plot_function

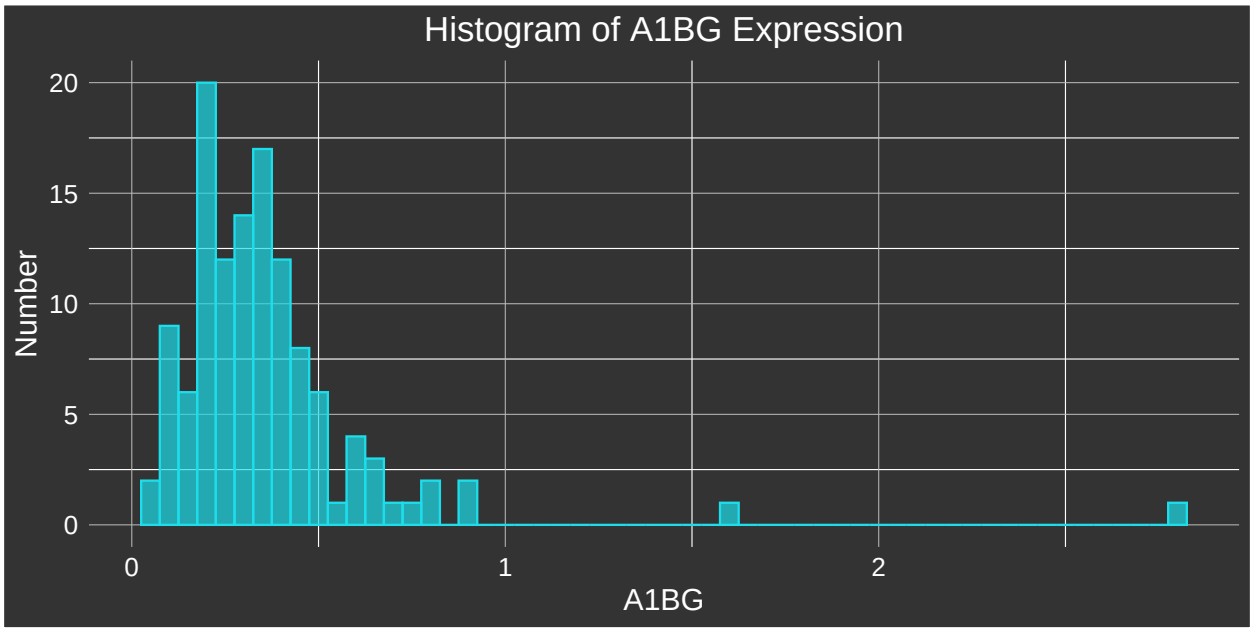
2024-08-08

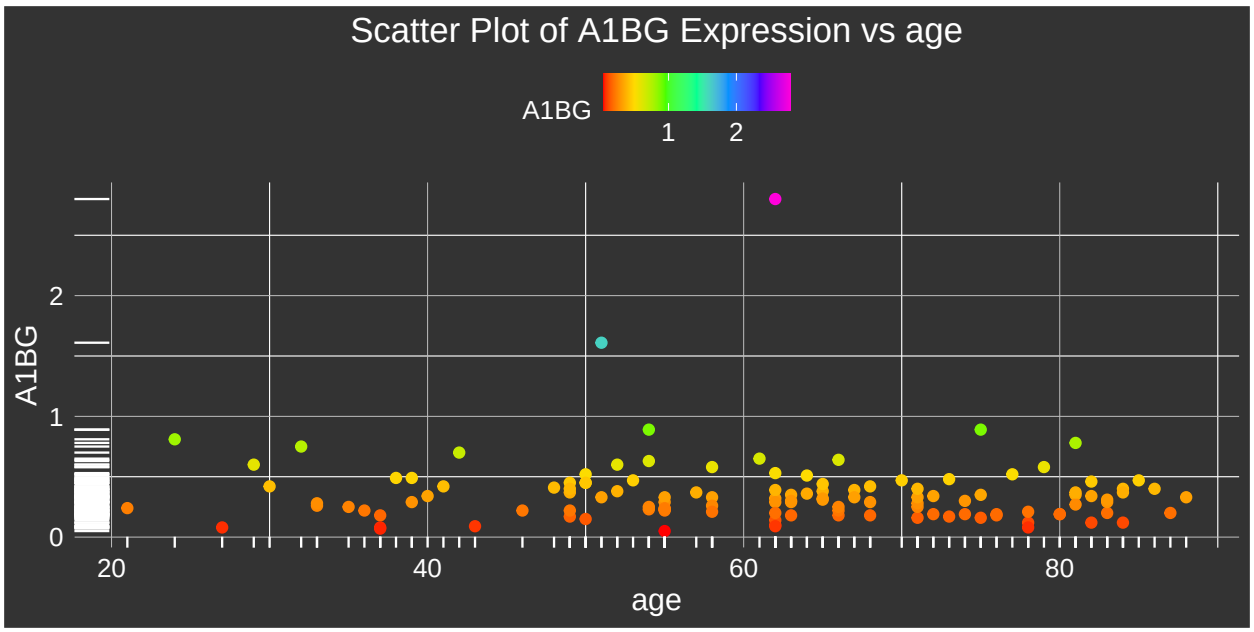
```
# Define a custom theme
custom_theme <- theme(
    panel.background = element_rect(fill = '#333333'),
    plot.title = element_text(hjust = 0.5, color = 'white', size = 16),
    axis.title = element_text(size = 14, color = 'white'),
    axis.text = element_text(size = 12, color = 'white'),
    plot.background = element_rect(fill = '#333333'),
    panel.grid.major = element_line(color = "gray", size = 0.1),
    legend.position = 'top',
    legend.background = element_rect(fill = "#333333"),
    legend.text = element_text(size = 12, color = 'white'),
    legend.title = element_text(size = 12, color = 'white'),
    legend.key = element_rect(fill = "#333333")
)
```

