R Final Submission

Eric Wang

2024-08-26

```
options(warn = -1)
# Load necessary libraries
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.5
## v forcats   1.0.0     v stringr   1.5.1
## v ggplot2   3.5.1     v tibble    3.2.1
## v lubridate 1.9.3     v tidyr     1.3.1
## v purrr     1.0.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

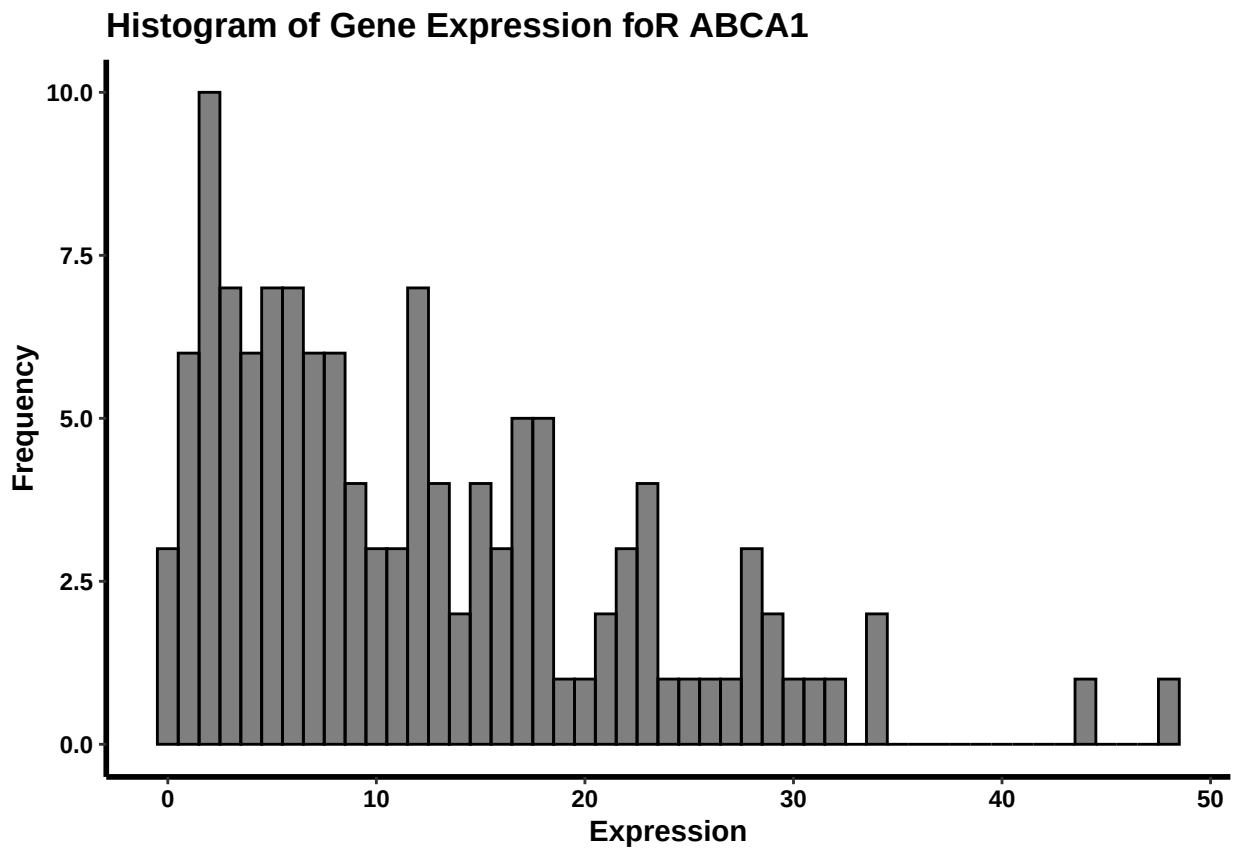
```
library(knitr)
library(kableExtra)
```

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

```
library(pheatmap)
library(ggplot2)

