Final Project

Eric Chen

2024-07-30

#1

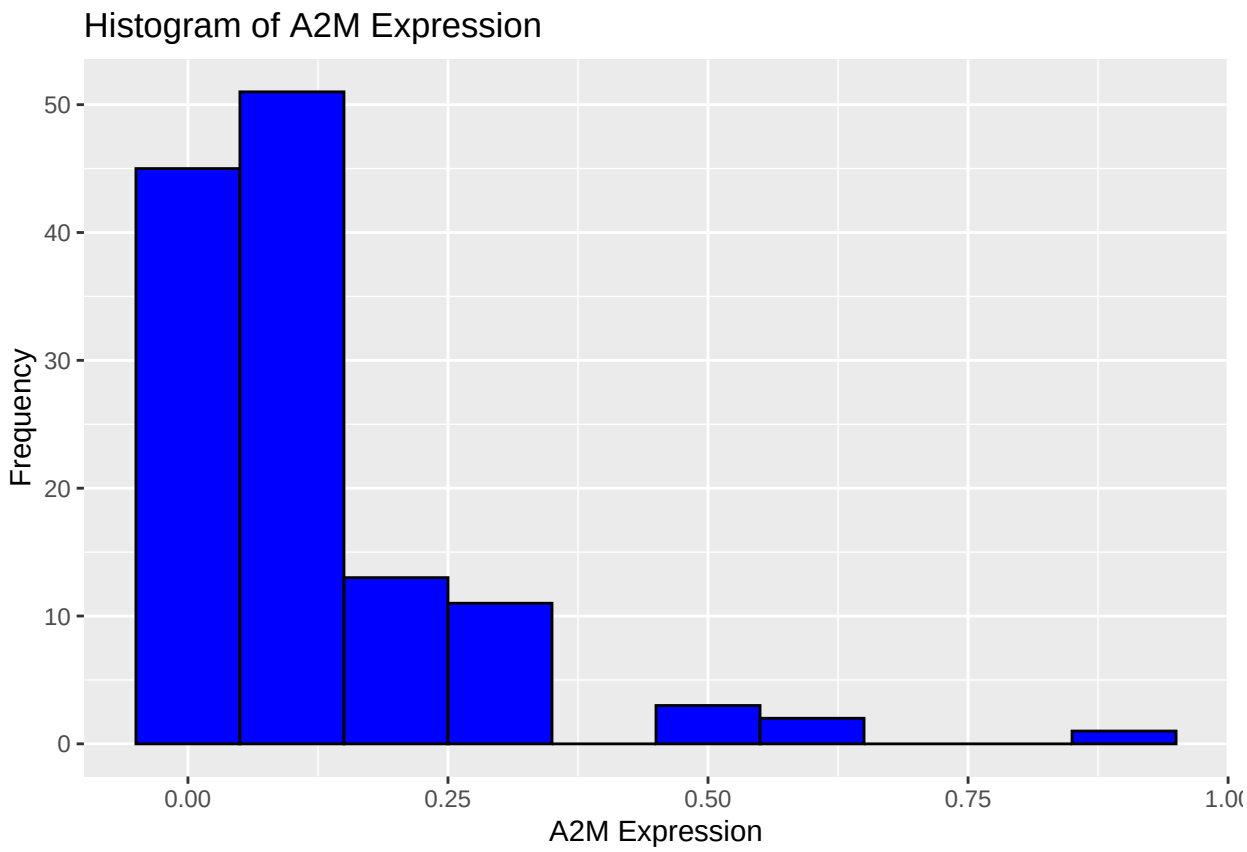
```
# Load necessary libraries
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.4
## v forcats   1.0.0     v stringr   1.5.0
## v ggplot2   3.4.4     v tibble    3.2.1
## v lubridate 1.9.3     v tidyr     1.3.0
## v purrr     1.0.1
## -- 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(tidyr)

# Read the data
metadata <- read.csv("QBS103_GSE157103_series_matrix.csv")
gene_expression <- read.csv("QBS103_GSE157103_genes.csv")

# Convert gene expression data to long format
gene_expression_long <- gene_expression %>%
  pivot_longer(cols = -X, names_to = "Sample", values_to = "Expression") %>%
  rename(Gene = X)

# Merge the data
merged_data <- left_join(gene_expression_long, metadata, by = c("Sample" = "participant_id"))

# Select a gene for analysis
selected_gene <- "A2M"
plot_data <- merged_data %>% filter(Gene == selected_gene)

# Convert continuous covariate (age) to numeric, handle non-numeric values
plot_data$age <- as.numeric(plot_data$age)
```