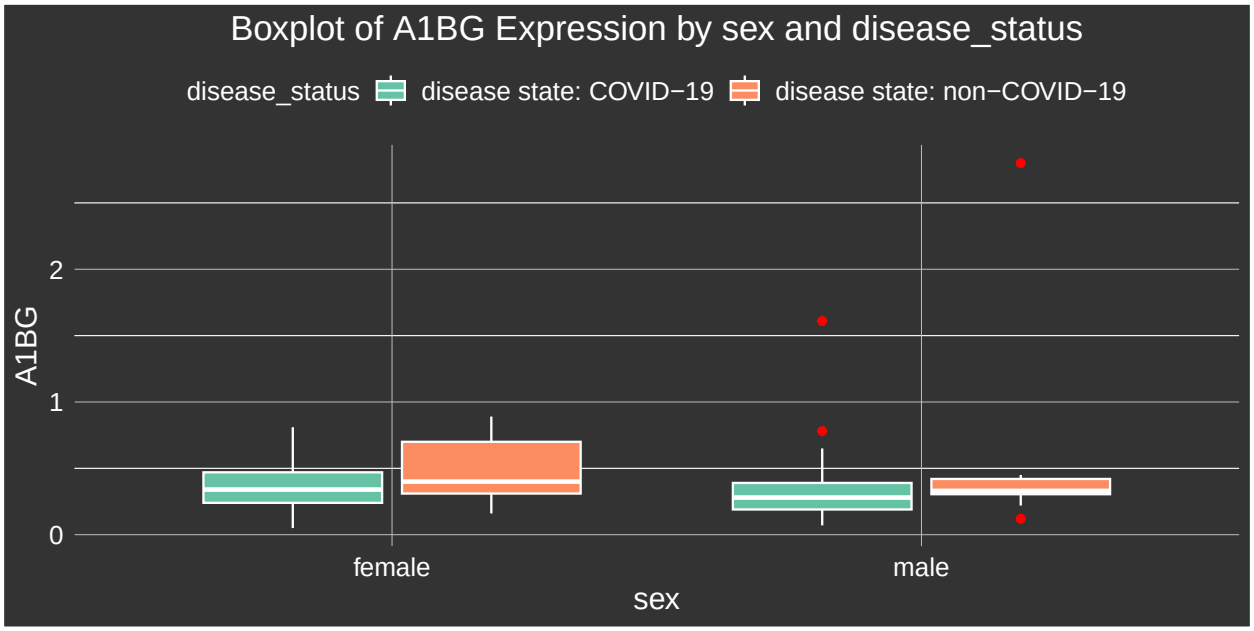
```
# Define the function to create plots
create_plots <- function(meta_data_path, gene_data_path, gene_names,
                         continuous_covariate, categorical_covariate1,
                         categorical_covariate2) {
    # Load the datasets
    genes_data <- read_csv(gene_data_path, show_col_types = FALSE)
    metadata <- read_csv(meta_data_path, show_col_types = FALSE)

    # Check if the data is read properly
    if (is.null(genes_data) || is.null(metadata)) {
      stop("Failed to read the datasets")
    }

    # Rename the column `...1` to `Genes` if it exists
    if ("...1" %in% colnames(genes_data)) {
        genes_data <- genes_data %>% rename(Genes = `...1`)
    } else {
        stop("Column '...1' not found in gene data")
    }

    # Set gene names as row names
    rownames(genes_data) <- genes_data$Genes

    for (gene_name in gene_names) {
        gene_name <- as.character(gene_name)
        #print(paste("Processing gene:", gene_name))
        if (!(gene_name %in% rownames(genes_data))) {
            warning(paste("Gene name", gene_name, "not found in the genes data"))
            next
```

