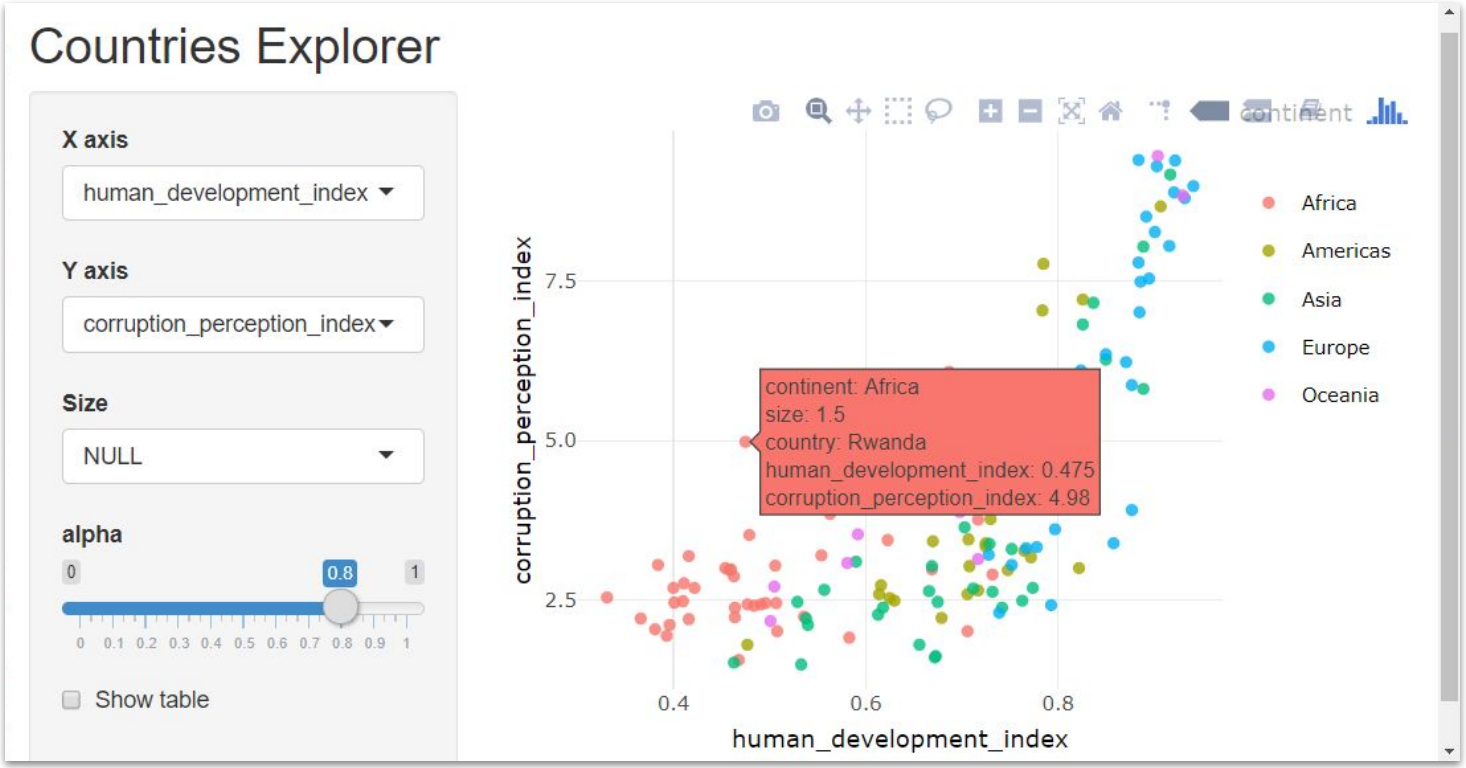
NULL

alpha

0.8

Show table

countries-07.R



Countries Explorer

X axis

human_development_index

Y axis

corruption_perception_index

Size

NULL

alpha

0 0.8 1

Show table

SHINY APP
FILE STRUCTURE

SHINY APP FILE STRUCTURE

Single File

app.R

Multiple Files

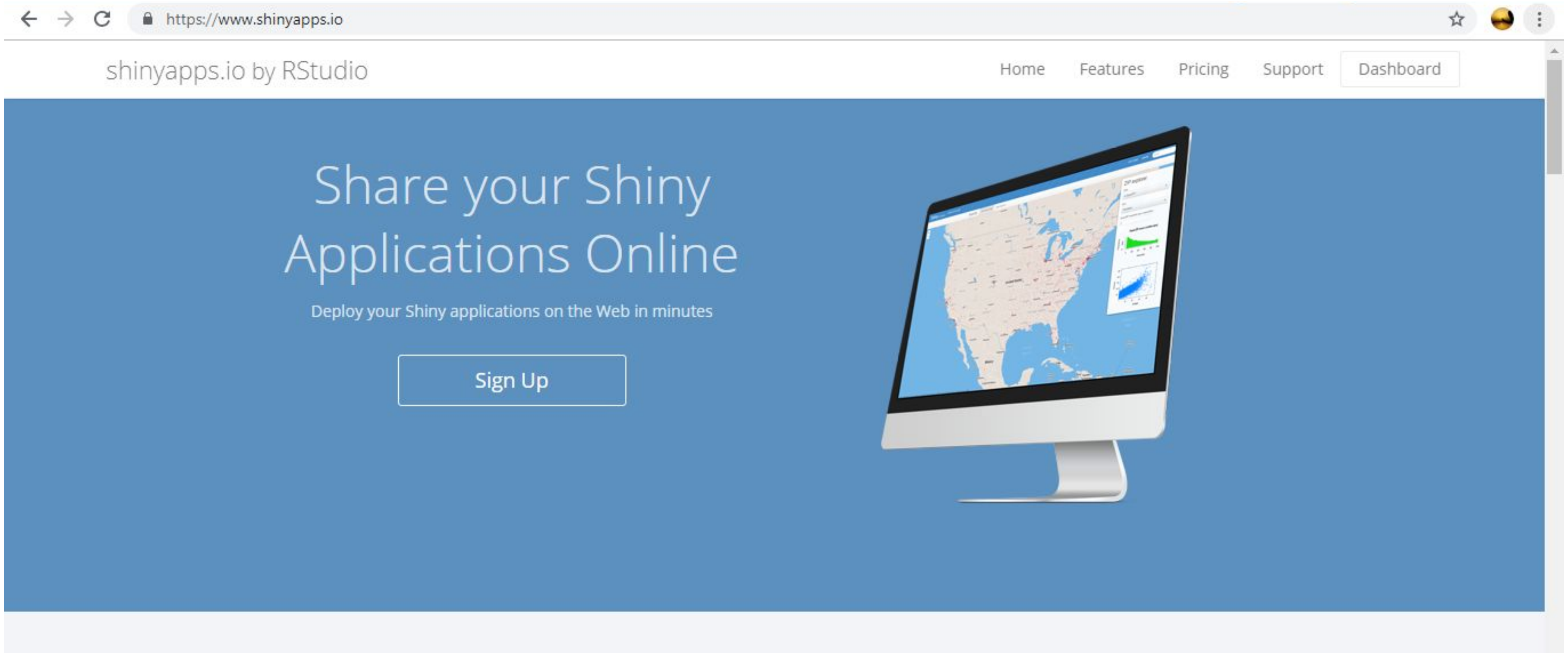
ui.R

server.R

Example:
https://github.com/rstudio/shiny-examples/tree/master/087-crandash

SHARING SHINY APPS

SHARING SHINY APPS



https://www.shinyapps.io

shinyapps.io by RStudio

Home Features Pricing Support Dashboard

Share your Shiny Applications Online

Deploy your Shiny applications on the Web in minutes

Sign Up