Mini Project 2

PSTAT100: Data Science Concepts and Analysis

Instructor: Ali Abuzaid

STUDENT NAME

- Phuc Lu (pdlu)
- Valerie De La Fuente (valeriedelafuente)

Instructions

- This mini project is designed to give you practical experience with real-world data using R and Shiny. You'll create an interactive web application that allows users to explore and visualize a dataset.
- Work in groups of **2 students** from the same discussion section.
- Individual submissions will not be accepted.
- Please use the provided `MP 2.qmd` file to type your Documentation and Presentation and submit it as a PDF file. You can utilize `RStudio`for this purpose. For guidance, refer to the Tutorial: Hello, Quarto).
- Please submit a `.zip` file that includes:

Your `app.R` file (fully working Shiny app).
A short project report (PDF).
Reminder: If your app fails to open or the .zip is incorrect, you will receive a score of **ZERO**. Test everything before submission.

Due Date

Due Date: Sunday, June 1, 2025, 11:59 PM

0.1 2 Tasks:

0.1.1 2.1.1: Data Loading:

```
library(tidyverse)
data(infert)
```

Variable	Description	Value
education	years of education	0–5 years, 6–11 years, 12+ years
age	age in years	22-24
parity	number of prior live births	1-6
induced	number of prior induced abortions	0, 1, 2 or more
case	case status	case, control
spontaneous	number of prior spontaneous abortions	0, 1, 2 or more
stratum	id for each matched set or group	1–83
pooled.stratum	represent a group or broader classification of strata (may be used for conditional logistic regression)	1 - 63

0.1.2 2.1.2: Data Preparation:

```
infert %>% head()
```

```
  education age parity induced case spontaneous stratum pooled.stratum
1    0-5yrs  26      6       1    1           2       1              3
2    0-5yrs  42      1       1    1           0       2              1
3    0-5yrs  39      6       2    1           0       3              4
4    0-5yrs  34      4       2    1           0       4              2
5   6-11yrs  35      3       1    1           1       5             32
6   6-11yrs  36      4       2    1           1       6             36
```

```
library(stringr)
# rename variables and remove "yrs" from education values. factorize necessary variables
infert_ds <- infert %>%
```

```r
  rename(
    Education = education,
    Age = age,
    Parity = parity,
    InducedAbortions = induced,
    SpontaneousAbortions = spontaneous,
    CaseStatus = case,
    StratumID = stratum,
    PooledStratumID = pooled.stratum
  ) %>%
  mutate(
    Education = factor(str_remove(Education, "yrs"), levels = c("0-5", "6-11", "12+")),
    CaseStatus = factor(CaseStatus, levels = c(0, 1), labels = c("control", "case")),
    Parity = factor(Parity),
    InducedAbortions = factor(InducedAbortions),
    SpontaneousAbortions = factor(SpontaneousAbortions),
    StratumID = factor(StratumID),
    PooledStratumID = factor(PooledStratumID)
  )

factor_vars <- names(infert_ds)[sapply(infert_ds, is.factor)]
numeric_vars <- names(infert_ds)[sapply(infert_ds, is.numeric)]
```

```r
summary(infert_ds)
```

```
 Education       Age        Parity InducedAbortions   CaseStatus
 0-5 : 12   Min.   :21.00   1:99   0:143            control:165
 6-11:120   1st Qu.:28.00   2:81   1: 68            case   : 83
 12+ :  0   Median :31.00   3:36   2: 37
 NA's:116   Mean   :31.50   4:18
            3rd Qu.:35.25   5: 6
            Max.   :44.00   6: 8

 SpontaneousAbortions   StratumID   PooledStratumID
 0:141                1      :  3   41     : 12
 1: 71                2      :  3   45     :  9
 2: 36                3      :  3   49     :  9
                      4      :  3   51     :  9
                      5      :  3   12     :  6
                      6      :  3   18     :  6
                      (Other):230   (Other):197
```

```r
str(infert_ds)
```

```
'data.frame':   248 obs. of  8 variables:
 $ Education           : Factor w/ 3 levels "0-5","6-11","12+": 1 1 1 1 2 2 2 2 2 2 ...
 $ Age                 : num  26 42 39 34 35 36 23 32 21 28 ...
 $ Parity              : Factor w/ 6 levels "1","2","3","4",..: 6 1 6 4 3 4 1 2 1 2 ...
 $ InducedAbortions    : Factor w/ 3 levels "0","1","2": 2 2 3 3 2 3 1 1 1 1 ...
 $ CaseStatus          : Factor w/ 2 levels "control","case": 2 2 2 2 2 2 2 2 2 2 ...
 $ SpontaneousAbortions: Factor w/ 3 levels "0","1","2": 3 1 1 1 2 2 1 1 2 1 ...
 $ StratumID           : Factor w/ 83 levels "1","2","3","4",..: 1 2 3 4 5 6 7 8 9 10 ...
 $ PooledStratumID     : Factor w/ 63 levels "1","2","3","4",..: 3 1 4 2 32 36 6 22 5 19 ...
```