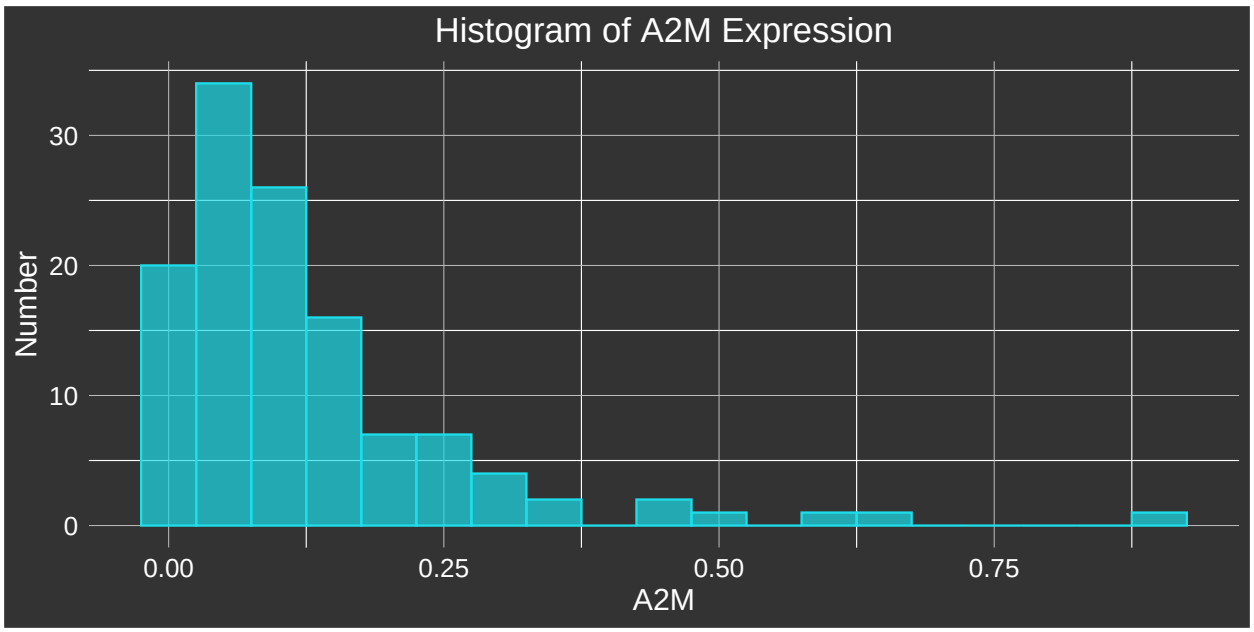
```
  }

# Select the gene row and transpose
gene <- genes_data[gene_name, ]
transposed_gene <- as.data.frame(t(gene))
colnames(transposed_gene) <- gene_name
# Combine two datasets with participant_id and gene
transposed_gene_data <- cbind(participant_id = rownames(transposed_gene), transposed_gene)
# Merge the dataset
data_frame <- merge(metadata, transposed_gene_data, by = "participant_id")
# Ensure the gene expression values are numeric
data_frame[[gene_name]] <- as.numeric(data_frame[[gene_name]])
# Filter out NA values for the continuous covariate
data_frame[[continuous_covariate]] <- as.numeric(data_frame[[continuous_covariate]])
data_frame <- data_frame %>% filter(!is.na(!!sym(continuous_covariate)))
# Filter out NA values for categorical covariate
data_frame <- data_frame %>% filter((!!sym(categorical_covariate1) != 'unknown'))
filtered_data <- data_frame %>% filter((!!sym(categorical_covariate2) != 'unknown'))
# Histogram for gene expression
gene_histogram <- ggplot(filtered_data, aes(x = !!sym(gene_name))) +
    geom_histogram(binwidth = 0.05, color = "#1dddeb", fill = "#1dddeb", alpha = 0.7) +
    labs(x = gene_name,
         y = "Number",
         title = paste("Histogram of",
                       gene_name,
                       "Expression")) +
    custom_theme

# Save the histogram
ggsave(paste0("plots/", gene_name, "_histogram.png"),
       plot = gene_histogram, width = 16, height = 9)
# Print the histogram
print(gene_histogram)

# Scatter Plot with rainbow colors
scatter_plot <- ggplot(filtered_data,
                       aes(x = !!sym(continuous_covariate),
                           y = !!sym(gene_name),
                           color = !!sym(gene_name))) +
    geom_point(size = 2) +
    geom_rug(color = "white") +  # Add a rug plot
    scale_color_gradientn(colors = rainbow(7)) +  # Apply rainbow colors
    labs(x = continuous_covariate, y = gene_name, title = paste("Scatter Plot of",
                                                                gene_name,
                                                                "Expression vs",
                                                                continuous_covariate)) +
    custom_theme

# Save the scatter plot
ggsave(paste0("plots/", gene_name, "_scatter_plot.png"),
       plot = scatter_plot, width = 16, height = 9)
# Print the scatter plot
print(scatter_plot)
```

