```r
#Loading all the necessary libraries to create table one in latex
library(tableone)
library(knitr)
library(kableExtra)
library(xtable)


# for data manipulation/cleaning
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following object is masked from 'package:kableExtra':
##
##     group_rows

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```r
# Reading the gene data from file

#library(readxl)

df <- read.csv("merged_data.csv", stringsAsFactors = FALSE)
```

```r
# Defining variable groups: 3 categorical and 3 continuous variables.
#Variable to stratify with is sex

cont_vars <- c("age", "ferritin.ng.ml.", "charlson_score")
cat_vars  <- c("sex", "icu_status", "disease_status")
strat_var <- "sex"


# Ensure categorical variables are factors
for (var in cat_vars) {
  df[[var]] <- as.factor(df[[var]])
}

# Get stratification levels
strata_levels <- levels(df[[strat_var]])

# Initialize empty list to hold table rows
table_rows <- list()


# CATEGORICAL VARIABLES
```

```r
# Getting factor levels for each categorical variable using a for loop
#and then I created a row label based on each factor level
for (var in cat_vars) {
  levels_var <- levels(df[[var]])
  for (lvl in levels_var) {
    row_name <- paste(var, lvl, sep = ": ")
    row_values <- c()
    for (group in strata_levels) {
      subset_df <- df[df[[strat_var]] == group, ]
#setting n as he count of observations for each levl
#total is for the total values that are present or not missing from that variable
      n <- sum(subset_df[[var]] == lvl, na.rm = TRUE)
      total <- sum(!is.na(subset_df[[var]]))

#Then i calculated the percentages and rounded them and appended it ot the row in the table
      pct <- if (total > 0) round((n / total) * 100, 1) else NA
      value <- paste0(n, " (", pct, "%)")
      row_values <- c(row_values, value)
    }
    table_rows[[length(table_rows) + 1]] <- c(row_name, row_values)
  }
}


# COUNTINUOUS VARIABLES

#Making sure to remove any unkoum values that may be mentioned as
#unknown/NA/na because i was getting an error without running this
#I also checked the values in these columns with unique() to make