# Load datasets from specified paths
df1 <- read.csv("/Users/wangdeyao/Downloads/QBS103/QBS103_GSE157103_genes.csv")
df2 <- read.csv("/Users/wangdeyao/Downloads/QBS103/QBS103_GSE157103_series_matrix.csv")

df3<-df1 # Creates a copy of df1.
df4<-df2 # Creates a copy of df2.
# Prepare df2 by selecting necessary columns and converting some columns to lowercase
df2 <- df2 %>% select("age",
                      "ferritin.ng.ml.",
                      "crp.mg.l.",
                      "sex",
                      "icu_status",
```

```
                    "mechanical_ventilation")
# Converts 'icu_status' to lowercase and removes any leading/trailing whitespace.
df2$icu_status <- tolower(trimws(df2$icu_status))

df2$mechanical_ventilation <- tolower(trimws(df2$mechanical_ventilation))
df2 <- df2 %>%
  mutate(
    age = as.numeric(age),  # Converts 'age' to numeric.
    ferritin_ng_ml = as.numeric(`ferritin.ng.ml.`), # Converts 'ferritin.ng.ml.' to numeric and renames
    crp_mg_l = as.numeric(`crp.mg.l.`) # Converts 'crp.mg.l.' to numeric and renames.
  ) %>%
  filter(sex != "unknown") # Filter out rows where sex is 'unknown'
```

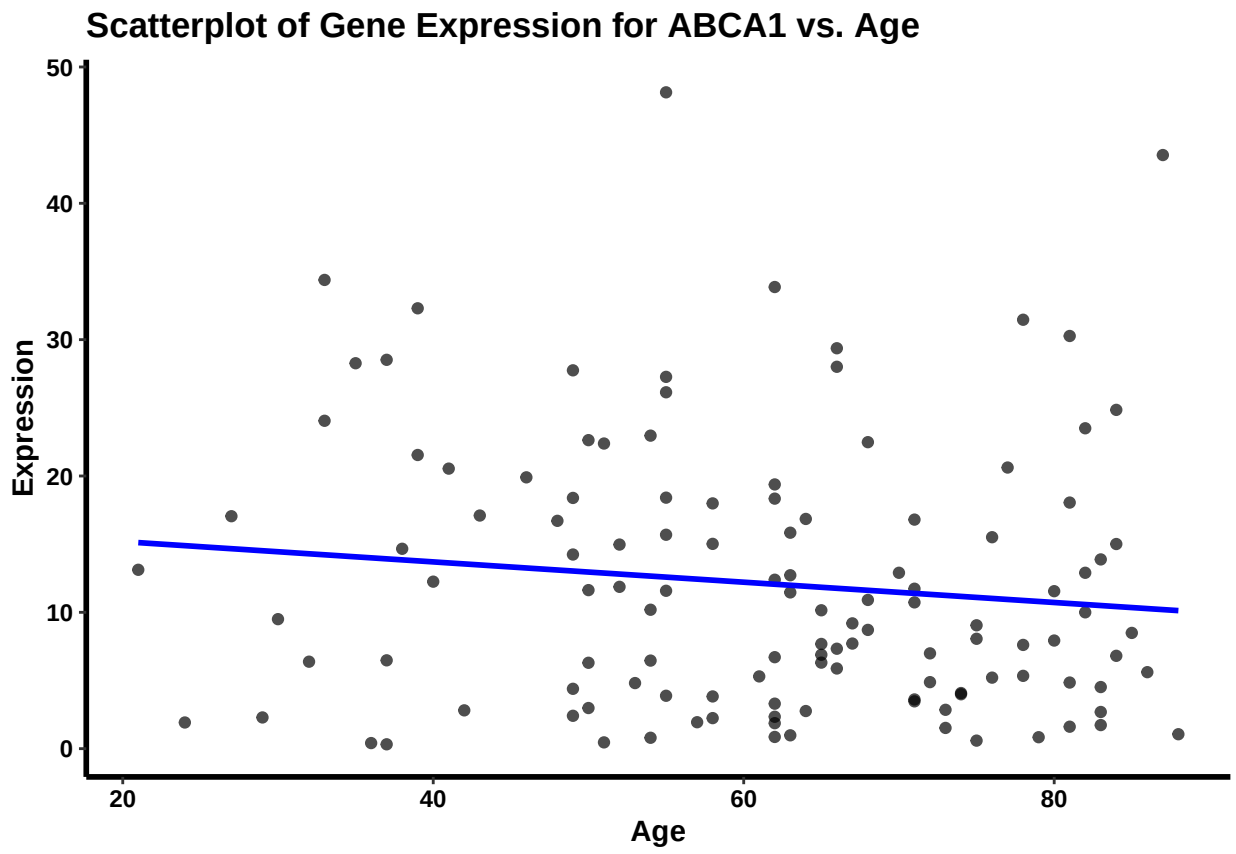
```
# Process data for the ABCA1 gene
data_long <- df3 %>%
  filter(X == "ABCA1")%>%  # Filters rows where the gene name is 'ABCA1'.
  gather(key = "Sample", value = "Expression",
         COVID_01_39y_male_NonICU:NONCOVID_26_36y_male_ICU)
# Merge datasets and select relevant columns
new_merged <- merge(data_long, df4, by.x = "Sample", by.y = "participant_id")%>%
  select("Sample", "Expression", "age", "sex","icu_status")
# Define a custom theme for plots
newTheme <- theme(
  panel.border = element_blank(), panel.grid.major = element_blank(),
  panel.grid.minor = element_blank(),
  axis.line = element_line(colour = "black", linewidth = rel(2)),
  plot.background = element_rect(fill = "white"),
  panel.background = element_blank(),
  legend.key = element_rect(fill = 'grey',color = "grey"),
  legend.background = element_rect(fill = 'grey'),
  legend.box.background = element_rect(color = "black"),
  text = element_text(face="bold",colour = "black"),
  axis.text = element_text(face="bold",colour = "black"),
        legend.position = 'bottom')