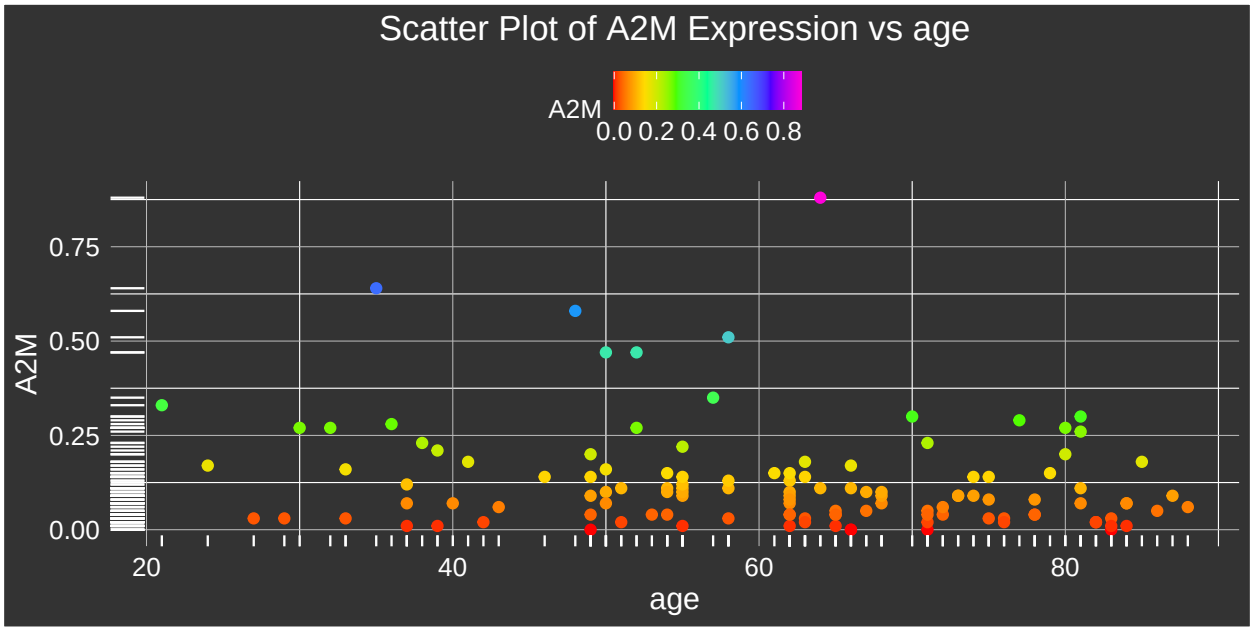
```
      # Define labels for the legend
      labels <- unique(filtered_data[[categorical_covariate2]])
      # Create the palette dynamically based on the labels
      color <- c("#66c2a5", "#fc8d62")
      palette <- setNames(color, labels)
      # https://www.statology.org/setnames-r/
      # Boxplot
      boxplot <- ggplot(filtered_data, aes(x = !!sym(categorical_covariate1),
                                           y = !!sym(gene_name),
                                           fill = !!sym(categorical_covariate2))) +
          geom_boxplot(outlier.color = "red", color = "white") +
          labs(x = categorical_covariate1, y = gene_name, fill = categorical_covariate2,
               title = paste("Boxplot of", gene_name, "Expression by",
                             categorical_covariate1, "and", categorical_covariate2)) +
          scale_fill_manual(values = palette, labels = labels) +
          custom_theme

      # Save the boxplot
      ggsave(paste0("plots/", gene_name, "_boxplot.png"),
             plot = boxplot, width = 16, height = 9)
      # Print the boxplot
      print(boxplot)
    }
}

create_plots('dataset/QBS103_GSE157103_series_matrix.csv',
             'dataset/QBS103_GSE157103_genes.csv',
             c("A1BG", "A2M", "AANAT"),
             "age", "sex",
             "disease_status")
```