#sure all the unknowns were removed

df$age[df$age %in% c("unknown", "NA", "n/a", "", "--")] <- NA
df$ferritin.ng.ml.[df$ferritin.ng.ml. %in% c("unknown", "NA", "n/a", "", "--")] <- NA
df$charlson_score[df$charlson_score %in% c("unknown", "NA", "n/a", "", "--")] <- NA

#Then I converted the continuous variables to numeric in case they were stored as strings
df$age <- as.numeric(as.character(df$age))
```

```
## Warning: NAs introduced by coercion
```

```r
df$ferritin.ng.ml. <- as.numeric(as.character(df$ferritin.ng.ml.))
```

```
## Warning: NAs introduced by coercion
```

```r
df$charlson_score <- as.numeric(as.character(df$charlson_score))

#Using a for loop to create a row label for each variable

for (var in cont_vars) {
  row_name <- paste(var, "(mean (sd))")
  row_values <- c()
```

Table 1: Table 1: Summary Statistics Stratified by Sex

Variable	female	male	unknown
sex: female	51 (100%)	0 (0%)	0 (0%)
sex: male	0 (0%)	73 (100%)	0 (0%)
sex: unknown	0 (0%)	0 (0%)	1 (100%)
icu_status: no	27 (52.9%)	32 (43.8%)	0 (0%)
icu_status: yes	24 (47.1%)	41 (56.2%)	1 (100%)
disease_status: disease state: COVID-19	38 (74.5%)	61 (83.6%)	0 (0%)
disease_status: disease state: non-COVID-19	13 (25.5%)	12 (16.4%)	1 (100%)
age (mean (sd))	59.3 (17.9)	62.3 (14.4)	83 (NA)
ferritin.ng.ml. (mean (sd))	619.3 (1054.3)	1000.3 (1019.8)	NaN (NA)
charlson_score (mean (sd))	3.6 (2.5)	3.4 (2.5)	8 (NA)

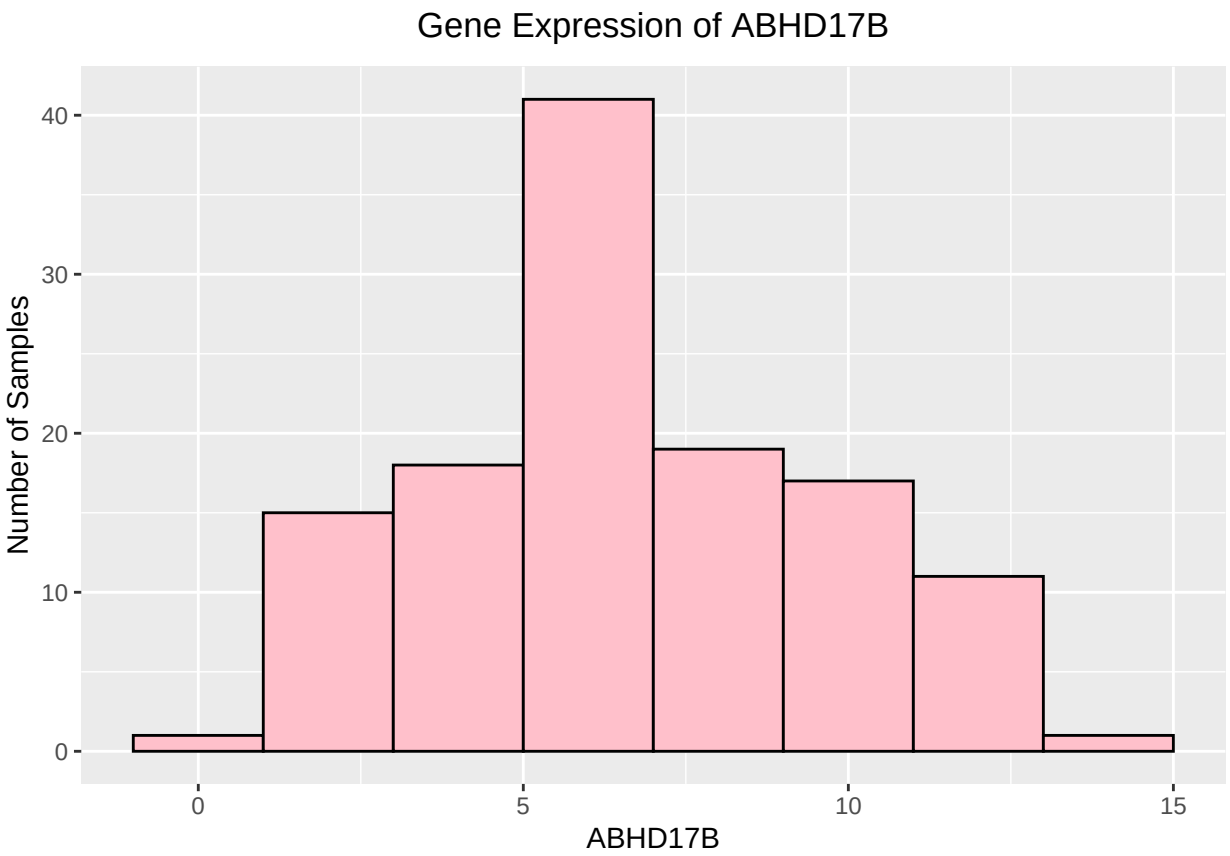
```
#I subset the dataset for the variable i want to stratify with and then extracted it's value
  for (group in strata_levels) {
    subset_df <- df[df[[strat_var]] == group, ]
    values <- subset_df[[var]]

#calculated mean and sd for all the values and rounded them off to 1 deciml place
    mean_val <- round(mean(values, na.rm = TRUE), 1)
    sd_val <- round(sd(values, na.rm = TRUE), 1)
#Made a string by combining the mean and sd so that they appear together in the table
    summary_str <- paste0(mean_val, " (", sd_val, ")")
    row_values <- c(row_values, summary_str)
  }
#After looping through all groups for this variable i store the row in table_rows.
