Intro To Shiny: Exercises

## Exercises

These exercises accompany the Intro to Shiny tutorial.

1. If you have not already, create a folder called ChicagoApp in your working directory and place the ui.R and server.R scripts from the tutorial in that folder. Use runApp() to open the application.

[Solution](#ex1)

### Advanced Exercises

1. Add a table of the data to the bottom of the application. Use dataTableOutput() in the ui.R script and renderDataTable() in the server.R script. See this page for explanation of what DataTables are, and see this page for a live example of how to use these functions in a Shiny app.

[Solution](#ex2)

1. Add a Leaflet map to the application with a marker at the location of the monitor (latitude 42.14 and longitude -87.79923). See this page for a demonstration of how to put a Leaflet map in a Shiny app. Remember to library the leaflet package in your ui.R and server.R scripts.

[Solution](#ex3)

1. Modify the ggplot2 output with the plotly package. See the project website for more information on plotly. See the documentation on how to use the ggplotly() function in a Shiny app. And change geom\_line() to geom\_point() for a cleaner image.

[Solution](#ex4)

## Solutions

### Solution 1

runApp("ChicagoApp")

[Back to exercises](#ex)

### Solution 2

**ui.R**

library(shiny)  
  
# Define UI for application that plots time series  
shinyUI(fluidPage(  
  
 # Application title  
 titlePanel("Chicago Air"),  
  
 # Sidebar with a dropdown for selecting parameter  
 sidebarLayout(  
 sidebarPanel(  
 selectInput("parameter",  
 "Select Parameter:",  
 c("Ozone" = "ozone",  
 "Temperature" = "temp",  
 "Solar Radiation" = "solar"))  
 ),  
  
 # Show a plot of the time series  
 mainPanel(  
 plotOutput("timePlot"),  
 dataTableOutput("dataTable")  
 )  
 )  
))

**server.R**

library(shiny)  
library(region5air)  
data(chicago\_air)  
chicago\_air$date <- as.Date(chicago\_air$date)  
  
 # Define server logic required to plot the time series  
shinyServer(function(input, output) {  
   
 output$timePlot <- renderPlot({  
 ggplot(chicago\_air, aes\_string("date", input$parameter)) + geom\_line()   
 })  
   
 output$dataTable <- renderDataTable({  
 chicago\_air  
 })  
})

runApp("ChicagoApp")

[Back to exercises](#ex)

### Solution 3

**ui.R**

library(shiny)  
library(leaflet)  
  
# Define UI for application that plots time series  
shinyUI(fluidPage(  
  
 # Application title  
 titlePanel("Chicago Air"),  
  
 # Sidebar with a dropdown for selecting parameter  
 sidebarLayout(  
 sidebarPanel(  
 leafletOutput("mymap"),  
 selectInput("parameter",  
 "Select Parameter:",  
 c("Ozone" = "ozone",  
 "Temperature" = "temp",  
 "Solar Radiation" = "solar"))  
 ),  
  
 # Show a plot of the time series  
 mainPanel(  
 plotOutput("timePlot"),  
 dataTableOutput("dataTable")  
 )  
 )  
))

**server.R**

library(shiny)  
library(region5air)  
library(leaflet)  
data(chicago\_air)  
chicago\_air$date <- as.Date(chicago\_air$date)  
  
 # Define server logic required to plot the time series  
shinyServer(function(input, output) {  
   
 output$timePlot <- renderPlot({  
 ggplot(chicago\_air, aes\_string("date", input$parameter)) + geom\_line()   
 })  
   
 output$dataTable <- renderDataTable({  
 chicago\_air  
 })  
   
 output$mymap <- renderLeaflet({  
 library(leaflet)  
 m <- leaflet()  
 m <- addTiles(m)  
 m <- addMarkers(m, lng=-87.79923, lat=42.14)  
 setView(m, lng=-87.79923, lat=42.14, zoom = 10)  
 })  
})

runApp("ChicagoApp")

[Back to exercises](#ex)

### Solution 4

**ui.R**

library(shiny)  
library(leaflet)  
library(plotly)  
  
# Define UI for application that plots time series  
shinyUI(fluidPage(  
  
 # Application title  
 titlePanel("Chicago Air"),  
  
 # Sidebar with a dropdown for selecting parameter  
 sidebarLayout(  
 sidebarPanel(  
 leafletOutput("mymap"),  
 selectInput("parameter",  
 "Select Parameter:",  
 c("Ozone" = "ozone",  
 "Temperature" = "temp",  
 "Solar Radiation" = "solar"))  
 ),  
  
 # Show a plot of the time series  
 mainPanel(  
 plotlyOutput("timePlot"),  
 dataTableOutput("dataTable")  
 )  
 )  
))

**server.R**

library(shiny)  
library(region5air)  
library(leaflet)  
library(ggplot2)  
library(plotly)  
  
data(chicago\_air)  
chicago\_air$date <- as.Date(chicago\_air$date)  
  
shinyServer(function(input, output) {  
   
 output$timePlot <- renderPlotly({  
 g <- ggplot(chicago\_air, aes\_string("date", input$parameter)) + geom\_point()  
 ggplotly(g)  
 })  
   
 output$dataTable <- renderDataTable({  
 chicago\_air  
 })  
   
 output$mymap <- renderLeaflet({  
 library(leaflet)  
 m <- leaflet()  
 m <- addTiles(m)  
 m <- addMarkers(m, lng=-87.79923, lat=42.14)  
 setView(m, lng=-87.79923, lat=42.14, zoom = 10)  
 })  
   
})

runApp("ChicagoApp")

[Back to exercises](#ex)