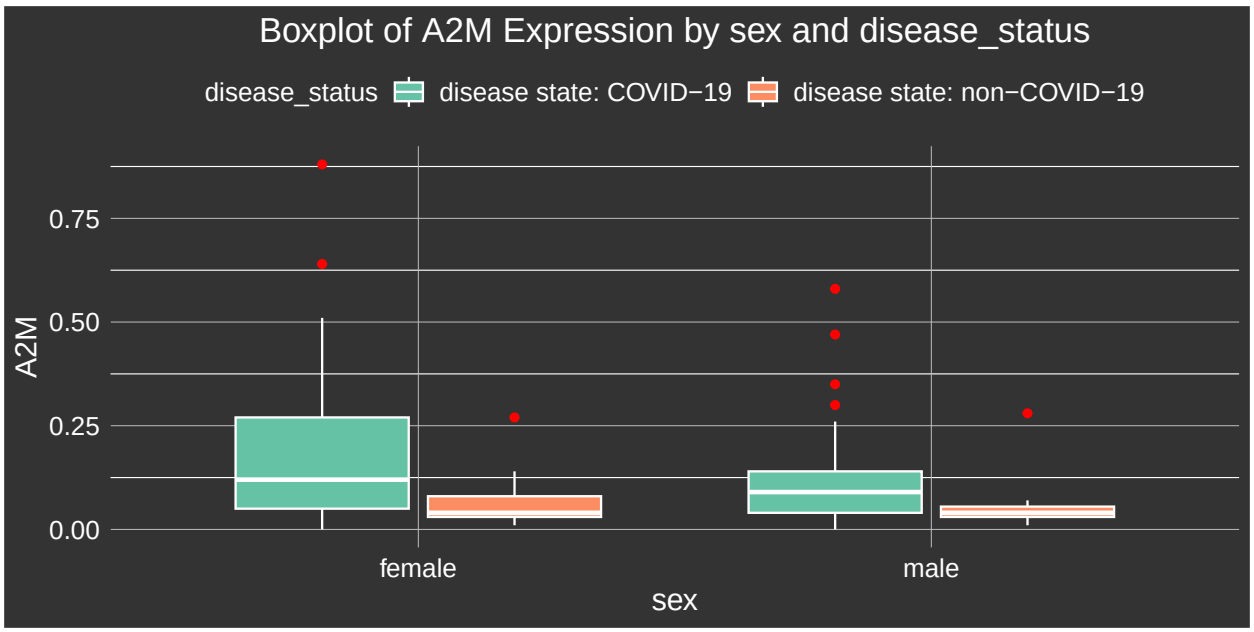


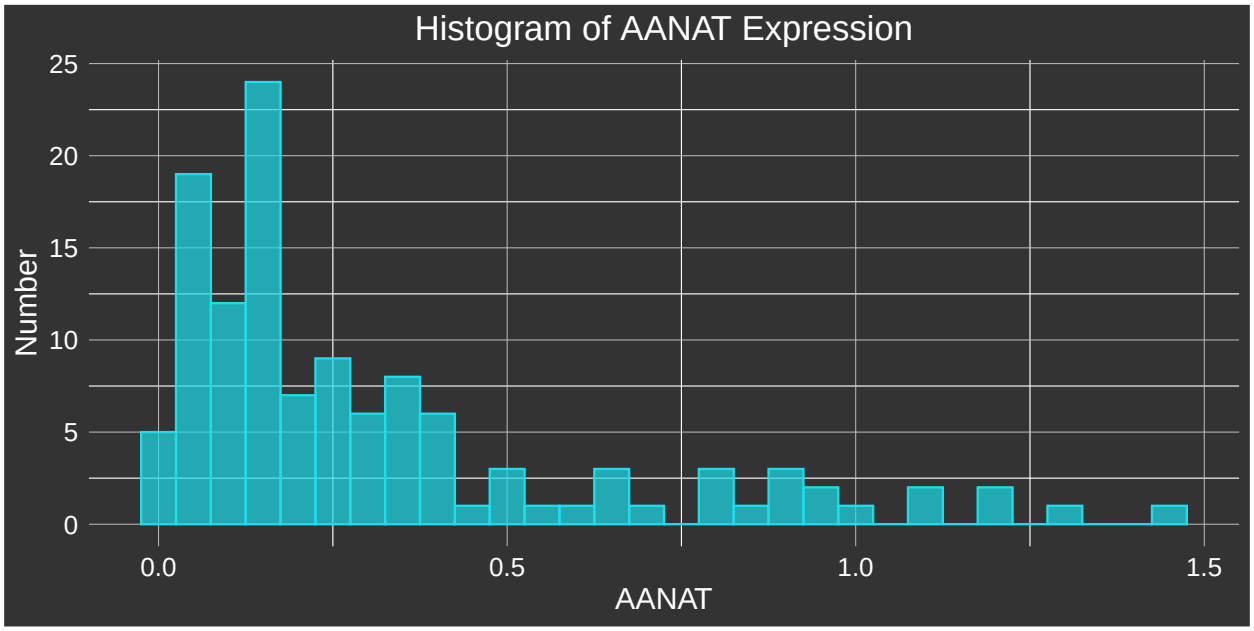

Scatter Plot of A1BG Expression vs age
A1BG
1
2
A1BG
0
1
2
20
40
60
80
age

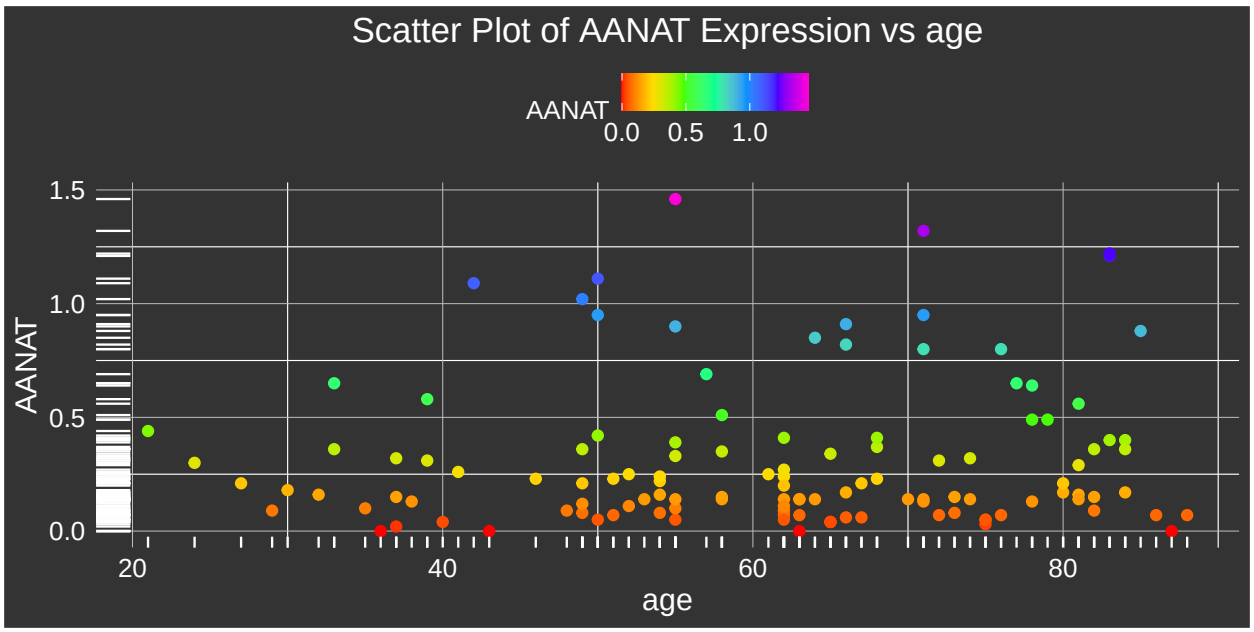

Boxplot of A1BG Expression by sex and disease_status
disease_status
disease state: COVID−19
disease state: non−COVID−19
A1BG
0
1
2
female
male
sex