# Create and save a histogram of gene expression for ABCA1
ggplot(new_merged, aes(x = Expression)) +
  geom_histogram(binwidth = 1, fill = "black", color = "black", alpha = 0.5) +
  labs(title = "Histogram of Gene Expression foR ABCA1",
       x = "Expression",
       y = "Frequency") +
  newTheme
```




```
ggsave("histogram_abca1.png", plot = last_plot(), width = 10, height = 8, dpi = 300)
```

```
options(warn = -1)
library(ggpubr)
# Drop NA values and convert the 'age' column to numeric
new_merged <- new_merged%>%drop_na()
new_merged$age <- as.numeric(new_merged$age)
# Generate a scatterplot of Gene Expression vs Age and save the plot
ggplot(new_merged, aes(x = age, y = Expression)) +
  geom_point(color = "black", alpha = 0.7) +
  scale_y_continuous(breaks = seq(0, 100, by = 10)) + # Sets y-axis breaks.
  geom_smooth(method = "lm", color = "blue", se = FALSE) +  # Adds a linear model line without confiden
  labs(title = "Scatterplot of Gene Expression for ABCA1 vs. Age",
       y = "Expression", x = "Age") +
  newTheme
```

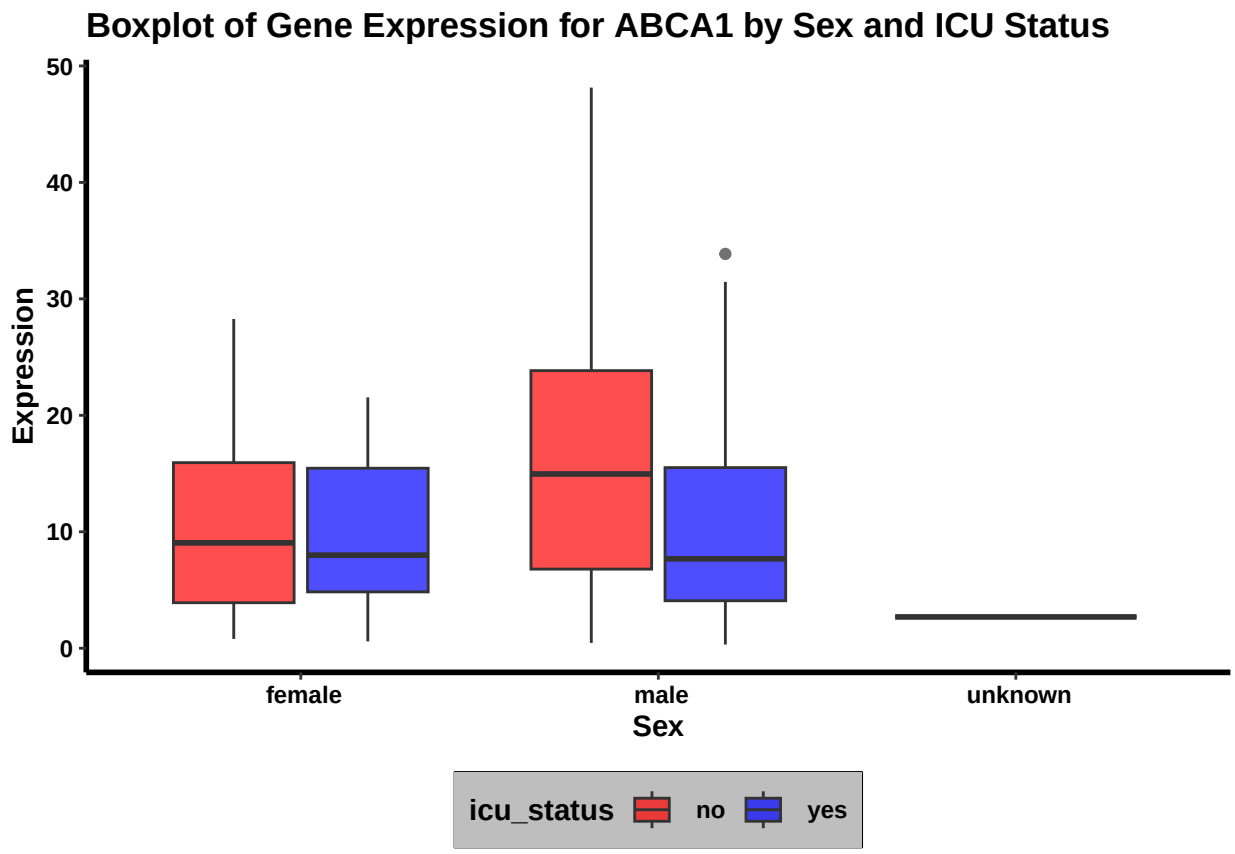
```
## `geom_smooth()` using formula = 'y ~ x'
```



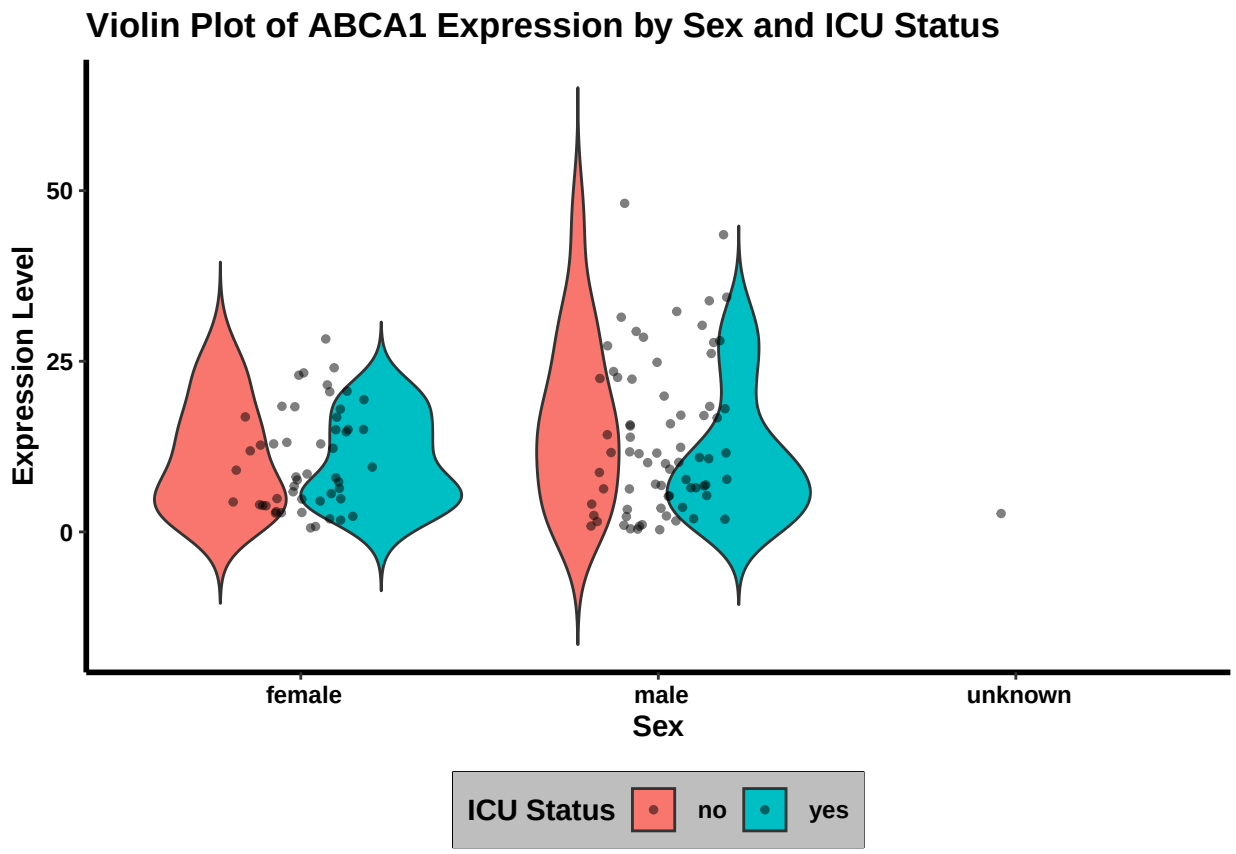