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

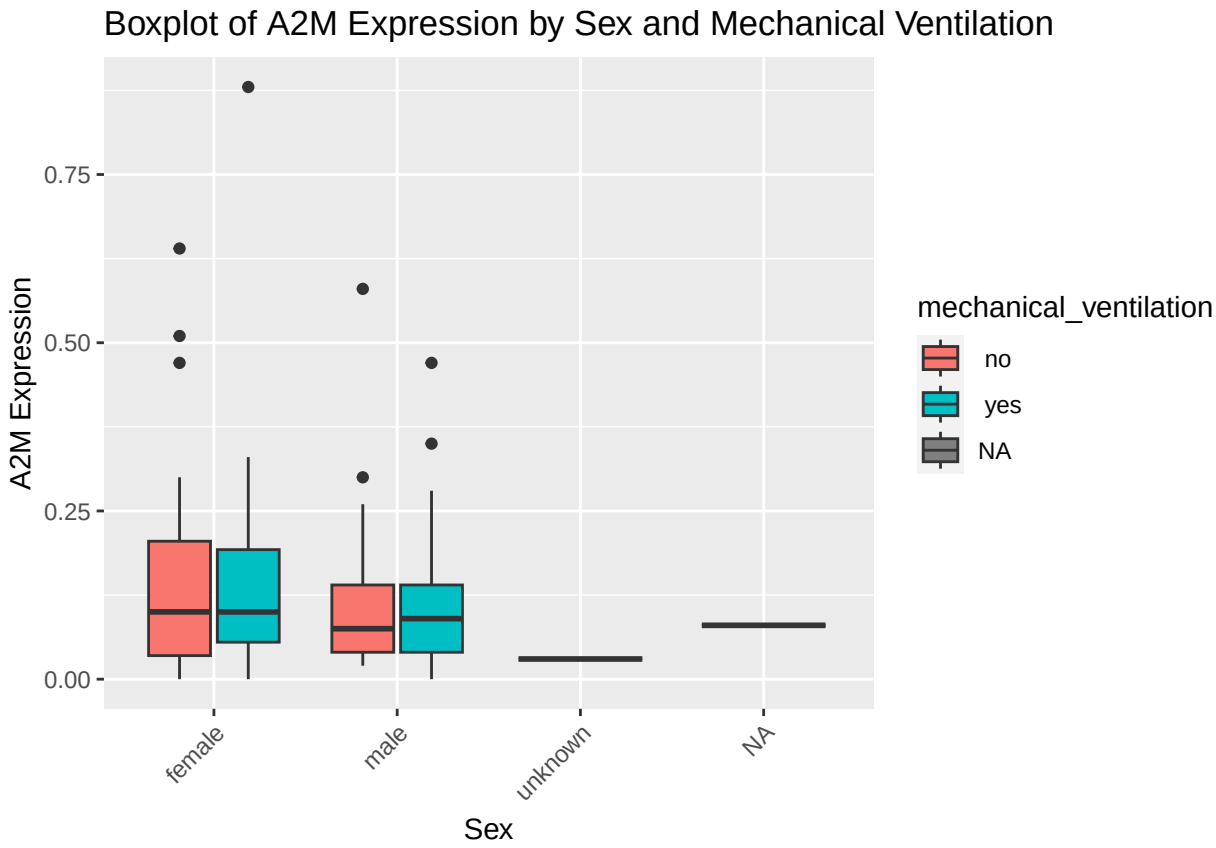
```r
# Generate a histogram for gene expression
ggplot(plot_data, aes(x = Expression)) +
  geom_histogram(binwidth = 0.1, fill = "blue", color = "black") +  # Adjust binwidth
  labs(title = paste("Histogram of", selected_gene, "Expression"), x = paste(selected_gene, "Expression
```




```r
# Generate a scatter plot for gene expression and continuous covariate (age)
# Remove NA values from age
plot_data_scatter <- plot_data %>% filter(!is.na(age))

ggplot(plot_data_scatter, aes(x = age, y = Expression)) +
  geom_point() +
  geom_smooth(method = "loess", color = "red") +  # Adding a smoothed line
  labs(title = paste("Scatterplot of", selected_gene, "Expression and Age"), x = "Age", y = paste(selec
```

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

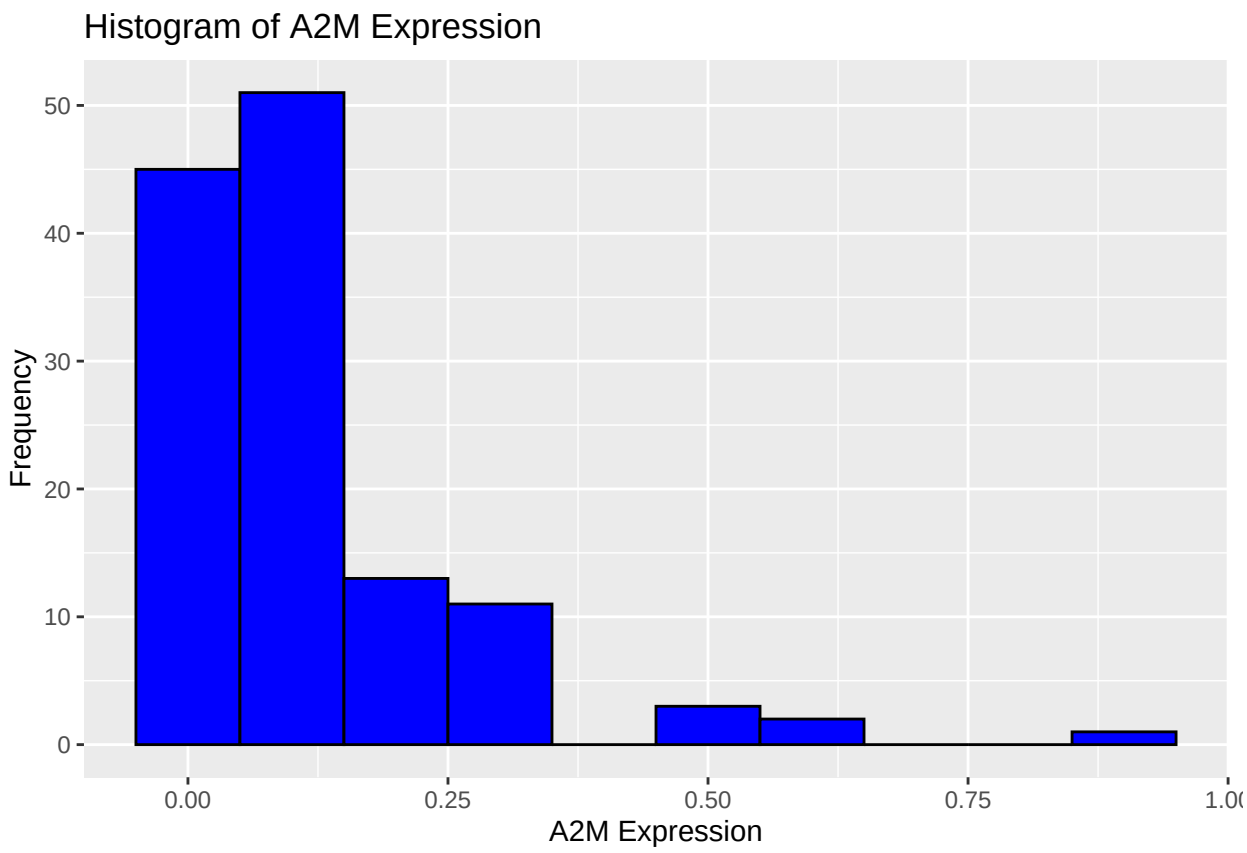

```r
# Generate a boxplot of gene expression separated by both categorical covariates (sex and mechanical_ve
# Handle 'unknown' value in sex variable
plot_data <- plot_data %>% mutate(sex = ifelse(sex == "unknown", "Unknown", sex))

ggplot(plot_data, aes(x = sex, y = Expression, fill = mechanical_ventilation)) +
  geom_boxplot() +
  labs(title = paste("Boxplot of", selected_gene, "Expression by Sex and Mechanical Ventilation"), x =
  theme(axis.text.x = element_text(angle = 45, hjust = 1))  # Rotate x-axis labels for better readabili
```



