

Problems in R

Interactive Web Applications with Shiny

- a) Expand the histogram-of-random-values example from the slides.
- (a) Add another input (`selectInput()` or `radioButtons()`) and let the user choose among drawing samples from three distributions: normal, uniform (`runif()`), or exponential (`rexp()`).
 - (b) Add a layout: Try both, (1) an automatic layout using `sidebarLayout()`, `sidebarPanel()`, and `mainPanel()`; (2) a custom layout using `fluidRow()` and `column()`.
- b) Create a Shiny app that displays a boxplot for the `PlantGrowth` data in R.
- (a) Let the user select the groups (one, two, or three) to display.
 - (b) Add an output that shows the results of a one-sample t test, a two-sample t test (`t.test()`), or a one-way ANOVA (`oneway.test()`), depending on whether one, two, or three groups are selected.
 - (c) Add an input that lets the user select whether to treat the variances in the groups as being equal. Accordingly, display the correct test. Hint: Use the `var.equal` argument.
- c) Create a Shiny app with the user interface consisting of an `actionButton()` input and two outputs: `plotOutput()` and `verbatimTextOutput()`. On each button click,
- a new 50-observations bootstrap sample is drawn with replacement from the `cars` data,
 - the scatter plot is displayed,
 - descriptive statistics (`summary()`) are shown.
- Hint: In the server function, use `getsamples <- reactive({...})` to be able to access the same data set both within `renderPlot()` and `renderPrint()`.
- d) Expand the previous Shiny app.
- (a) Add a check box to allow the user to show the regression line.
 - (b) Add a second button that restores the sample to the original `cars` data. Hint: To get two buttons to work in the server function, combine `reactiveValues()` and `observe()`. More here: <http://shiny.rstudio.com/articles/action-buttons.html>.