Written Assignment Unit 4

Hang Tuan Thien

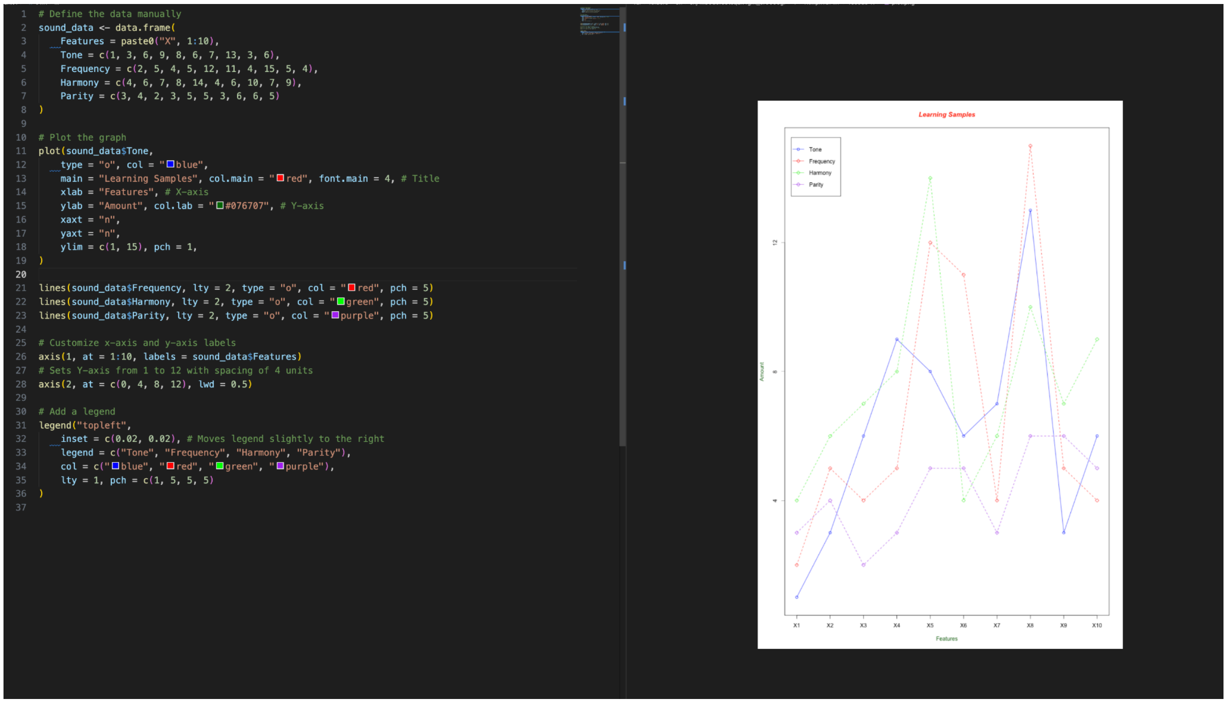
MSIT 5226 - Foundations of Machine Learning

Dr. Zaharaddeen Sani

February 25, 2025

In this written assignment, I will imply how to read data from a text file in R. The program will plot the Voice Recognition project's learning samples on a Line Chart, save the output as a PNG file, and later modify the program to read from an external data file instead of a manually coded table.