#2

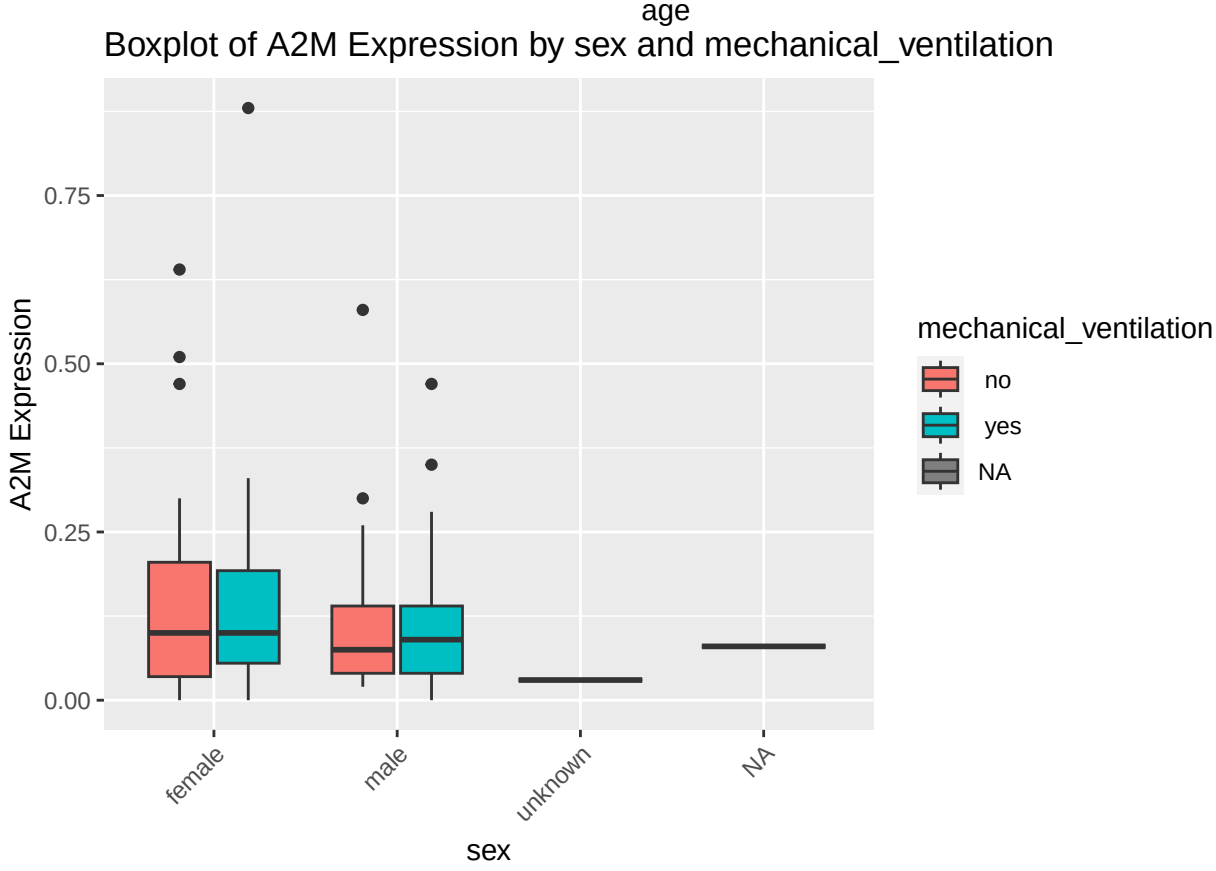
```
# Function to create plots
create_plots <- function(data, genes, continuous_covariate, categorical_covariate1, categorical_covariat
  for (gene in genes) {
    plot_data <- data %>% filter(Gene == gene)

    # Convert continuous covariate to numeric, handle non-numeric values
    plot_data[[continuous_covariate]] <- as.numeric(plot_data[[continuous_covariate]])

    # Histogram for gene expression
    print(
      ggplot(plot_data, aes(x = Expression)) +
        geom_histogram(binwidth = 0.1, fill = "blue", color = "black") +
        labs(title = paste("Histogram of", gene, "Expression"), x = paste(gene, "Expression"), y = "Freq
    )

    # Scatter plot for gene expression and continuous covariate
    plot_data_scatter <- plot_data %>% filter(!is.na(plot_data[[continuous_covariate]]))

    print(
      ggplot(plot_data_scatter, aes_string(x = continuous_covariate, y = "Expression")) +
        geom_point() +
        geom_smooth(method = "loess", color = "red") +
        labs(title = paste("Scatterplot of", gene, "Expression and", continuous_covariate), x = continu
    )

    # Boxplot of gene expression separated by both categorical covariates
    plot_data <- plot_data %>% mutate(!!categorical_covariate1 := ifelse(get(categorical_covariate1) ==
```