0.1.3 2.2 Shiny App Development

User Interface (UI):

```r
# Define UI
ui <- fluidPage(
  titlePanel("Infertility Dataset Explorer"),

  sidebarLayout(
    sidebarPanel(
      selectInput("variable", "Select Variable:",
                  choices = names(infert_ds)),

      uiOutput("plotTypeUI"),

      conditionalPanel(
        condition = "input.plotType == 'Histogram'",
        sliderInput("bins", "Number of Bins:", min = 5, max = 50, value = 10),
        checkboxInput("showDensity", "Overlay Density Curve", value = FALSE)
      ),

      conditionalPanel(
        condition = "input.plotType == 'Boxplot'",
        selectInput("boxplotGroup", "View Boxplot Across:",
                    choices = factor_vars),
        checkboxInput("showJitter", "Add Jittered Data Points", value = FALSE)
      ),

      checkboxInput("includeNA", "Include NA in Grouped Summary", value = FALSE),
```

```
      h4("Numeric Filters"),
      sliderInput("ageFilter", "Age Range:",
                  min(infert_ds$Age), max(infert_ds$Age),
                  value = c(min(infert_ds$Age), max(infert_ds$Age))),

      selectInput("groupBy", "Group Summary By:",
                  choices = factor_vars)
    ),

    mainPanel(
      plotOutput("mainPlot", hover = "plot_hover"),
      verbatimTextOutput("hoverInfo"),
      h4("Summary"),
      verbatimTextOutput("summaryOutput"),
      h4("Grouped Summary"),
      tableOutput("groupedSummary")
    )
  )
)
```