```
ggsave("scatterplot_abca1.png", plot = last_plot(), width = 10, height = 8, dpi = 300)
```

```
## `geom_smooth()` using formula = 'y ~ x'
```

```
colorpalette <- c("red", "blue", "yellow") # Defines a color palette.
# Create a boxplot of Gene Expression by Sex and ICU Status
ggplot(new_merged, aes(x = sex, y = Expression, fill = icu_status))+
geom_boxplot(alpha = 0.7)+  # Creates boxplots, setting transparency for aesthetic purposes.
labs(title = "Boxplot of Gene Expression for ABCA1 by Sex and ICU Status",
     x = "Sex", y = "Expression")+
scale_fill_manual(values = colorpalette)+ # Applies manual color filling based on ICU status.
newTheme
```




```
# Create a violin plot to show the distribution of expression data and save it
options(warn = -1)
ggplot(new_merged, aes(x = sex, y = Expression, fill = icu_status)) +
  geom_violin(trim = FALSE,drop = FALSE) +  # Generates violin plots, showing the full range of data.
  geom_jitter(width = 0.2, size = 1, alpha = 0.5) +  # Adds jittered points to show individual data poi
  labs(title = "Violin Plot of ABCA1 Expression by Sex and ICU Status",
       x = "Sex",
       y = "Expression Level",
       fill = "ICU Status") +
  newTheme
```




```
ggsave("Violin_ABCA1.png",plot = last_plot(), width = 10, height = 8, dpi = 300)
```

```
# Compute summary statistics and reshape the data for display in a LaTeX table
stats <- df2 %>%
  group_by(sex) %>%
  # Calculates mean and standard deviation for 'age' grouped by 'sex', formats them as mean (sd).
  summarize(
    age = paste0(round(mean(age, na.rm = TRUE), 2), " (",
                 round(sd(age, na.rm = TRUE), 2), ")"),
    ferritin = paste0(round(mean(
      ferritin_ng_ml, na.rm = TRUE # Does the same for 'ferritin_ng_ml'.
    ), 2), " (", round(sd(
      ferritin_ng_ml, na.rm = TRUE
    ), 2), ")"),
    crp = paste0(round(mean(crp_mg_l, na.rm = TRUE), 2), " (",
                 round(sd(crp_mg_l, na.rm = TRUE), 2), ")") # And for 'crp_mg_l'.
  ) %>%
  pivot_longer(
    cols = c(age, ferritin, crp),
    names_to = "variable",
    values_to = "value"
  ) %>%
  pivot_wider(names_from = sex, values_from = value) # Reshapes the data for better visualization, from
stats <- stats %>%
  select(-contains("unknown")) # Excludes columns that contain 'unknown'.
# Print the final 3x2 table
```