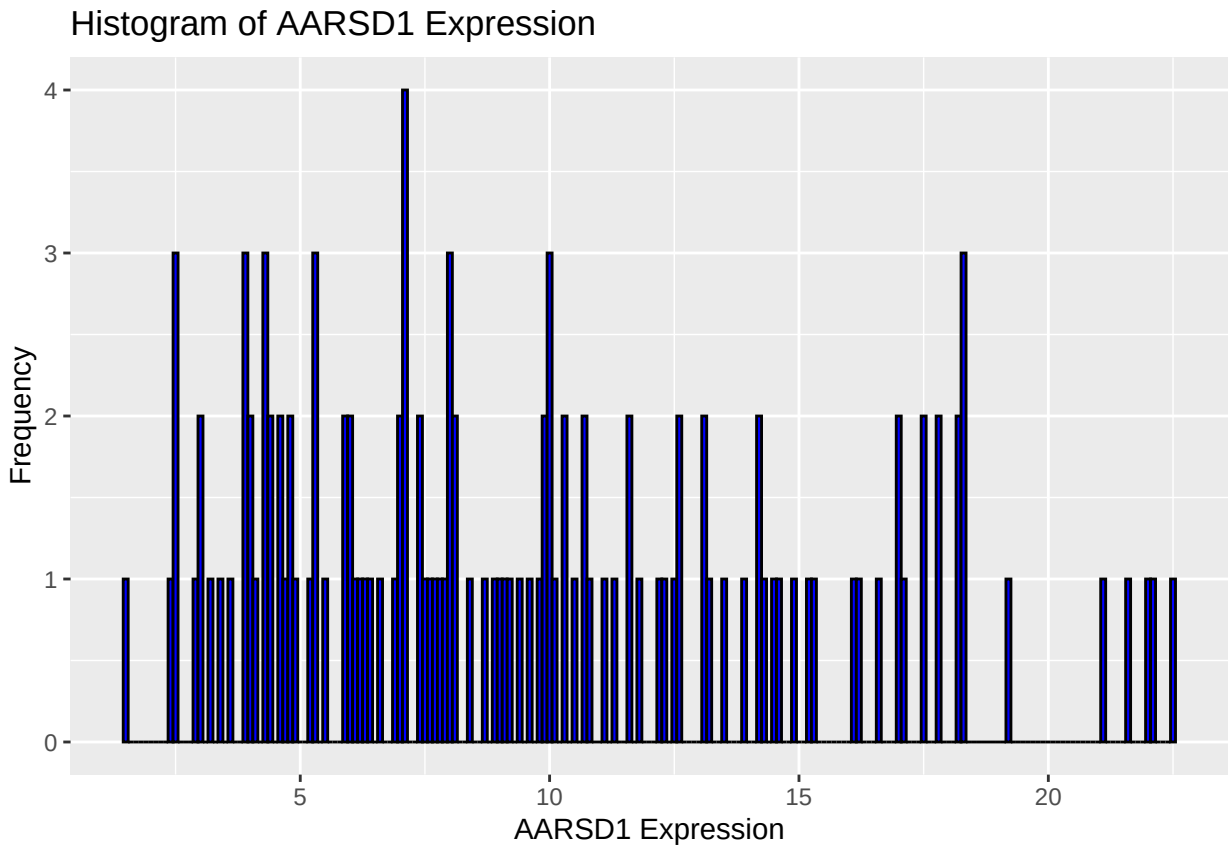
```
    print(
      ggplot(plot_data, aes_string(x = categorical_covariate1, y = "Expression", fill = categorical_cova
        geom_boxplot() +
        labs(title = paste("Boxplot of", gene, "Expression by", categorical_covariate1, "and", categori
        theme(axis.text.x = element_text(angle = 45, hjust = 1))
    )
  }
}

# Select additional genes
additional_genes <- c("A2M", "AARSD1", "ABHD2")

# Generate figures using the function
create_plots(data = merged_data, genes = additional_genes, continuous_covariate = "age", categorical_co
```