  table_rows[[length(table_rows) + 1]] <- c(row_name, row_values)
}


# Convert to data frame

table_matrix <- do.call(rbind, table_rows)
table_df <- as.data.frame(table_matrix, stringsAsFactors = FALSE)

# Add column names
colnames(table_df) <- c("Variable", strata_levels)


# Printed as Latex table

kable(table_df,
      format = "latex", booktabs = TRUE, caption = "Table 1: Summary Statistics Stratified by Sex")
```

```
#Generate final a publication quality  histogram, scatter plot, and boxplot from submission 1

# Selecting a gene and 3 covariates

genes<- read.csv("QBS103_GSE157103_genes.csv", header = TRUE)
```

```r
# Read participant data from file
matrix<- read.csv("QBS103_GSE157103_series_matrix-1.csv", header = TRUE)

# Transposing the gene data to make rows participant data
genes_trans <- as.data.frame(t(genes))
colnames(genes_trans) <- as.character(genes_trans[1,])

#remove the first row since it's no longer data, just headers after transposing
genes_trans <- genes_trans[-1,]

# Merge gene data with participant data
genes_trans$participant_id <- rownames(genes_trans)
rownames(matrix)<- matrix$participant_id
merged_df <- merge(genes_trans, matrix , by = "row.names")
colnames(merged_df)[1] <- "participant_id"

selected_cov <- c("age", "sex", "disease_status", "ABHD17B")

subset_merged <- merged_df[,selected_cov]
subset_merged$age <- as.numeric(subset_merged$age)
```

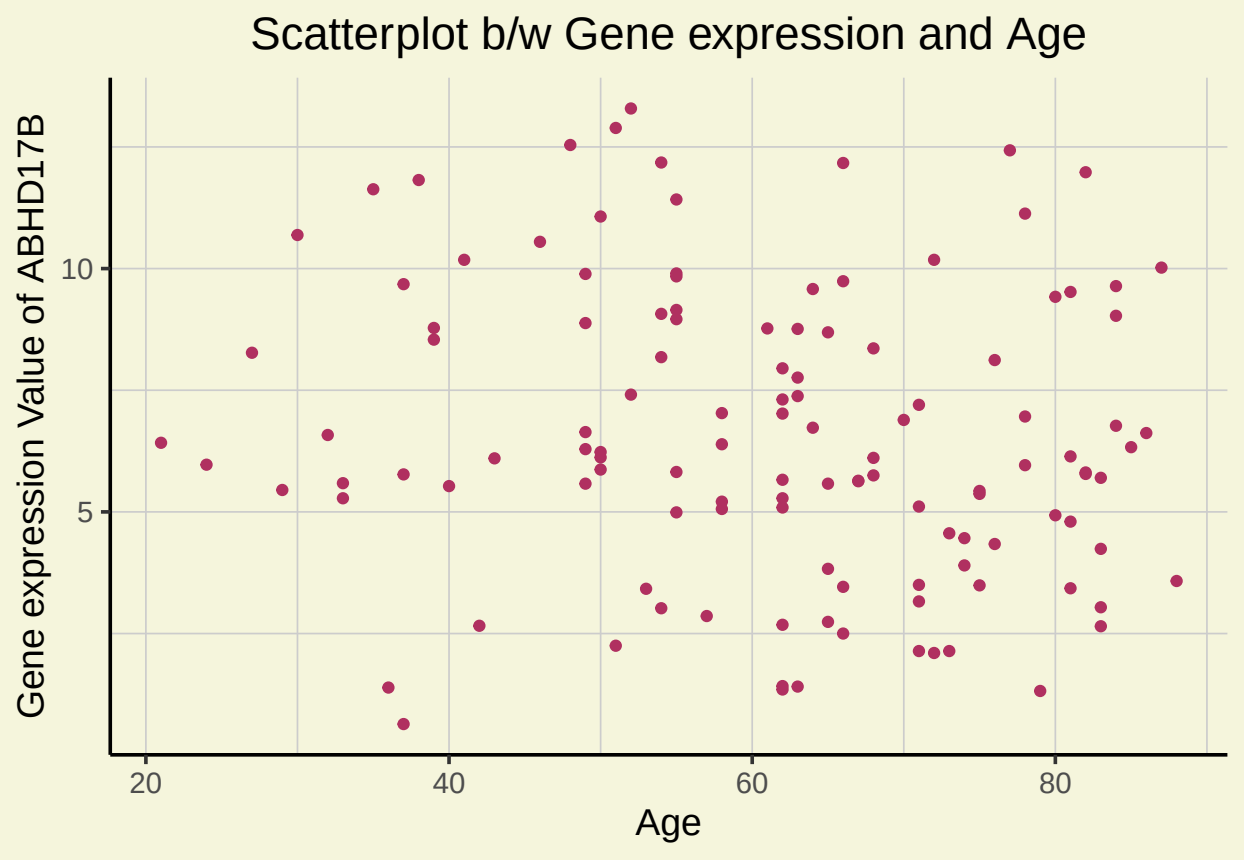
```
## Warning: NAs introduced by coercion
```

```r
subset_merged$ABHD17B <- as.numeric(subset_merged$ABHD17B)
subset_merged <- na.omit(subset_merged)

library(ggplot2)
#To make sure the ABHD17B column is numeric.