```r
print(stats)
```

```
## # A tibble: 3 x 3
##   variable ‘ female‘        ‘ male‘
##   <chr>    <chr>            <chr>
## 1 age      59.3 (17.92)     62.28 (14.41)
## 2 ferritin 619.28 (1054.33) 993.35 (1013.05)
## 3 crp      112.87 (99.77)   144.46 (102.55)
```

```r
# Compute ICU status and mechanical ventilation stats, then combine with the previous stats
cat_stats <- df2 %>%
  group_by(sex) %>%
  # Calculates and formats the proportion of 'yes' responses for ICU status.
  summarize(
    icu_status_yes = paste(sum(icu_status == "yes"), " (",
                           round(100 * sum(icu_status == "yes") / n(), 2), "%)", sep = ""),
    icu_status_no = paste(sum(icu_status == "no"), " (",
                          round(100 * sum(icu_status == "no") / n(), 2), "%)", sep = ""),

    mechanical_ventilation_yes = paste(
      sum(mechanical_ventilation == "yes"),
      " (",
      round(100 * sum(mechanical_ventilation == "yes") / n(), 2),
      "%)",
      sep = ""
    ), # Does the same for mechanical ventilation 'yes'.
    mechanical_ventilation_no = paste(
      sum(mechanical_ventilation == "no"),
      " (",
      round(100 * sum(mechanical_ventilation == "no") / n(), 2),
      "%)",
      sep = ""
    ),

  ) %>%
  pivot_longer(
    cols = c(icu_status_yes, icu_status_no, mechanical_ventilation_yes,
             mechanical_ventilation_no),
    names_to = "variable",
    values_to = "value"
  ) %>%
  pivot_wider(names_from = sex, values_from = value)
# Reshapes categorical data similarly to previous blocks for easier comparison.
final_stats <- bind_rows(stats, cat_stats) # Combines the two sets of stats into one dataset.
final_stats <- final_stats %>%
  select(-contains("unknown")) # Excludes 'unknown' categories once again for clarity.
```

```r
# Generate and print the final LaTeX table
latex_table <- kable(
  final_stats,
  format = "latex", # Specifies that the output format should be LaTeX.
  booktabs = TRUE,  # Uses booktabs style which is more professional looking.
  align = 'lcc',    # Aligns columns; 'l' for text and 'c' for numeric columns.
```

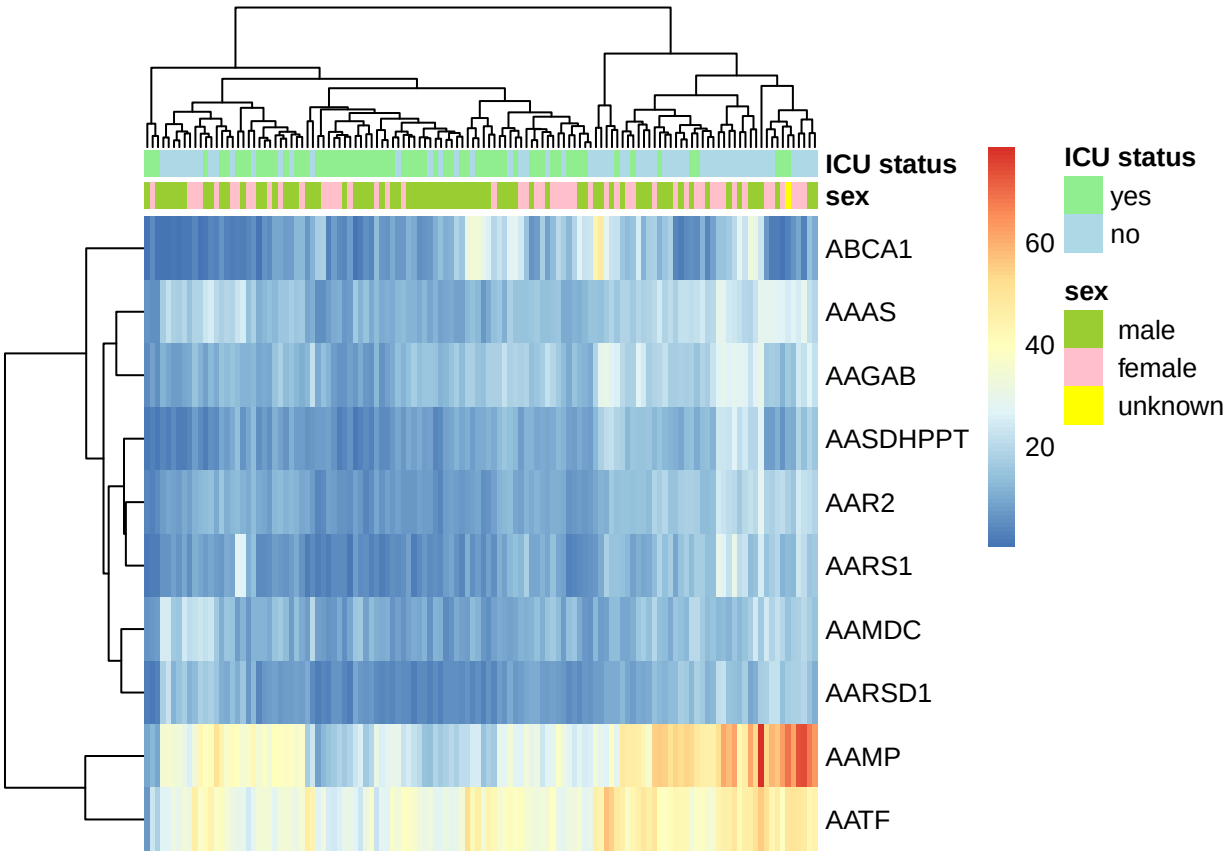
Table 1: Summary Statistics Stratified by Sex

variable	female	male
age	59.3 (17.92)	62.28 (14.41)
ferritin	619.28 (1054.33)	993.35 (1013.05)
crp	112.87 (99.77)	144.46 (102.55)
icu_status_yes	24 (47.06%)	41 (55.41%)
icu_status_no	27 (52.94%)	33 (44.59%)
mechanical_ventilation_yes	16 (31.37%)	35 (47.3%)
mechanical_ventilation_no	35 (68.63%)	39 (52.7%)