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

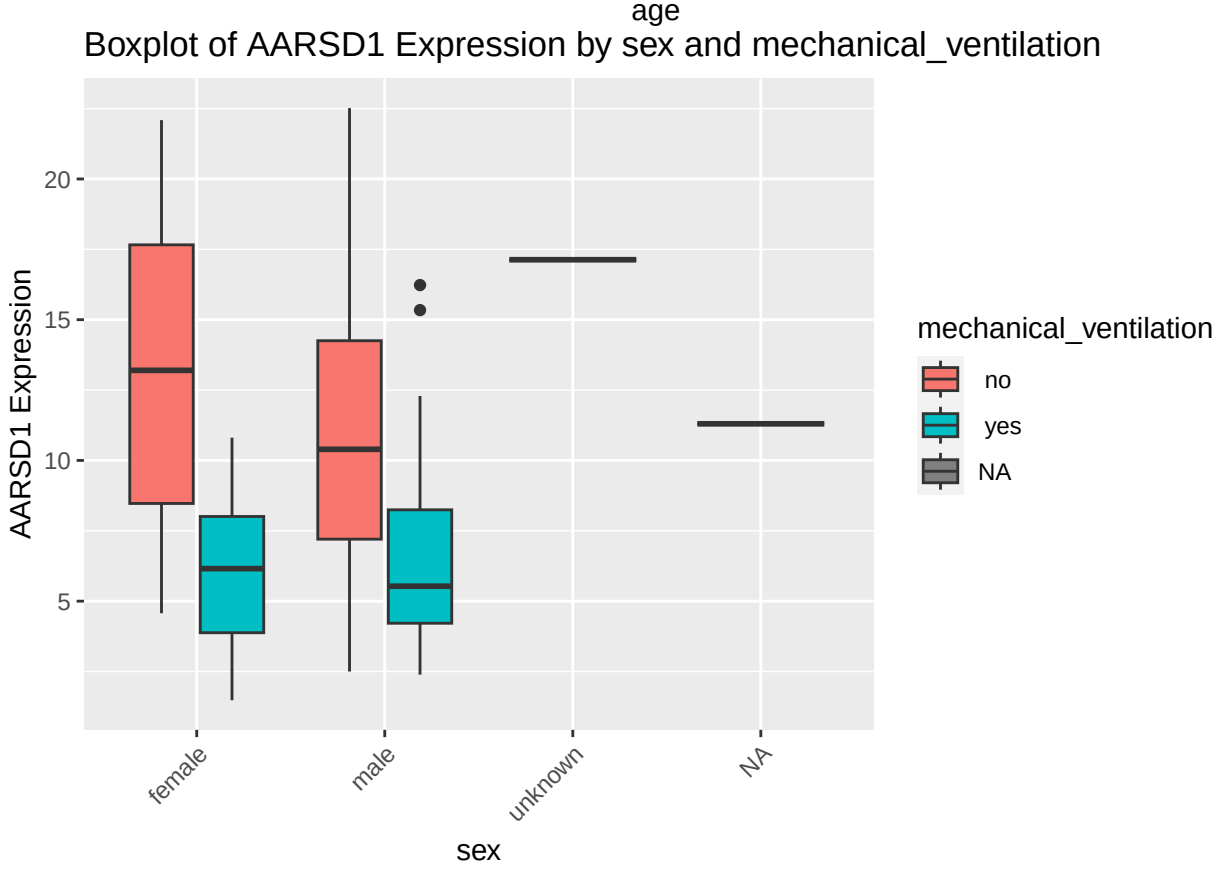

Scatterplot of A2M Expression and age
A2M Expression
0.75
0.50
0.25
0.00
20
40
60
80
age

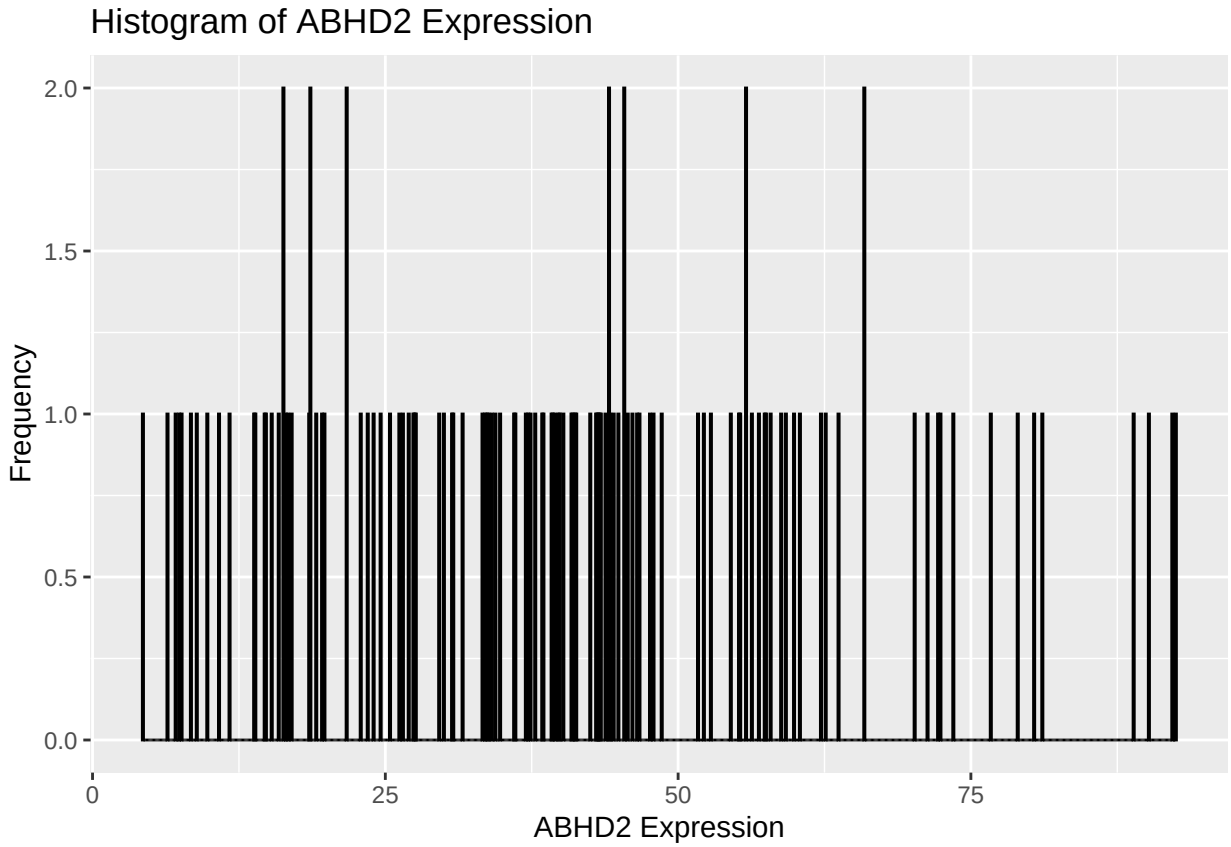

Boxplot of A2M Expression by sex and mechanical_ventilation
A2M Expression
0.75
0.50
0.25
0.00
female
male
unknown
NA
sex
mechanical_ventilation
no
yes
NA




