

1. In RStudio create a new file of type "Shiny Web App".
2. Paste this code into the file just below the line that loads shiny: "library(shiny)".

```
global_monitor <- tibble(  
  scientist_work = c(rep("Benefits", 80000), rep("Doesn't benefit", 20000)))
```

```
sample_props50 <- global_monitor %>%  
  rep_sample_n(size = 50, reps = 15000, replace = TRUE) %>%  
  count(scientist_work) %>%  
  mutate(p_hat = n / sum(n)) %>%  
  filter(scientist_work == "Doesn't benefit")
```

```
# Define UI for application that draws a histogram
```

```
shinyApp(  
  ui <- fluidPage(  
  
    # Sidebar with a slider input for number of bins  
    sidebarLayout(  
      sidebarPanel(  
  
        selectInput("outcome",  
          "Outcome of interest:",  
          choices = c("Benefits", "Doesn't benefit"),  
          selected = "Doesn't benefit"),  
  
        numericInput("n_samp",  
          "Sample size:",  
          min = 1,  
          max = nrow(global_monitor),  
          value = 30),  
  
        numericInput("n_rep",  
          "Number of samples:",  
          min = 1,  
          max = 30000,  
          value = 15000),  
  
        hr(),  
  
        sliderInput("binwidth",  
          "Binwidth:",  
          min = 0, max = 0.5,  
          value = 0.02,  
          step = 0.005)
```

```

    ),

    # Show a plot of the generated distribution
    mainPanel(
      plotOutput("sampling_plot"),
      textOutput("sampling_mean"),
      textOutput("sampling_se")
    )
  )
),

server <- function(input, output) {

  # create sampling distribution
  sampling_dist <- reactive({
    global_monitor %>%
      rep_sample_n(size = input$n_samp, reps = input$n_rep, replace = TRUE) %>%
      count(scientist_work) %>%
      mutate(p_hat = n / sum(n)) %>%
      filter(scientist_work == input$outcome)
  })

  # plot sampling distribution
  output$sampling_plot <- renderPlot({

    ggplot(sampling_dist(), aes(x = p_hat)) +
      geom_histogram(binwidth = input$binwidth) +
      xlim(0, 1) +
      labs(
        x = paste0("p_hat (", input$outcome, ")"),
        title = "Sampling distribution of p_hat",
        subtitle = paste0("Sample size = ", input$n_samp, " Number of samples = ", input$n_rep)
      ) +
      theme(plot.title = element_text(face = "bold", size = 16))
  })

  ggplot(data = sample_props50, aes(x = p_hat)) +
    geom_histogram(binwidth = 0.02) +
    labs(
      x = "p_hat (Doesn't benefit)",
      title = "Sampling distribution of p_hat",
      subtitle = "Sample size = 50, Number of samples = 15000"
    )

  # mean of sampling distribution
  output$sampling_mean <- renderText({
    paste0("Mean of sampling distribution = ", round(mean(sampling_dist()$p_hat), 2))
  })
}

```

```

})

# mean of sampling distribution
output$sampling_se <- renderText({
  paste0("SE of sampling distribution = ", round(sd(sampling_dist())$p_hat, 2))
})
},

options = list(height = 900)
)

```

3. Save the file with some name like "scientists" to a folder where the .RData and .Rhistory files are for your RStudio project for this particular lab. For example, those files for me are in a folder: C:\CUNY\DATA606\Chapter05\Lab.

4. In RStudio at the top right of the code editing pane there is a button to "Run App". After you run it it should change to Reload App.

5. A browser page should open where you can select Outcome of Interest, adjust Sample Size, adjust Number of Samples and adjust Binwidth. Changes to these parameters should cause a histogram to replot.

