

BUILDING INTERACTIVE, R-POWERED APPLICATIONS with 'Shiny' and 'RGtk2'

Jingyi Guo

1, Oct, 2015

Shiny

- ▶ Open-Sourced by RStudio 11/2012 on CRAN
- ▶ New model for web-accessible R code
- ▶ Able to generate basic web UIs
- ▶ Built on a “Reactive Programming” model

Getting started: Setup

- ▶ `R> install.packages("shiny")` from CRAN
- ▶ Create directory `HelloShiny`
- ▶ Edit `ui.r`
- ▶ Edit `server.r`
- ▶ `R> shiny::runApp("HelloShiny")`

Getting started: server.R

The Core Component with functionality for input and output as plots, tables and plain text.

```
shinyServer(function(input, output) {  
  output$distPlot <- renderPlot({  
    dist <- rnorm(input$obs)  
    hist(dist)  
  })  
})
```

Getting started: ui.R

This file creates the structure of HTML

```
shinyUI(fluidPage(  
  headerPanel("Example Hello Shiny"),  
  sidebarPanel(  
    sliderInput("obs", "", 0, 1000, 500)  
  ),  
  mainPanel(  
    plotOutput("distPlot")  
  )  
))
```

Getting started: A simple shiny App

A shiny App contains two parts:

ui.R

```
shinyUI(fluidPage(  
  ))
```

server.R

```
shinyServer(function(input, output){  
  })
```

UI Layout

Simple layout: Sidebar Layout

ui.R

```
shinyUI(fluidPage(  
  titlePanel("title panel"),  
  
  sidebarLayout(  
    sidebarPanel("sidebar panel"),  
    mainPanel("main panel")  
  )  
))
```

server.R

```
shinyServer(function(input, output){  
  })
```

UI Layout

Fancy layout:

- ▶ Grid Layout
- ▶ Tabsets Panel
- ▶ Navlists Panel

UI Input - Add control widgets

The widgets in Shiny

- ▶ `actionButton`
- ▶ `checkboxGroupInput` & `checkboxInput`
- ▶ `dateInput` & `dateRangeInput`
- ▶ `fileInput`
- ▶ `helpText`
- ▶ `numericInput`
- ▶ `radioButtons`
- ▶ `selectInput`
- ▶ `sliderInput`
- ▶ `submitButton`
- ▶ `textInput`

UI Input - How do they work?

A example of Action Button.....

UI Output

ui.R

- ▶ `htmlOutput`
- ▶ `plotOutput`
- ▶ `tableOutput`
- ▶ `textOutput`
- ▶ `verbatimTextOutput`
- ▶ `downloadButton`

Reactive functions

Functions that you use in your application's server side code, assigning them to outputs that appear in your user interface.

server.R

- ▶ `renderPlot` (`renderImage`)
- ▶ `renderText`
- ▶ `renderTable`
- ▶ `renderPrint`
- ▶ `renderUI`

Example - A simple R-INLA tutorial

This example shows how to use the UI and containers....

ui.R

```
shinyUI(navbarPage("A Simple R-INLA tutorial",  
  tabPanel("R-INLA"),  
  tabPanel("SPDE")  
)  
)
```

See **Step 1**....

Example - A simple R-INLA tutorial

This example shows how to use the UI and containers....

"R-INLA" tabPanel

```
tabPanel("R-INLA",  
  fluidRow(  
    column(4,  
      wellPanel("Sidebar Panel")  
    ),  
    column(8,  
      wellPanel("Main Panel")  
    )  
  )  
)
```

See **Step 2**....

Example - A simple R-INLA tutorial

This example shows how to use the UI and containers....

"R-INLA" fluidRow

```
column(4,  
  wellPanel("UI-Input 1"),  
  wellPanel("UI-Input 2"),  
  wellPanel("UI-Input 3")  
)  
  
column(8,  
  wellPanel("UI-Output 1"),  
  wellPanel("UI-Output 2"),  
  wellPanel("UI-Output 3")  
)
```

See **Step 3**....

Example - A simple R-INLA tutorial

This example shows how to use the UI and containers....

UI-Input 1

```
wellPanel(selectInput("latent",  
                      label = "Latent models",  
                      width='100%',  
                      choices = list("NULL" = 1,  
                                    "iid" = 2,  
                                    "rw2" = 3),  
                      selected = 1))
```

See **Step 4**....

Example - A simple R-INLA tutorial

This example shows how to use the UI and containers....

UI-Input 3

```
wellPanel(radioButtons("radio",  
                        label = "Show simulate data",  
                        choices = list("No" = 1,  
                                      "Yes" = 2),  
                        selected = 1,  
                        inline = TRUE))
```

See **Step 5**....

Example - A simple R-INLA tutorial

This example shows how to use the UI and containers....

"R-INLA" UI-Output

```
column(8,  
  conditionalPanel(  
    condition = "input.radio == 2",  
    wellPanel(plotOutput("dataPlot"))),  
  wellPanel(htmlOutput("inlafuncs",  
                        container = span)),  
  wellPanel(verbatimTextOutput("inlares"))  
)
```

See **Step 6**....

Example - A simple R-INLA tutorial

This example shows how to use the UI and containers....

Corresponding reaction

```
output$dataPlot <- renderPlot({  
  plot(y,ylab="y")  
})
```

See **Step 6**....

Example - A simple R-INLA tutorial

This example shows how to use the UI and containers....

Corresponding reaction

```
output$inlafuncs <- renderUI({
  if(isolate(input$latent==1)){
    if(input$likelihood==1){
      return(HTML("r = inla(y ~ 1 + x,
        data = data.frame(y, x, E),
        family = \"nbinomial\", E=E)"))
    }
    if(isolate(input$likelihood==2)){
      return(HTML("r = inla(y ~ 1 + z,
        data = list(y=y, z=z),
        family = \"gaussian\")"))
    }
  }
})
```

Example - A simple R-INLA tutorial

This example shows how to use the UI and containers....

Corresponding reaction

```
output$inlares <- renderPrint({  
  summary(res)  
})
```

See **Step 6**....

RGtk2

- ▶ RGtk2 is a GUI toolkit for R
- ▶ RGtk2 provides programmatic access to **GTK+** (GIMP ToolKit)
- ▶ cross-platform (Windows, Mac, and Linux)
- ▶ each UI defined as a gObject

Layout

- ▶ `gtkWindow`
- ▶ `gtkAlignment`
- ▶ `gtkVBox` and `gtkHBox`
- ▶ `gtkNotebook`