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

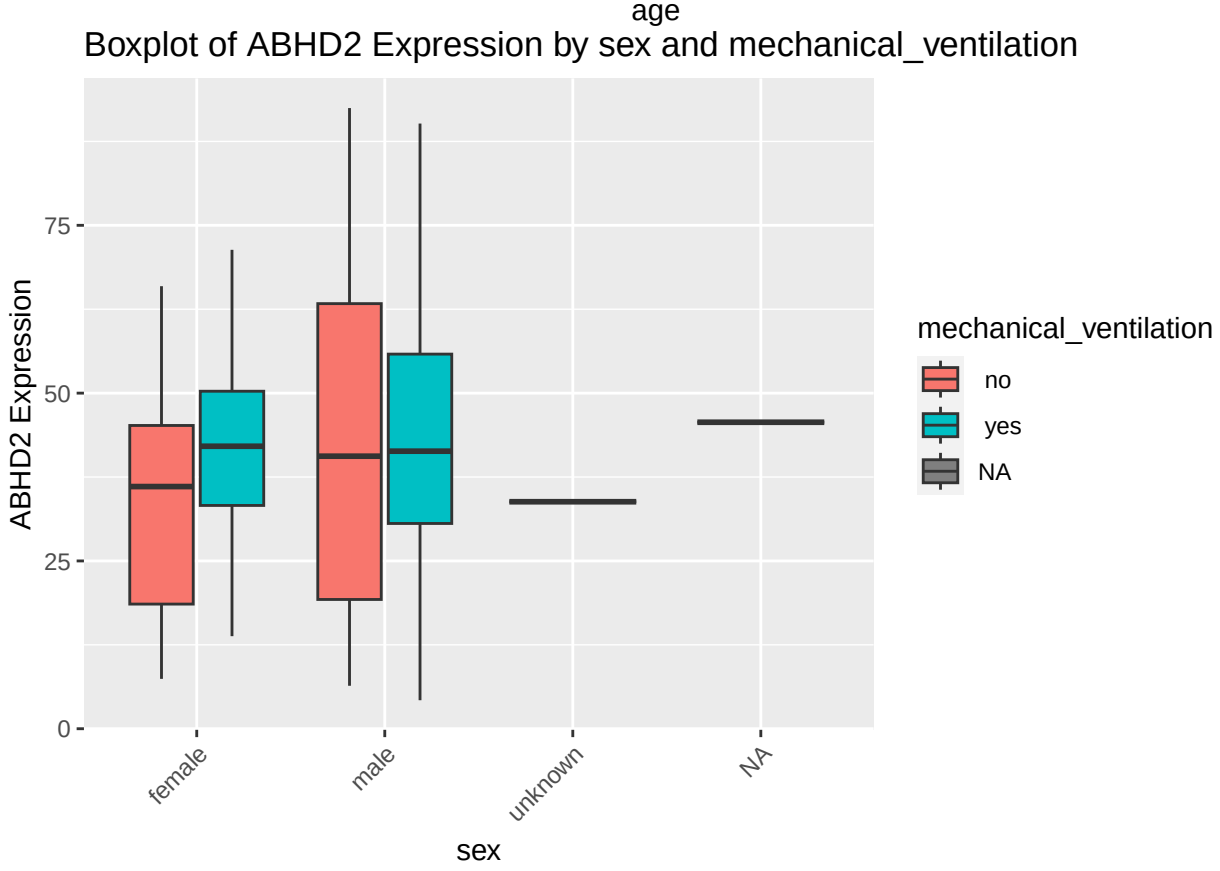

Scatterplot of AARSD1 Expression and age
AARSD1 Expression
20
15
10
5
20
40
60
80
age


Boxplot of AARSD1 Expression by sex and mechanical_ventilation
AARSD1 Expression
20
15
10
5
female
male
unknown
NA
sex
mechanical_ventilation
no
yes
NA




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


Scatterplot of ABHD2 Expression and age
ABHD2 Expression
0
25
50
75
20
40
60
80
age


Boxplot of ABHD2 Expression by sex and mechanical_ventilation
ABHD2 Expression
0
25
50
75
female
male
unknown
NA
sex
mechanical_ventilation
no
yes
NA

#Final

```r
# Load necessary libraries
library(tidyverse)
library(tidyr)
library(pheatmap)

# Read the data
metadata <- read.csv("QBS103_GSE157103_series_matrix.csv")
gene_expression <- read.csv("QBS103_GSE157103_genes.csv")

# Convert gene expression data to long format
gene_expression_long <- gene_expression %>%
  pivot_longer(cols = -X, names_to = "Sample", values_to = "Expression") %>%
  rename(Gene = X)

# Merge the data
merged_data <- left_join(gene_expression_long, metadata, by = c("Sample" = "participant_id"))

# Select a gene for analysis
selected_gene <- "A2M"
plot_data <- merged_data %>% filter(Gene == selected_gene)

# Convert continuous covariate (age) to numeric, handle non-numeric values
plot_data$age <- as.numeric(plot_data$age)

# Generate a histogram for gene expression
hist_plot <- ggplot(plot_data, aes(x = Expression)) +
  geom_histogram(binwidth = 0.1, fill = "blue", color = "black") +
  labs(title = paste("Histogram of", selected_gene, "Expression"), x = paste(selected_gene, "Expression

# Save the histogram
ggsave("histogram_a2m_expression.pdf", plot = hist_plot)

# Generate a scatter plot for gene expression and continuous covariate (age)
# Remove NA values from age
plot_data_scatter <- plot_data %>% filter(!is.na(age))

scatter_plot <- ggplot(plot_data_scatter, aes(x = age, y = Expression)) +
  geom_point() +
  geom_smooth(method = "loess", color = "red") +  # Adding a smoothed line
  labs(title = paste("Scatterplot of", selected_gene, "Expression and Age"), x = "Age", y = paste(selec

# Save the scatter plot
ggsave("scatterplot_a2m_age.pdf", plot = scatter_plot)

# Generate a boxplot of gene expression separated by both categorical covariates (sex and mechanical_ve
plot_data <- plot_data %>% mutate(sex = ifelse(sex == "unknown", "Unknown", sex))

box_plot <- ggplot(plot_data, aes(x = sex, y = Expression, fill = mechanical_ventilation)) +
  geom_boxplot() +
  labs(title = paste("Boxplot of", selected_gene, "Expression by Sex and Mechanical Ventilation"), x =
  theme(axis.text.x = element_text(angle = 45, hjust = 1))

# Save the boxplot
```

```r
ggsave("boxplot_a2m_sex_ventilation.pdf", plot = box_plot)
```

#Generate a summary table

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

```
## Warning in !is.null(rmarkdown::metadata$output) && rmarkdown::metadata$output
## %in% : 'length(x) = 2 > 1' in coercion to 'logical(1)'
```

```
##
## Attaching package: 'kableExtra'
```