Step 1: Drawing the Line Chart in R

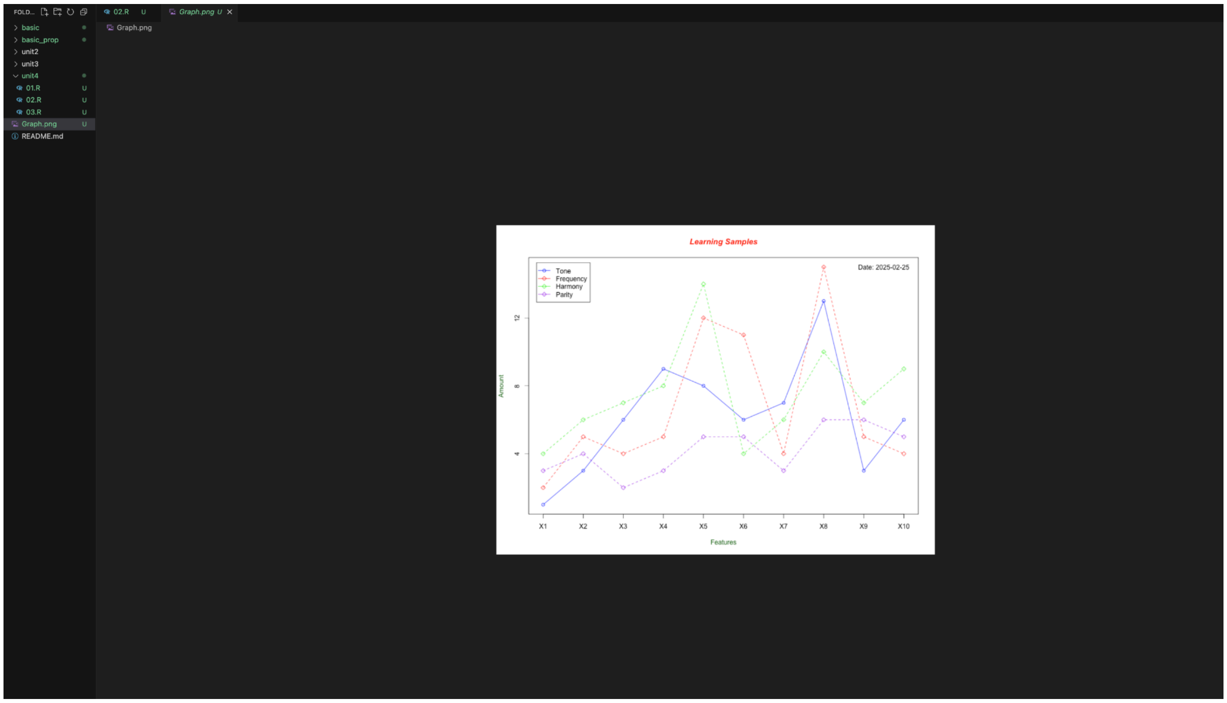


```
# Define the data manually
sound_data <- data.frame(
    Features = paste0("X", 1:10),
    Tone = c(1, 3, 6, 9, 8, 6, 7, 13, 3, 6),
    Frequency = c(2, 5, 4, 5, 12, 11, 4, 15, 5, 4),
    Harmony = c(4, 6, 7, 8, 14, 4, 6, 10, 7, 9),
    Parity = c(3, 4, 2, 3, 5, 5, 3, 6, 6, 5)
)

# Plot the graph
plot(sound_data$Tone,
    type = "o", col = "🟦blue",
    main = "Learning Samples", col.main = "🟥red", font.main = 4, # Title
    xlab = "Features", # X-axis
    ylab = "Amount", col.lab = "🟩#076707", # Y-axis
    xaxt = "n",
    yaxt = "n",
    ylim = c(1, 15), pch = 1,
)

lines(sound_data$Frequency, lty = 2, type = "o", col = "🟥red", pch = 5)
lines(sound_data$Harmony, lty = 2, type = "o", col = "🟩green", pch = 5)
lines(sound_data$Parity, lty = 2, type = "o", col = "🟪purple", pch = 5)

# Customize x-axis and y-axis labels
axis(1, at = 1:10, labels = sound_data$Features)
# Sets Y-axis from 1 to 12 with spacing of 4 units
axis(2, at = c(0, 4, 8, 12), lwd = 0.5)

# Add a legend
legend("topleft",
    inset = c(0.02, 0.02), # Moves legend slightly to the right
    legend = c("Tone", "Frequency", "Harmony", "Parity"),
    col = c("🟦blue", "🟥red", "🟩green", "🟪purple"),
    lty = 1, pch = c(1, 5, 5, 5)
)
```

```
# Define the data manually

sound_data <- data.frame(

    Features = paste0("X", 1:10),

    Tone = c(1, 3, 6, 9, 8, 6, 7, 13, 3, 6),

    Frequency = c(2, 5, 4, 5, 12, 11, 4, 15, 5, 4),

    Harmony = c(4, 6, 7, 8, 14, 4, 6, 10, 7, 9),
    Parity = c(3, 4, 2, 3, 5, 5, 3, 6, 6, 5)
)


# Plot the graph
plot(sound_data$Tone,
    type = "o", col = "blue",
    main = "Learning Samples", col.main = "red", font.main = 4, # Title
    xlab = "Features", # X-axis
    ylab = "Amount", col.lab = "#076707", # Y-axis
    xaxt = "n",
    yaxt = "n",
    ylim = c(1, 15), pch = 1,
)


lines(sound_data$Frequency, lty = 2, type = "o", col = "red", pch = 5)
lines(sound_data$Harmony, lty = 2, type = "o", col = "green", pch = 5)
lines(sound_data$Parity, lty = 2, type = "o", col = "purple", pch = 5)


# Customize x-axis and y-axis labels
axis(1, at = 1:10, labels = sound_data$Features)
# Sets Y-axis from 1 to 12 with spacing of 4 units
axis(2, at = c(0, 4, 8, 12), lwd = 0.5)
```

```
# Add a legend

legend("topleft",

    inset = c(0.02, 0.02), # Moves legend slightly to the right

    legend = c("Tone", "Frequency", "Harmony", "Parity"),

    col = c("blue", "red", "green", "purple"),

    lty = 1, pch = c(1, 5, 5, 5)

)
```