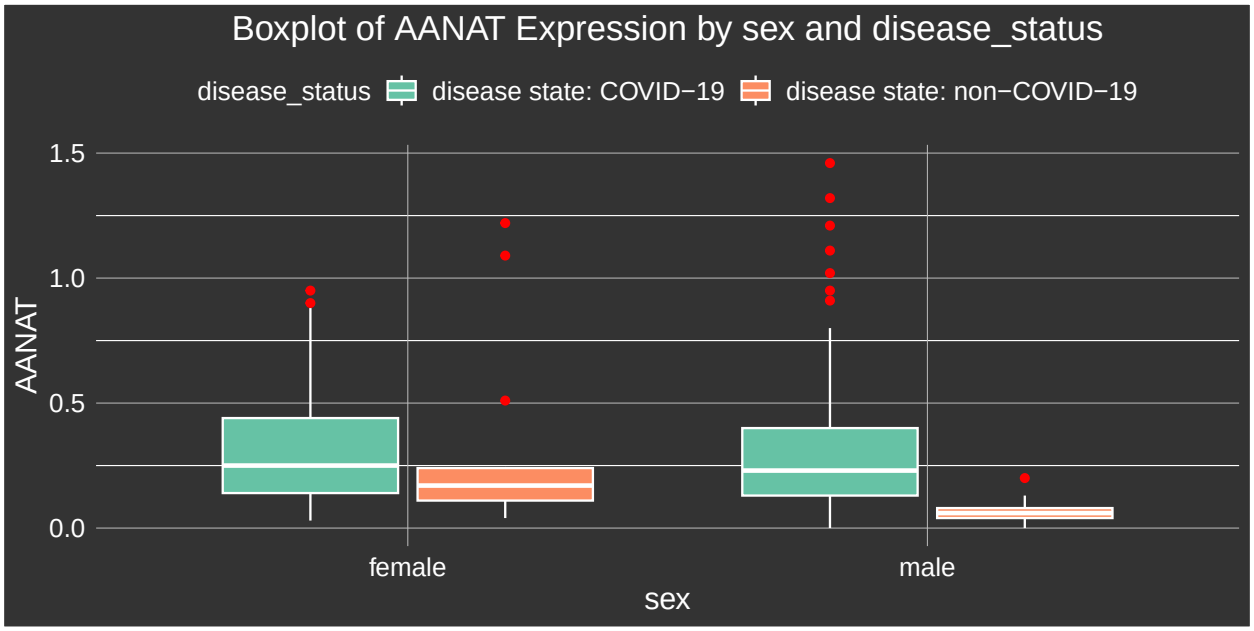

Histogram of A2M Expression
Number
30
20
10
0
0.00
0.25
0.50
0.75
A2M


Scatter Plot of A2M Expression vs age
A2M
0.0 0.2 0.4 0.6 0.8
A2M
0.75
0.50
0.25
0.00
20
40
60
80
age


Boxplot of A2M Expression by sex and disease_status
disease_status
disease state: COVID−19
disease state: non−COVID−19
A2M
0.75
0.50
0.25
0.00
female
male
sex


Histogram of AANAT Expression
Number
25
20
15
10
5
0
0.0
0.5
1.0
1.5
AANAT


Scatter Plot of AANAT Expression vs age
AANAT
0.0
0.5
1.0
1.5
1.0
0.5
0.0
AANAT
20
40
60
80
age


Boxplot of AANAT Expression by sex and disease_status
disease_status
disease state: COVID−19
disease state: non−COVID−19
1.5
1.0
0.5
0.0
AANAT
female
male
sex