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

```r
# Clean and Convert the Data
metadata_clean <- metadata %>%
  mutate(
    age = as.numeric(gsub("[^0-9]", "", age)),
    mechanical_ventilation = trimws(mechanical_ventilation),
    sex = trimws(sex),
    ferritin = as.numeric(replace(ferritin.ng.ml., ferritin.ng.ml. == "unknown", NA)),
    crp = as.numeric(replace(crp.mg.l., crp.mg.l. == "unknown", NA)),
    disease_status = factor(disease_status)
  ) %>%
  filter(sex != "unknown")
```

```
## Warning: There were 2 warnings in `mutate()`.
## The first warning was:
## i In argument: `ferritin = as.numeric(...)`.
## Caused by warning:
## ! NAs introduced by coercion
## i Run `dplyr::last_dplyr_warnings()` to see the 1 remaining warning.
```

```r
# Split data by disease status
covid_data <- metadata_clean %>% filter(disease_status == "disease state: COVID-19")
non_covid_data <- metadata_clean %>% filter(disease_status == "disease state: non-COVID-19")

# Continuous Variables: mean (sd)
covid_age_mean_sd <- paste0(round(mean(covid_data$age, na.rm = TRUE), 1), " (", round(sd(covid_data$age
covid_ferritin_mean_sd <- paste0(round(mean(covid_data$ferritin, na.rm = TRUE), 1), " (", round(sd(covi
covid_crp_mean_sd <- paste0(round(mean(covid_data$crp, na.rm = TRUE), 1), " (", round(sd(covid_data$crp

non_covid_age_mean_sd <- paste0(round(mean(non_covid_data$age, na.rm = TRUE), 1), " (", round(sd(non_co
non_covid_ferritin_mean_sd <- paste0(round(mean(non_covid_data$ferritin, na.rm = TRUE), 1), " (", round
non_covid_crp_mean_sd <- paste0(round(mean(non_covid_data$crp, na.rm = TRUE), 1), " (", round(sd(non_co

# Categorical Variables: n (%)
covid_sex_counts <- covid_data %>%
```

```
  count(sex) %>%
  mutate(perc = n / sum(n) * 100,
         result = paste0(n, " (", round(perc, 1), "%)"))

covid_mechanical_vent_counts <- covid_data %>%
  count(mechanical_ventilation) %>%
  mutate(perc = n / sum(n) * 100,
         result = paste0(n, " (", round(perc, 1), "%)"))

non_covid_sex_counts <- non_covid_data %>%
  count(sex) %>%
  mutate(perc = n / sum(n) * 100,
         result = paste0(n, " (", round(perc, 1), "%)"))

non_covid_mechanical_vent_counts <- non_covid_data %>%
  count(mechanical_ventilation) %>%
  mutate(perc = n / sum(n) * 100,
         result = paste0(n, " (", round(perc, 1), "%)"))

# Combine all results into a final table
final_table <- data.frame(
  Variable = c("Age mean (sd)", "Ferritin (ng/ml)", "CRP (mg/L)",
               "Female n (%)", "Male n (%)",
               "Mechanical Ventilation (Yes) n (%)", "Mechanical Ventilation (No) n (%)"),

  COVID_19 = c(covid_age_mean_sd, covid_ferritin_mean_sd, covid_crp_mean_sd,
               covid_sex_counts$result[covid_sex_counts$sex == "female"],
               covid_sex_counts$result[covid_sex_counts$sex == "male"],
               covid_mechanical_vent_counts$result[covid_mechanical_vent_counts$mechanical_ventilation =
               covid_mechanical_vent_counts$result[covid_mechanical_vent_counts$mechanical_ventilation =

  Non_COVID_19 = c(non_covid_age_mean_sd, non_covid_ferritin_mean_sd, non_covid_crp_mean_sd,
                   non_covid_sex_counts$result[non_covid_sex_counts$sex == "female"],
                   non_covid_sex_counts$result[non_covid_sex_counts$sex == "male"],
                   non_covid_mechanical_vent_counts$result[non_covid_mechanical_vent_counts$mechanical_v
                   non_covid_mechanical_vent_counts$result[non_covid_mechanical_vent_counts$mechanical_v
)


print(final_table)
```

```
##                            Variable      COVID_19  Non_COVID_19
## 1                       Age mean (sd)   61.1 (16.3)       63 (16)
## 2                    Ferritin (ng/ml)  932.8 (1094) 250.5 (238.2)
## 3                          CRP (mg/L) 140.5 (103.6)   73.8 (71.1)
## 4                        Female n (%)      38 (38%)      13 (52%)
## 5                          Male n (%)      62 (62%)      12 (48%)
## 6 Mechanical Ventilation (Yes) n (%)      42 (42%)       9 (36%)
## 7  Mechanical Ventilation (No) n (%)      58 (58%)      16 (64%)
```

```
# Define the table columns
col_names <- c("Variable", "COVID-19 (n = 100)", "Non-COVID-19 (n = 25)")
```