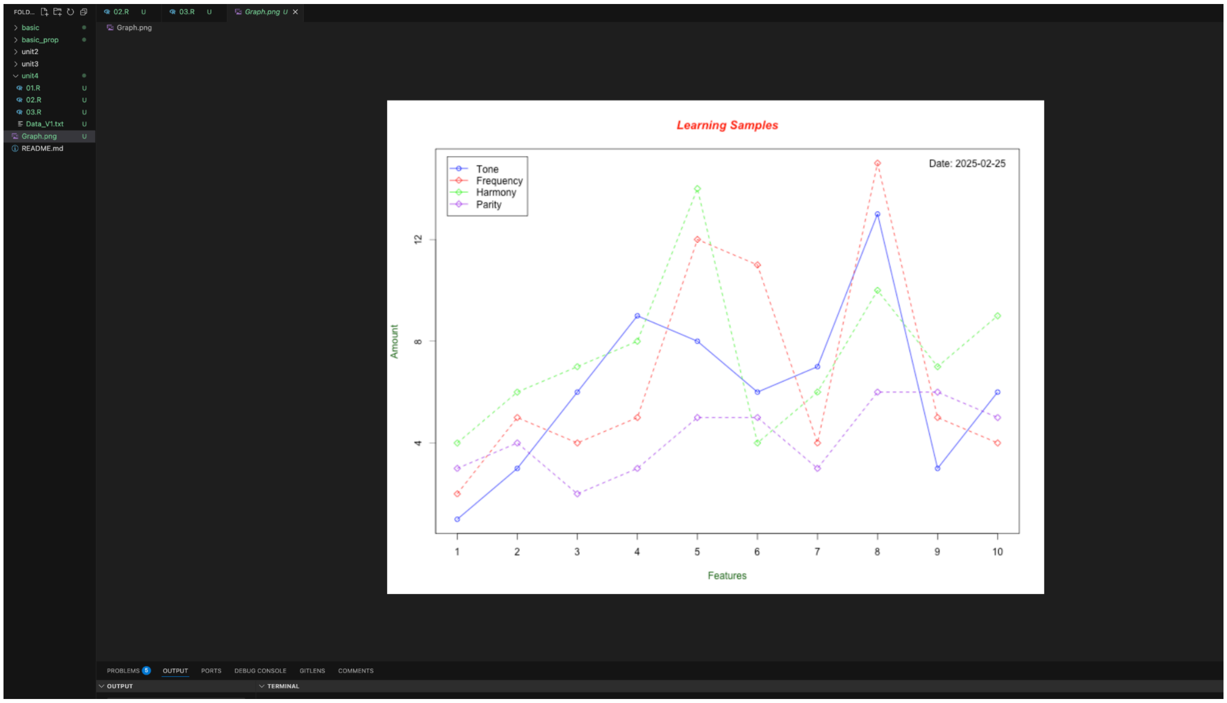
Step 2: Save the output of your program in a graphical file or PNG format



```
# Define the data manually
```

```
sound_data <- data.frame(
    Features = paste0("X", 1:10),
    Tone = c(1, 3, 6, 9, 8, 6, 7, 13, 3, 6),
    Frequency = c(2, 5, 4, 5, 12, 11, 4, 15, 5, 4),
    Harmony = c(4, 6, 7, 8, 14, 4, 6, 10, 7, 9),
    Parity = c(3, 4, 2, 3, 5, 5, 3, 6, 6, 5)
)

# Save the output as a PNG file
png(filename = "Graph.png", height = 600, width = 800, bg = "white")

# Plot the graph
plot(sound_data$Tone,
    type = "o", col = "blue",
    main = "Learning Samples", col.main = "red", font.main = 4, # Title
    xlab = "Features", # X-axis
    ylab = "Amount", col.lab = "#076707", # Y-axis
    xaxt = "n",
    yaxt = "n",
    ylim = c(1, 15), pch = 1,
)

lines(sound_data$Frequency, lty = 2, type = "o", col = "red", pch = 5)
lines(sound_data$Harmony, lty = 2, type = "o", col = "green", pch = 5)
lines(sound_data$Parity, lty = 2, type = "o", col = "purple", pch = 5)

# Customize x-axis and y-axis labels
axis(1, at = 1:10, labels = sound_data$Features)
# Sets Y-axis from 1 to 12 with spacing of 4 units
axis(2, at = c(0, 4, 8, 12), lwd = 0.5)

# Add a legend
legend("topleft",
    inset = c(0.02, 0.02), # Moves legend slightly to the right
    legend = c("Tone", "Frequency", "Harmony", "Parity"),
    col = c("blue", "red", "green", "purple"),
    lty = 1, pch = c(1, 5, 5, 5)
)

# Add the current date to the graph
text(9.5, 15, paste("Date:", Sys.Date()), col = "black", cex = 1)

# Close the PNG device to save the file
dev.off()
```

