APP-4VARs-SELECTION

P.V. (SUNDAR) Balakrishnan

2023-04-10

COMMENT CODE: LIBRARIES LOAD

```
# Load the shiny package
library(shiny)

# Load kableExtra package for advanced table formatting
library(kableExtra)

# Load tidyverse package for data wrangling and visualization
library(tidyverse)

# Load the modelsummary package for easy and elegant model output summarization
library(modelsummary)

# Load the corrplot package for easy correlation plot creation
library(corrplot)

# Load the correlation package for advanced correlation analysis
library(correlation)

# Load the psych package for psychological research tools
library(psych)

# Load the report package for creating HTML or PDF reports
library(report)

# Load the easystats package for common statistical models and tests
library(easystats)
```

COMMENT UI.r

```
# Define the user interface for the shiny app
ui <- fluidPage(

     # Add a title panel to the UI
     titlePanel('Interactive Variable Selection'),

     # Define the layout of the sidebar and main panels
     sidebarLayout(
          # Define the sidebar panel
          sidebarPanel(

               # Add a heading to the sidebar panel
               h3('Select your Variables'),

               # Add an action button to allow users to select variables
               actionButton('select_vars', 'Select Variables')
          ),

          # Define the main panel
          mainPanel(

               # Add a heading and a space to display summary of the selected variables
               h3('Summary'),
               verbatimTextOutput('summary'),

               # Add a heading and a space to display the correlation test of the selected variables
               h3('Correlation Test'),
               verbatimTextOutput('corr_test'),

               # Add a heading and a space to display a report
               h3('Report'),
               verbatimTextOutput('report')
          )
     )
)
```

COMMENT server.r

```
# This is a server function for a Shiny app.
#
# The `iris` dataset is loaded, and then its column names are renamed in the `iris2` data frame.
#
# `selected_vars` is a reactive value used for storing user-selected variables.
#
# `observeEvent` is used to create a pop-up window with a list of variable names from the `iris2` data 
#
# There are three `output` functions:
# - `output$summary` prints the summary of the `iris2` data frame including only selected variables.
# - `output$corr_test` prints the results of the correlation test among the selected variables of `iris
# - `output$report` prints a report of the data including global information about the variables and th
#
# All of these functions require that the user has selected variables, as they call `req(selected_vars(

The given R code defines a Shiny server function that provides a reactive interface to display summary 

```
library(shiny)
library(dplyr)
```
```

```
library(psych)
library(flextable)
library(rmarkdown)

server <- function(input, output, session) {

  # Reactive variable to store selected variables
  selected_vars <- reactiveVal()

  # Load iris dataset and rename columns
  iris2 <- iris %>%
    rename(SW = Sepal.Width,
           PL = Petal.Length,
           PW = Petal.Width)

  # Select variables using select.list and store in reactive value
  observeEvent(input$select_vars, {
    vars <- select.list(names(iris2), multiple = TRUE,
                        title = 'Select your Variables',
                        graphics = TRUE)
    selected_vars(vars)
  })

  # Render summary statistics for selected variables
  output$summary <- renderPrint({
    req(selected_vars())
    newiris <- iris2[, c(selected_vars())]
    summary(newiris)
  })

  # Render correlation test for selected variables
  output$corr_test <- renderPrint({
    req(selected_vars())
    newiris <- iris2[, c(selected_vars())]
    corr.test(newiris)
  })

  # Render formatted report for selected variables
  output$report <- renderPrint({
    req(selected_vars())
    newiris <- iris2[, c(selected_vars())]
    ft <- flextable(summary(newiris))
    report <- rmarkdown::render(paste0("### Summary statistics for selected variables\n",
                                       as.character(ft)))
    cat(report)
  })

}
```

This code loads the necessary libraries and defines a Shiny server function that uses reactive variable

SHINYAPP EXECUTE

```
# The shinyApp function constructs and launches a Shiny web application.
# The 'ui' argument is the user interface (UI) for the web app.
# The 'server' argument is the server-side logic for the web app.
# The two arguments are passed into the shinyApp function to create and #launch the app.

shinyApp(ui = ui, server = server)
```

# This code represents the creation and launching of a Shiny web application. ‘ui‘ refers to the user interface of the app, and ‘server‘ refers to the logic that handles user input and produces output for the interface. This code should be used at the end of the script after both ‘ui‘ and ‘server‘ have been defined.