virtual7 | CONFERENCE 2019

RAPID PROTOTYPING DATA ANALYTICS (WEB) APPS WITH R/SHINY

PETER MEISSNER

2019-06-28

(UPDATED: 2019-06-26)

You are a data scientist now doing ...

... algorithm, analytics, compliance/control, reporting, automation

SPECTACULAR

Nobody Cares

PROBLEM

WHY?

😟 no time 😟

😱 no trust 😱

😩 no availability 😩

😒 no explanation 😒

🙃 no workflow fit 🙃

😟 no orders 😟

PROBLEM

YOU HAVE TO DO BETTER

show value

build trust

You need a GUI. FAST!

with minimal amount of resources

SOLUTION

SHINY

Intro

- R based **server-client** application framework
- Server: **R**
- Client: **browser + HTML + Javascript**
- Communication: **Websockets**

Writing an app

- Server function
- UI function
- Widgets are automtically functions of reactive data

Productivity Boosters

- Reactivity
- Widgets
  - **D3.js** bindings
  - Highcharts bindings
  - Leavelet bindings
  -

Production Ready

A SIMPLE MODEL

global.R

```
# model data
mdl <-
  glm(
    formula = women ~ height,
    data    = height,
    family  = binomial(link = logit)
  )


# make prediction available in app sessions
APPDATA$predict_sex <-
  function(height){
    pred <-
      broom::augment(
        mdl,
        newdata = data.frame(height = height),
        type.predict = "response"
      )
    round(pred$.fitted * 100,2)
  }
```

DEFINE LAYOUT

ui.R

```
material_card(
  material_slider(
    input_id      = "input_height",
    label         = "your height in cm",
    min_value     = 140,
    max_value     = 190,
    step_size     = 1,
    initial_value = 165
  )
)
```

DEFINE LAYOUT

ui.R

```
material_card(
  "Prediction P(height -> women)",
  uiOutput(
    outputId = "out_prediction"
  )
)
```

DEFINE LAYOUT

ui.R

```
material_card(
  plotOutput("out_plot")
)
```

COMPUTE OUTPUT

server.R

```
## prediction
output$out_prediction <-
  renderUI(
    {
      tmp <- APPDATA$predict_sex(input$input_height)
      tags$span(
        paste(tmp, "%"),
        style = "font-size: 150%;",
        class =
          if ( tmp < 50 ){
            "light-blue"
          } else {
            "orange"
          }
      )
    }
  )
```

COMPUTE OUTPUT

server.R

```
## plot model
output$out_plot <-
  renderPlot({
    df <-
      data.frame(
        height  = 140:190,
        P_women = APPDATA$predict_sex(140:190)
      )

    ggplot(df, aes(x = height, y = P_women)) +
      geom_line() +
      geom_vline(aes(xintercept = input$input_height), color = "steelblue") +
      geom_point(data = height, aes(color = women, x = height, y = women * 100)) +
      scale_color_manual(values = c("lightblue", "orange")) +
      theme_bw()
  })
```