Shiny App User Guidelines

A shiny app was created to visualize the spectrums of different oil types for various concentration levels. This document consists of the guidelines to follow, when using the app.

1. Shiny App code

```
##
app.R
##


        library(shiny)
        library(shinydashboard)
        library(ggplot2)
        library(plotly)
        library(tidyverse)

        oilData <- read_csv("Tidy Data.csv")

        #oilData$`Palm olein concentration(C)` <- as.factor(oilData$`Palm olein
        concentration(C)`)
        # oilData$`Replicate No` <- as.factor(oilData$`Replicate No`)
        oilData <- rename(oilData, Concentration = `Palm olein concentration(C)`,
                          Replicate = `Replicate No`)



        ui <- dashboardPage(
          dashboardHeader(title= h3("Spectrum Viz")),
          dashboardSidebar(
            selectInput("Series", label = h4("Select Series"),
                        choices = unique(oilData$Series), multiple = TRUE),
            selectInput("Concentration", label = h4("Select Palm Olein Concentration"),
                        choices = NULL),
            selectInput("Replicate", label = h4("Select Replicate"),
                        choices = NULL)
          ),
          dashboardBody(
            fluidRow(
              box(plotlyOutput("plot", height = 570), width = 12)
            )

          )
        )
```

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

  # updating filters
  series <- reactive({
    req(input$Series)
    filter(oilData, Series == input$Series)
  })

  concentration <- reactive({
    req(input$Concentration)
    filter(series(), Concentration == input$Concentration)
  })

  replicate <- reactive({
    req(input$Replicate)
    filter(concentration(), Replicate == input$Replicate)
  })

  # observing event to update next filter
  observeEvent(series(), {
    updateSelectInput(session, "Concentration",
                      choices = sort(unique(series()$Concentration)),
                      selected = c("Pure Palm Oil","Pure VCO"))
  })

  observeEvent(concentration(), {
    updateSelectInput(session, "Replicate",
                      choices = unique(concentration()$Replicate), selected = 1)
  })



  output$plot <- renderPlotly({
    plot <- replicate() %>%
      ggplot(aes(x = `Wave Number (cm-1)(W)`, y = `Absorption (A)`)) +
      geom_line() +
      theme_bw() +
      labs(x = "Wave Number (cm-1)", y = "Absorption", title = paste("Spectrum of",
input$Series,
                                                                     "For
Replicate", input$Replicate))
    ggplotly(plot)
```

```
  })
}

shinyApp(ui, server)
```

2. Guidelines for using the app

- Save the code in the project folder using the name “app.R”.
- Include the dataset needed for the app in the same folder.
- Rename the data set using the name “Tidy Data.csv” (See yellow highlighted line of code in Section 1).
- The data set needed to run the app should be arranged according to the structure shown in Figure 2.1.

Series	Palm olein concentration(C)	Replicate No	Wave Number (cm-1)(W)	Absorption(A)
Pure Palm Oil	1	1	5500	0.0017
Pure Palm Oil	1	1	5499	0.0017
Pure Palm Oil	1	1	5498	0.0018
.	.	.	.	.
.	.	.	.	.
.	.	.	.	.
Pure Palm Oil	0	15	5499	0.0017
Pure Palm Oil	0	15	5498	0.0018

Figure 2. 1 Data structure for Shiny app

- Open the app.R script and click “Run app”.

- The app will not show as soon as you run it. Figure 2.2 shows the initial view of the app.

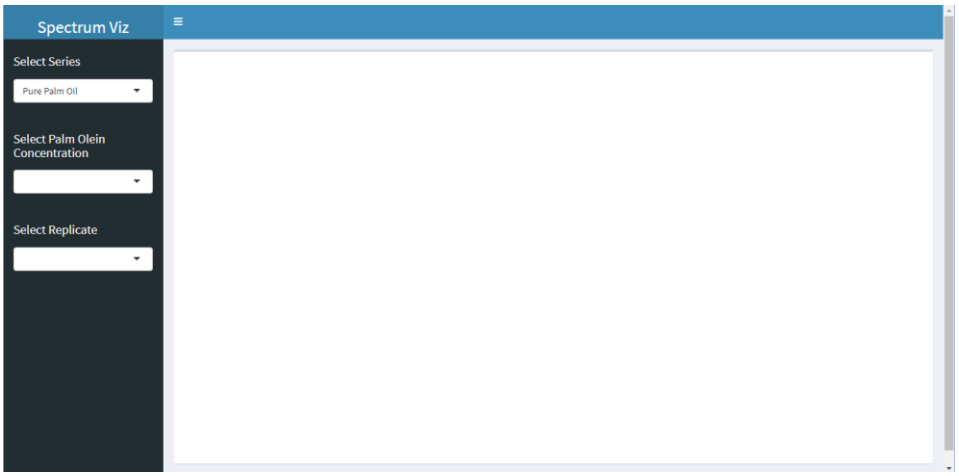

Figure 2. 2 Initial view of app

- Select the palm oil concentration from the given filters on the left, to see visualizations. Figure 2.3 shows the view of one such visualization.

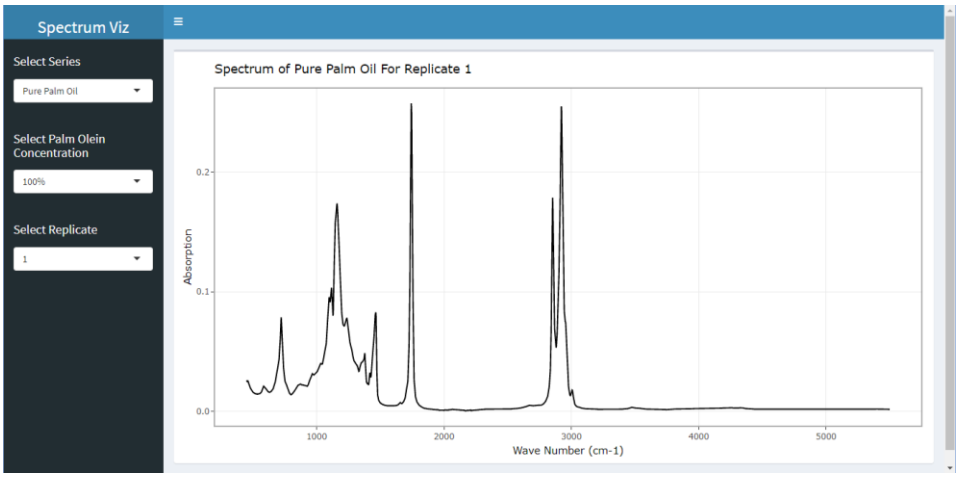

Figure 2. 3 App view with filter selections

- The 3 filters on the left as shown in Figure 2. 3 allows the user to visualize various spectrums as per their liking.