subset_merged$ABHD17B <- as.numeric(subset_merged$ABHD17B)

ggplot(subset_merged, aes(x = ABHD17B)) +
  #each bar covers 2 units of ABHD17B values, adds
  #the colour as pink and makes the outline black

  geom_histogram(binwidth = 2.0, fill = "pink", color = "black") +
  #Title and axis labels
  labs(title = "Gene Expression of ABHD17B", x = "ABHD17B", y = "Number of Samples")+
  #This adjusts the position of the plot title
  theme(plot.title = element_text(hjust = 0.5, vjust = 2))
```

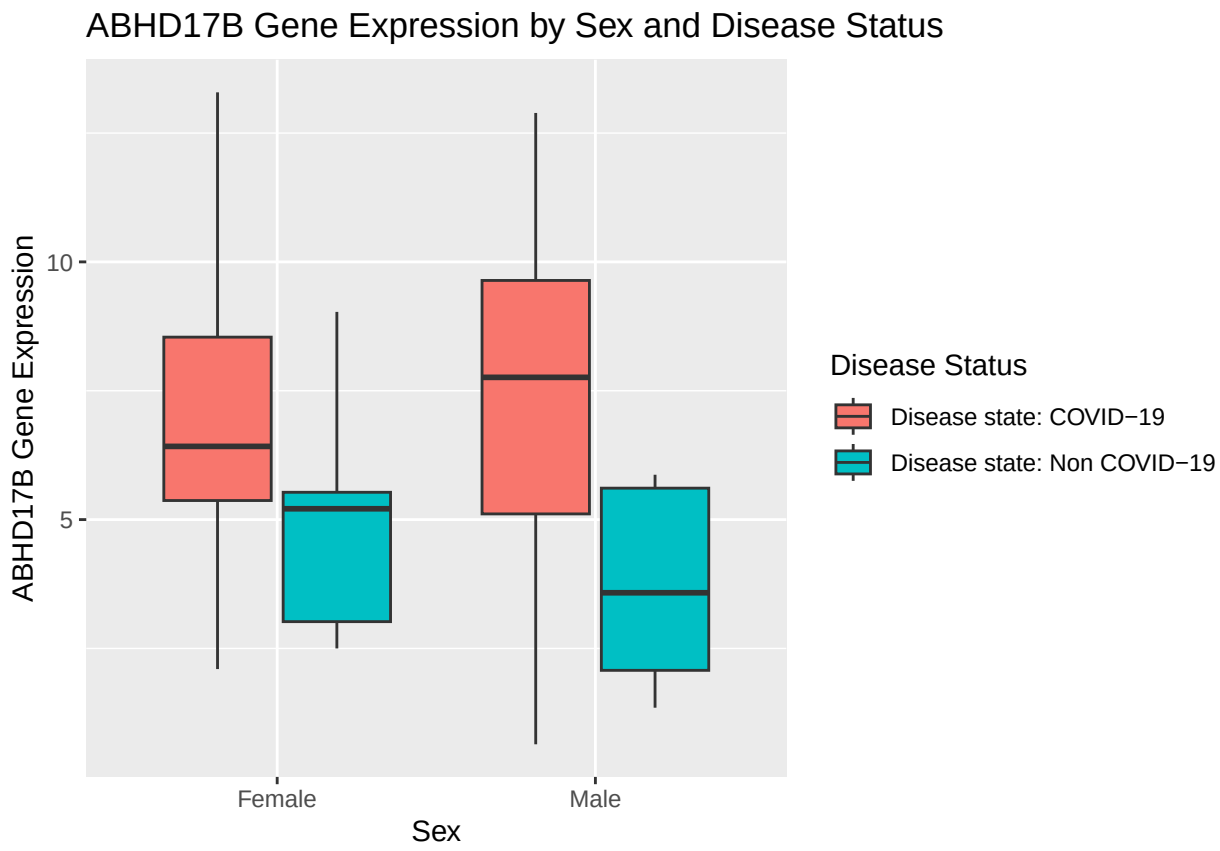



```
ggplot(subset_merged, aes(x = age, y = ABHD17B)) +
  geom_point(color = 'maroon') +
  labs(title = "Scatterplot b/w Gene expression and Age",
       y = "Gene expression Value of ABHD17B", x = "Age")+
        theme_classic(base_family = 'sans',base_size = 14) +
  theme(panel.border = element_blank(),
        plot.title = element_text(hjust = 0.5),
        panel.grid.major = element_line(color = "gray80", size = 0.3),
        panel.grid.minor = element_line(color = "gray80", size = 0.3),
        # Define my axis
        axis.line = element_line(colour = "black", linewidth = rel(1)),
        # Set plot background
        plot.background = element_rect(fill = "beige"),
        panel.background = element_blank())
```