Table 1: Summary Table by Disease Status

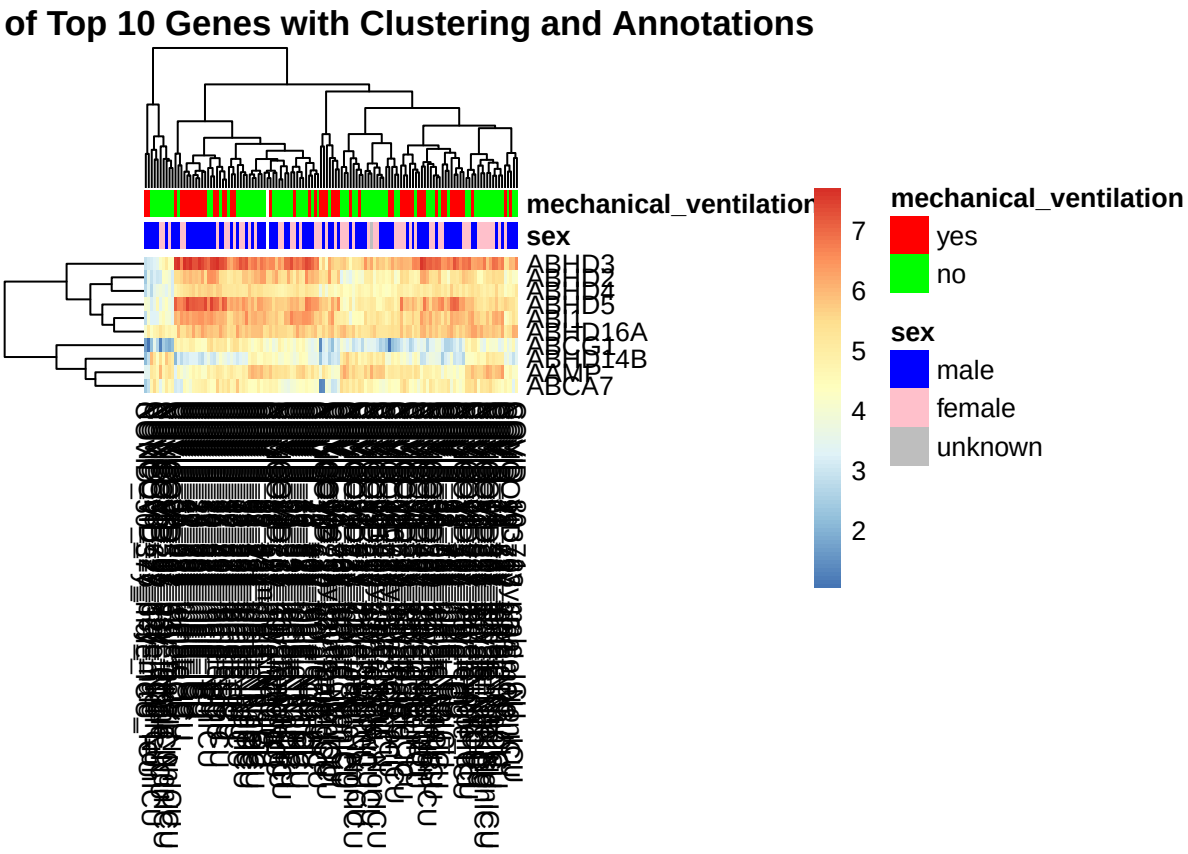
Variable	COVID-19 (n = 100)	Non-COVID-19 (n = 25)
Age mean (sd)	61.1 (16.3)	63 (16)
Ferritin (ng/ml)	932.8 (1094)	250.5 (238.2)
CRP (mg/L)	140.5 (103.6)	73.8 (71.1)
Female n (%)	38 (38%)	13 (52%)
Male n (%)	62 (62%)	12 (48%)
Mechanical Ventilation (Yes) n (%)	42 (42%)	9 (36%)
Mechanical Ventilation (No) n (%)	58 (58%)	16 (64%)

```
# Print the table
tab <- kable(x = final_table, caption = 'Summary Table by Disease Status',
             format = 'latex', booktabs = T,
             col.names = col_names,
             align = c('l','r','r'), escape = T) %>%
  add_indent(positions = c(4, 5, 6, 7), level_of_indent = 1)  # Adjusted positions

tab
```

#Generate the heatmap and selected new plot type

```
# Read the data
metadata <- read.csv("QBS103_GSE157103_series_matrix.csv")
gene_expression <- read.csv("QBS103_GSE157103_genes.csv")

# Convert gene expression data to long format
gene_expression_long <- gene_expression %>%
  pivot_longer(cols = -X, names_to = "Sample", values_to = "Expression") %>%
  rename(Gene = X)

# Merge the gene expression data with metadata
merged_data <- left_join(gene_expression_long, metadata, by = c("Sample" = "participant_id"))

# Check for duplicates and aggregate by taking the mean of duplicates
heatmap_data <- merged_data %>%
  group_by(Gene, Sample) %>%
  summarize(Expression = mean(Expression, na.rm = TRUE)) %>%
  ungroup()

# Convert to wide format
heatmap_data_wide <- heatmap_data %>%
  pivot_wider(names_from = Sample, values_from = Expression) %>%
  column_to_rownames(var = "Gene")

# Calculate variance for each gene and select the top 10 most variable genes
variance <- apply(heatmap_data_wide, MARGIN = 1, FUN = var)
top_genes <- names(sort(variance, decreasing = TRUE))[1:10]
heatmap_data_top <- heatmap_data_wide[top_genes,]

# Log2 transform the data to normalize the expression values
```

```r
log2_heatmap_data <- log2(heatmap_data_top + 1)

# Prepare annotation data for tracking bars
annotation_data <- metadata %>%
  select(participant_id, sex, mechanical_ventilation) %>%
  filter(participant_id %in% colnames(log2_heatmap_data)) %>%
  column_to_rownames(var = "participant_id")

# Define the annotation colors
annotation_colors <- list(
  sex = c("male" = "blue", "female" = "pink", "unknown" = "grey"),
  mechanical_ventilation = c("yes" = "red", "no" = "green")
)

#Remove whitespace
annotation_data$sex <- trimws(annotation_data$sex)
annotation_data$mechanical_ventilation <- trimws(annotation_data$mechanical_ventilation)

# Save the heatmap
pdf("heatmap_top10_genes.pdf",width = 20, height = 20)
pheatmap(log2_heatmap_data,
         clustering_distance_cols = 'euclidean',
         clustering_distance_rows = 'euclidean',
         annotation_col = annotation_data,
         annotation_colors = annotation_colors,
         main = "Heatmap of Top 10 Genes with Clustering and Annotations")

# Generate a density plot to compare the distribution of A2M expression by sex and mechanical ventilati
density_plot <- ggplot(plot_data, aes(x = Expression, fill = sex)) +
  geom_density(alpha = 0.5) +
  facet_wrap(~ mechanical_ventilation) +
  labs(title = paste("Density Plot of", selected_gene, "Expression by Sex and Mechanical Ventilation"),
       x = paste(selected_gene, "Expression"),
       y = "Density") +
  theme_minimal()
```




```
# Save the density plot
ggsave("density_plot_a2m_expression.pdf", plot = density_plot)
```