

VISUALIZATION OF THE BIOPSY DATA USING R

Oluwadare, Margaret

10/6/2020

OVERVIEW

This RStudio shiny application is developed as a part of final project in the Developing Data Products course in Coursera Data Science Specialization track offered by John Hopkin University. The application visualizes the biospy data. It is an adapted app from <https://rviews.rstudio.com/2018/09/20/shiny-r2d3/>

The project includes:

- ▶ A D3 code using r2d3 containing canvas size functions, attributes, The on functions track named events, The click function, and The mouseover and mouseout.
- ▶ A R/shiny code The app is hosted at: <https://margaretoluwadare.shinyapps.io/VisualizingBiopsydata/>

APPLICATION WIDGET USED:

This application the following widgets listed below:

- ▶ `r2d3` scripts: containing canvas size functions, attributes, The `on` functions track named events, The `click` function, and The `mouseover` and `mouseout`. It also contain the `select` button,
- ▶ `r2d3_file <- tempfile()` and then `writeLines(r2d3_script, r2d3_file)` is used to keep the D3 and R code in one location.
- ▶ Attributes (`.attr`) which are named after each variable.
 - The `on()` functions track named events e.g. `click`, `mouseover`, and `mouseout`.

OPERATIONS IN THE APP AND AOUTPUT

The reactivity of the shiny application widgets is controlled by using a `Select Button`. Based on user variable selection and data input, and using the `simpler2d3` code the application displays:

- ▶ Variables of data read by the user
- ▶ barplot of the chosen variable
- ▶ observed values of the variable in relation to other variable in the data

```
library("knitr")  
library(shiny)  
library(dplyr)  
library(r2d3)  
library(forcats)  
library(MASS)  
library(DT)  
library(rlang)
```

```
Biopsy <- attach(biopsy)
```

```

r2d3_script <- "
// !preview r2d3 data= data.frame(y = 0.1, ylabel = '1%', t
function svg_height() {return parseInt(svg.style('height'))
function svg_width() {return parseInt(svg.style('width'))}
function col_top() {return svg_height() * 0.05; }
function col_left() {return svg_width() * 0.20; }
function actual_max() {return d3.max(data, function (d) {re
function col_width() {return (svg_width() / actual_max())
function col_heighth() {return svg_height() / data.length *

var bars = svg.selectAll('rect').data(data);
bars.enter().append('rect')
    .attr('x',      col_left())
    .attr('y',      function(d, i) { return i * col_heighth
    .attr('width',  function(d) { return d.y * col_width()
    .attr('height', col_heighth() * 0.9)
    .attr('fill',   function(d) {return d.fill; })
    .attr('id',     function(d) {return (d.label); })
    .on('click', function(){
        Shiny.setInputValue('bar_clicked', d3.select(this).at

```

```

ui <- fluidPage(
  selectInput("var", "Variable",
              list("ID", "age", "mnp", "ts", "inv", "ndc",
                   selected = "class"),
  d3Output("d3"),
  DT::dataTableOutput("table"),
  textInput("val", "Value", "class")
)

```

```

server <- function(input, output, session) {
  output$d3 <- renderD3({
    biopsy %>%
      mutate(label = !!sym(input$var)) %>%
      group_by(label) %>%
      tally() %>%
      arrange(desc(n)) %>%
      mutate(
        y = n,
        ylabel = prettyNum(n, big.mark = ","),
        fill = ifelse(label != input$val, "#E69F00", "red")
      )
  })
}

```