Server Logic:

```
# Define server logic
server <- function(input, output, session) {

  # Dynamically adjust plot options
  output$plotTypeUI <- renderUI({
    req(input$variable)
    var <- input$variable

    if (is.numeric(infert_ds[[var]])) {
      selectInput("plotType", "Select Plot Type:", choices = c("Histogram", "Boxplot"))
    } else {
      selectInput("plotType", "Select Plot Type:", choices = c("Bar Chart"))
    }
  })

  # Reactive filtered dataset
  filtered_data <- reactive({
    infert_ds %>%
      filter(
```

```
      Age >= input$ageFilter[1], Age <= input$ageFilter[2]
    )
})

# Generate plots
output$mainPlot <- renderPlot({
  req(input$variable, input$plotType)
  df <- filtered_data()
  var <- input$variable

  if (input$plotType == "Histogram") {
    p <- ggplot(df, aes_string(x = var)) +
      geom_histogram(bins = input$bins, fill = "lightpink", color = "black") +
      labs(title = paste("Histogram of", var), x = var, y = "Count") +
      theme_minimal()

    if (input$showDensity) {
      p <- p + geom_density(aes(y = ..count..), color = "blue", size = 1)
    }

    p

  } else if (input$plotType == "Boxplot") {
    group_var <- input$boxplotGroup

    p <- ggplot(df, aes_string(x = group_var, y = var)) +
      geom_boxplot(fill = "royalblue") +
      labs(title = paste("Boxplot of", var, "across", group_var), x = group_var, y = var) +
      theme_minimal()

    if (input$showJitter) {
      p <- p + geom_jitter(width = 0.2, alpha = 0.4)
    }

    p

  } else if (input$plotType == "Bar Chart") {
    ggplot(df, aes_string(x = var)) +
      geom_bar(fill = "lavender", color = "black") +
      labs(title = paste("Bar Chart of", var), x = var, y = "Count") +
      theme_minimal()
  }
```

```r
  })

  # Show hover info
  output$hoverInfo <- renderPrint({
    hover <- input$plot_hover

    if (input$plotType == "Histogram" || input$plotType == "Boxplot") {
      if (!is.null(hover)) {
        cat("Hovered at:\n")
        cat("x:", round(hover$x, 2), "\n")
        cat("y:", round(hover$y, 2), "\n")
      } else {
        cat("Hover over the plot to see coordinates.")
      }
    } else {
      cat("Hover info disabled for bar charts (categorical x-axis).")
    }
  })

  # Show summary statistics
  output$summaryOutput <- renderPrint({
    req(input$variable)
    summary(filtered_data()[[input$variable]])
  })

  # Grouped summary
  output$groupedSummary <- renderTable({
    df <- filtered_data()
    group_var <- input$groupBy

    if (!input$includeNA) {
      df <- df %>% drop_na(all_of(group_var))
    }

    df %>%
      group_by(across(all_of(group_var))) %>%
      summarise(across(where(is.numeric), mean, na.rm = TRUE), .groups = "drop")
  }, rownames = TRUE)
}

# Run the app
shinyApp(ui = ui, server = server)
```

1 Documentation and Presentation

1.1 App Purpose

The purpose of this app is make it easy for the users to understand the infertility data set without having to do coding on their own. All it takes is point, click, and drag to get beautiful graphs or summary statistics of the data.

1.2 How it Works

The select variable feature allow the users to select one of eight variables from the `infert` data set to analyze. The app will automatically select a graph that works with the variable of interest. The user can also adjust the number of bins there are in the graph. If a boxplox is chosen, the user can select another variable to compare with. The user can also add jitters to the boxplot, as well include missing (`NA`) values in the boxplot. The user can alo group the summary by a certain variable as well. The grouped summary will be displayed at the very bottom of the page. On the graphs, the user can hover their mouse over the graph to view the coordinates (x, y) for more precise information. Below the graph feature are summary statistics. These include the minimum, 1st quartile, median, mean, 3rd quartile, and maximum value.

1.3 Insights

The initial questions that this data set tries to answer is what factors are associated with secondary infertility in women. Two of the main components to this questions are spontaneous versus induced abortions. Spontaneous abortions is abortion due to natural causes, while induced abortion is to deliberately terminate the pregnancy. From the data gathered in this data set, the distribution of spontaneous and induced abortions are similar, in that the majority of women in this data reported to having 0 of either types of abortions. On the other hand, about 1/3 of the total women in the data set reported having 2 abortions in the respective type of abortion.

It's valuable look at the distribution of the age of women in this data set. The surveyed women are from 20 to 45 years old. The distribution of the age variable has center at around 28-30 years old and follow as a bell shape with heavier tail on both sides. Another interesting variable is parity, which is the number of prior live births. This variable has a negative slope where the majority of women reported to having 1 live birth and as the number of live birth increase, the number of women reported having said number of live births decreases.

1.4 Reflections

Since one person has more experienced with Rshiny, rather than having the person learn RShiny entirely, we decided to just let the RShiny expert person code the app with all of its features, while the less experienced person handle other tasks. The second person is very confident in writing, so they're tasked with writing and testing the app. Their job is to tinker with the app to figure out and document how the app works, how easy the app is to use, and to catch any potential app quirks from the user experience. This division of tasks is more time efficient, minimizes friction, and lets both people focus on tasks that best suits their expertise.

As of now, this app cannot compare two variables to learn about their relationships. This feature can be implemented in a future version of this app.