```
# Load libraries
library(shiny)
library(tidyverse)

# Read in data
adult <- read_csv("adult.csv")
# Convert column names to lowercase for convenience
names(adult) <- tolower(names(adult))

# Define server logic
shinyServer(function(input, output) {

  df_country <- reactive({
    adult %>% filter(native_country == input$country)
  })

  # TASK 5: Create logic to plot histogram or boxplot
  output$p1 <- renderPlot({
    if (input$graph_type == "histogram") {
      # Histogram
      ggplot(df_country(), aes_string(x = input$continuous_variable)) +
        geom_histogram +  # histogram geom
        labs(x = input$continuos_variable, y = 'Count', title = 'Histogram') +  # labels
        facet_wrap(~prediction)    # facet by prediction
    }
    else {
      # Boxplot
      ggplot(df_country(), aes_string(y = input$continuous_variable)) +
        geom_boxplot() +  # boxplot geom
        coord_flip() +  # flip coordinates
        labs(x='Prediction', y=input$continuous_variable, title='Boxplot') +  # labels
        facet_wrap(~prediction)    # facet by prediction
    }

  })

  # TASK 6: Create logic to plot faceted bar chart or stacked bar chart
  output$p2 <- renderPlot({
    # Bar chart
    p <- ggplot(df_country(), aes_string(x = input$categorical_variable)) +
      labs(x='Count',y=input$categorical_variable,title='Bar Chart') +  # labels
      theme(element_text(angle=45),legend.position('bottom'))    # modify theme to change text
angle and legend position

    if (input$is_stacked) {
      p + geom_bar(aes_string(fill='prediction',position='stack'))  # add bar geom and use
prediction as fill
    }
    else{
      p +
        geom_bar(aes_strinig(fill=input$categorical_variable), position='stack') + # add bar geom
and use input$categorical_variables as fill
        facet_wrap(~prediction)   # facet by prediction
    }
  })

})
```