Step 3: Read the same data from a text file



```
# Define the data manually

sound_data <- read.table("unit4/Data_V1.txt", header = TRUE, sep = "\t")



# Save the output as a PNG file

png(filename = "Graph.png", height = 600, width = 800, bg = "white")



# Plot the graph

plot(sound_data$Tone,

    type = "o", col = "blue",

    main = "Learning Samples", col.main = "red", font.main = 4, # Title

    xlab = "Features", # X-axis
    ylab = "Amount", col.lab = "#076707", # Y-axis
    xaxt = "n",
    yaxt = "n",
    ylim = c(1, 15), pch = 1,
)

lines(sound_data$Frequency, lty = 2, type = "o", col = "red", pch = 5)
lines(sound_data$Harmony, lty = 2, type = "o", col = "green", pch = 5)
lines(sound_data$Parity, lty = 2, type = "o", col = "purple", pch = 5)

# Customize x-axis and y-axis labels
axis(1, at = 1:10, labels = sound_data$Features)
# Sets Y-axis from 1 to 12 with spacing of 4 units
axis(2, at = c(0, 4, 8, 12), lwd = 0.5)

# Add a legend
legend("topleft",
    inset = c(0.02, 0.02), # Moves legend slightly to the right
    legend = c("Tone", "Frequency", "Harmony", "Parity"),
    col = c("blue", "red", "green", "purple"),
    lty = 1, pch = c(1, 5, 5, 5)
)
```

```
# Add the current date to the graph

text(9.5, 15, paste("Date:", Sys.Date()), col = "black", cex = 1)



# Close the PNG device to save the file

dev.off()
```

In this assignment, I imply how to plot Voice Recognition project data in R using both manually defined and external text file data. The graph is saved as a PNG file, and the current date is added to the output, showcasing a flexible approach for visualizing and saving project data.

References

GeeksforGeeks. (n.d.). *Remove axis values of plot in base R. GeeksforGeeks.* https://www.geeksforgeeks.org/remove-axis-values-of-plot-in-base-r/

GeeksforGeeks. (n.d.). *Saving graphs as files in R. GeeksforGeeks.* https://www.geeksforgeeks.org/saving-graphs-as-files-in-r/

W3Schools. (n.d.). R graph plot. W3Schools. https://www.w3schools.com/r/r_graph_plot.asp