```
## Warning: The ‘size‘ argument of ‘element_line()‘ is deprecated as of ggplot2 3.4.0.
## i Please use the ‘linewidth‘ argument instead.
## This warning is displayed once every 8 hours.
## Call ‘lifecycle::last_lifecycle_warnings()‘ to see where this warning was
## generated.
```

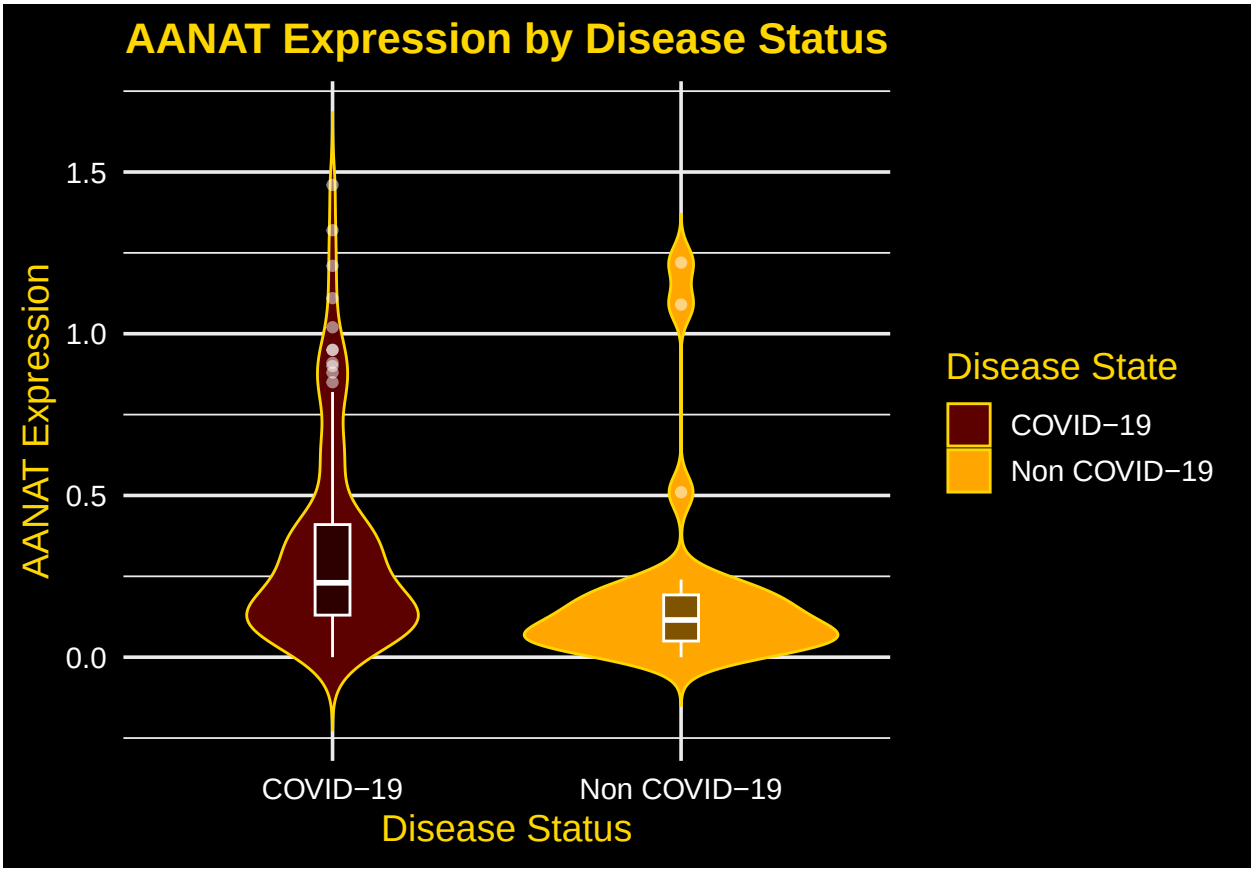



```r
library(ggplot2)
#Removed the unknowns in the sex column and used the cleaned dataset to plot
subset_merged_clean <- subset_merged[subset_merged$sex != " unknown", ]

ggplot(subset_merged_clean, aes(x = sex, y = ABHD17B, fill = disease_status)) +
  geom_boxplot() +
  labs(
    title = "ABHD17B Gene Expression by Sex and Disease Status",
    x = "Sex",
    y = "ABHD17B Gene Expression",
    fill = "Disease Status"
  ) +
  # Change disease status labels in legend to capitalize tags
  scale_fill_discrete(labels = c("Disease state: COVID-19", "Disease state: Non COVID-19")) +
  # Changed x axis labels to capitalize
  scale_x_discrete(labels = c(" male" = "Male", " female" = "Female")) +
  # Style legend by settting position and background
  theme(
    legend.key = element_rect(fill = 'white'),
    legend.position = 'right'
  )
```




```
library(harrypotter)
ggplot(df, aes(x = disease_status, y = AANAT, fill = disease_status)) +

  # Add violin shape to show distribution
  geom_violin(trim = FALSE, color = "gold") +

  # Overlay boxplot for median & quartiles
  geom_boxplot(width = 0.1, fill = "black", color = "white", alpha = 0.5) +

  # Apply Gryffindor colors and rename legend entries
  scale_fill_hp(
    discrete = TRUE, option = "Gryffindor",
    labels = c("COVID-19", "Non COVID-19")
  ) +

  # Rename x axis categories
  scale_x_discrete(
    labels = c("COVID-19", "Non COVID-19")
  ) +

  # Titles and labels
  labs(
    title = "AANAT Expression by Disease Status",
    x = "Disease Status",
    y = "AANAT Expression",
    fill = "Disease State"
```

```
  ) +

  # Theme customization
  theme_minimal(base_size = 14) +
  theme(
    legend.position = "right",
    legend.background = element_rect(fill = "black"),
    legend.text = element_text(color = "white"),
    legend.title = element_text(color = "gold"),