```
  escape = TRUE,      # Enables escaping of LaTeX special characters within the table.
  caption = "Summary Statistics Stratified by Sex"
) %>%
  kable_styling(
    full_width = FALSE,
    position = "center" # Centers the table in the LaTeX document.
  ) %>%
  row_spec(0, bold = TRUE)# Makes the first row bold for emphasis.
latex_table
```

```
# Generate a heatmap of the selected top genes with annotations
top_genes <- df1 %>% filter(
    X == "AAAS" |
    X == "AATF" |
    X == "AAGAB" |
    X == "ABCA1" |
    X == "AAMDC" |
    X == "AAMP" | X == "AAR2" | X == "AARS1" |
    X == "AARSD1" | X == "AASDHPPT"
)# Filters for rows in df1 where the gene name matches one of the specified genes.
rownames(top_genes) <- top_genes$X # Sets the gene names as row names for the heatmap.
top_genes <- top_genes[, -1]  # Removes the first column (gene names) after setting it as rownames.

annotations <- data.frame(
  sex = factor(df2$sex), # Creates a factor of 'sex' for coloring in the heatmap.
  `ICU status` = factor(df2$icu_status), # Creates a factor of 'ICU status' for coloring.
  check.names = FALSE # Prevents data.frame from trying to make variable names syntactically valid.
)
# Ensures the column names of the top_genes are used as rownames for the annotation frame.
rownames(annotations) <- colnames(top_genes)
annotation_colors <- list(
  sex = c(
    " male" = "yellowgreen",
    " female" = "pink",
    " unknown" = 'yellow'
  ),
  `ICU status` = c("yes" = "lightgreen", "no" = "lightblue")
) # Defines custom colors for the annotations based on 'sex' and 'ICU status'.

p <- pheatmap(
```

```
  top_genes,
  annotation_col = annotations,
  annotation_colors = annotation_colors,
  show_colnames = FALSE, # Hides column names in the heatmap.
  clustering_distance_rows = "euclidean", # Uses Euclidean distance for row clustering.
  clustering_distance_cols = "euclidean" # Uses Euclidean distance for column clustering.
) # Generates the heatmap with the specified annotations and distance measures.
```




```
ggsave("heatmap_Final_Submission.png", plot = p, width = 10, height = 8, dpi = 300)
```