Workflow

Importing Data

Cleaning Data

Analysis

RStudio

Some text here to explain what is going on

shinyApp()

Reports

Shiny Apps

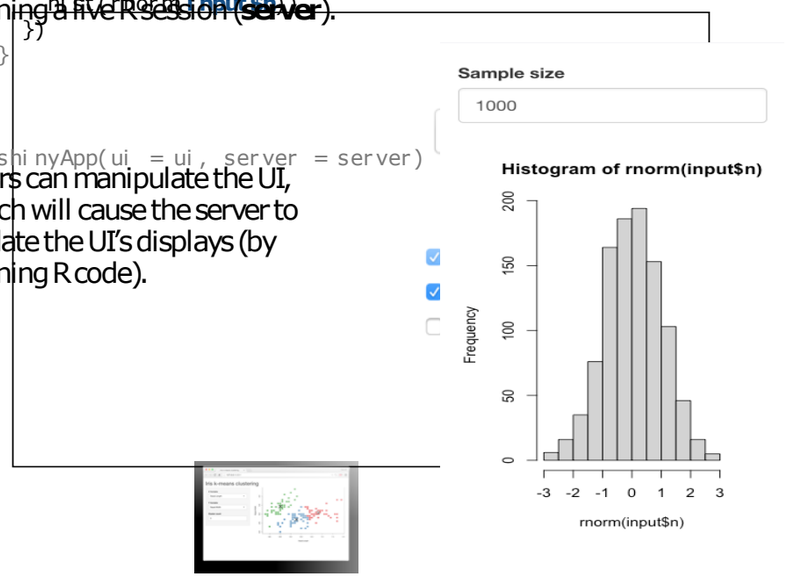
Created by ARI biometrics

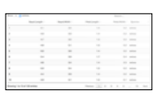
Building an App

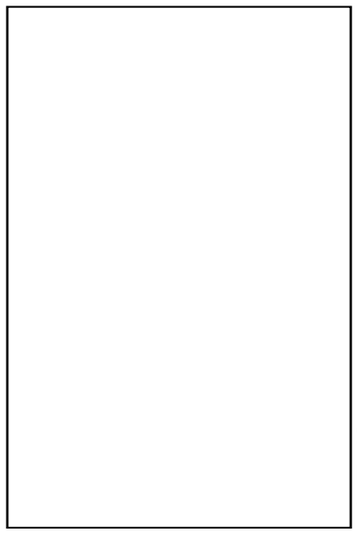
A **Shiny** app is a web page (**ui**) connected to a computer running a live R session (**server**).

Users can manipulate the UI, which will cause the server to update the UI's displays (by running R code).

```
# app.R
library(shiny)

ui <- fluidPage(
  numericInput(inputId = "n",
    "Sample size", value = 25),
  plotOutput(outputId = "hist")
)

server <- function(input, output, session) {
  output$hist <- renderPlot({
    hist(rnorm(input$n))
  })
}

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

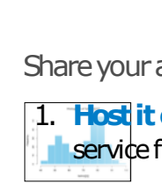






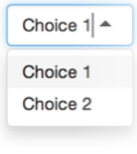
See annotated examples of Shiny apps by running **runExample(**<example name>**)**. Run **runExample()** with no arguments for a list of example names.

Call shinyApp() to combine ui and server into an interactive app!



Share your app in three ways:

1. **Host it on shinyapps.io**, a cloud based service from Posit. To deploy Shiny apps:
   - Create a free or professional account at **shinyapps.io**
   - Click the Publish icon in RStudio IDE, or run: **rsconnect::deployApp(**"<path to directory>"**)**
2. **Purchase Posit Connect**, a publishing platform for R and Python. **posit.co/products/enterprise/connect/**
3. **Build your own Shiny Server** **posit.co/products/open-source/shinyserver/**



Outputs

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

DT::**renderDataTable(**expr, options, searchDelay, callback, escape, env, quoted, outputArgs**)**	**dataTableOutput(**outputId**)**
renderImage(expr, env, quoted, deleteFile, outputArgs**)**	**imageOutput(**outputId, width, height, click, dblclick, hover, brush, inline**)**
renderPlot(expr, width, height, res, ... , alt, env, quoted, execOnResize, outputArgs**)**	**plotOutput(**outputId, width, height, click, dblclick, hover, brush, inline**)**
renderPrint(expr, env, quoted, width, outputArgs**)**	**verbatimTextOutput(**outputId, placeholder**)**
renderTable(expr, striped, hover, bordered, spacing, width, align, rownames, colnames, digits, na, ... , env, quoted, outputArgs**)**	**tableOutput(**outputId**)**
renderText(expr, env, quoted, outputArgs, sep**)**	**textOutput(**outputId, container, inline**)**
renderUI(expr, env, quoted, outputArgs**)**	**uiOutput(**outputId, inline, container, ... **)** **htmlOutput(**outputId, inline, container, ... **)**

foo

Inputs

Collect values from the user.

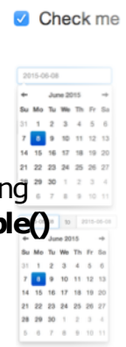
Access the current value of an input object with **input$<inputId>**. Input values are **reactive**.

actionButton(inputId, label, icon, width, ... **)**

actionLink(inputId, label, icon, ... **)**

checkboxGroupInput(inputId, label, choices, selected, inline, width, choiceNames, choiceValues**)**

checkboxInput(inputId, label, value, width**)**



dateInput(inputId, label, value, min, max, format, startview, weekstart, language, width, autoclose, datesdisabled, daysofweekdisabled**)**

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

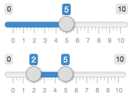
fileInput(inputId, label, multiple, accept, width, buttonLabel, placeholder**)**

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

passwordInput(inputId, label, value, width, placeholder**)**

radioButtons(inputId, label, choices, selected, inline, width, choiceNames, choiceValues**)**

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, timeFormat, timezone, dragRange**)**

textInput(inputId, label, value, width, placeholder**)** Also **textAreaInput()**