    plot.background = element_rect(fill = "black", color = NA),
    panel.background = element_rect(fill = "black"),

    axis.title = element_text(color = "gold"),
    axis.text = element_text(color = "white"),

    plot.title = element_text(color = "gold", face = "bold", size = 16, hjust = 0.5)
  )
```




```
library(pheatmap)
library(RColorBrewer)

# Replace "unknown" with NA
df[df == "unknown"] <- NA
```

```r
# Clean up sex column to remove spaces and also make ot title case for the map
df$sex <- trimws(df$sex)
df$sex[df$sex == "male"] <- "Male"
df$sex[df$sex == "female"] <- "Female"
df$sex[df$sex == "unknown"] <- "Unknown"
df$mechanical_ventilation[df$mechanical_ventilation == ' yes'] <- "Yes"
df$mechanical_ventilation[df$mechanical_ventilation == ' no'] <- "No"


# Select genes
selected_genes <- c("A1BG","ABHD16A","A2M","ABHD17A","A3GALT2",
                    "ABHD14B","ABHD12","AAAS","AACS","AANAT")

# Extract gene data
heat_data <- as.data.frame(df[, selected_genes])
heat_data <- apply(heat_data, 2, as.numeric)

# Log2 transformed the data (My classmate Rhea Sarmah helped me figure this step out!)
logged_data <- log2(heat_data + 1)

# Add participant IDs as rownames
rownames(logged_data) <- df$participant_id

# Annotation (samples = rows)
annotation <- data.frame(
  Ventilation = df$mechanical_ventilation,
  Sex = df$sex
)
rownames(annotation) <- df$participant_id



# Annotation colours
ann_colors <- list(
  Ventilation = c("Yes" = "pink", "No" = "orchid"),
  Sex =c("Male" = "blue", "Female" = "orange", "Unknown" = "gray")
)

# Define color gradient for the heatmap cells
cell_colors <- colorRampPalette(rev(brewer.pal(n = 11, name = "RdBu")))(100)

# Heatmap
pheatmap(logged_data,
         color = cell_colors,
         clustering_distance_rows = "euclidean",
         clustering_distance_cols = "euclidean",
         clustering_method = "complete",
         annotation_row = annotation,
         annotation_colors = ann_colors,
         show_rownames = FALSE,
         show_colnames = TRUE,
         fontsize_col = 10,
         fontsize_row = 4,
```

```
        main = "Gene Expression Heatmap")
```