Test Submission #2

Eric Wang

2024-08-08

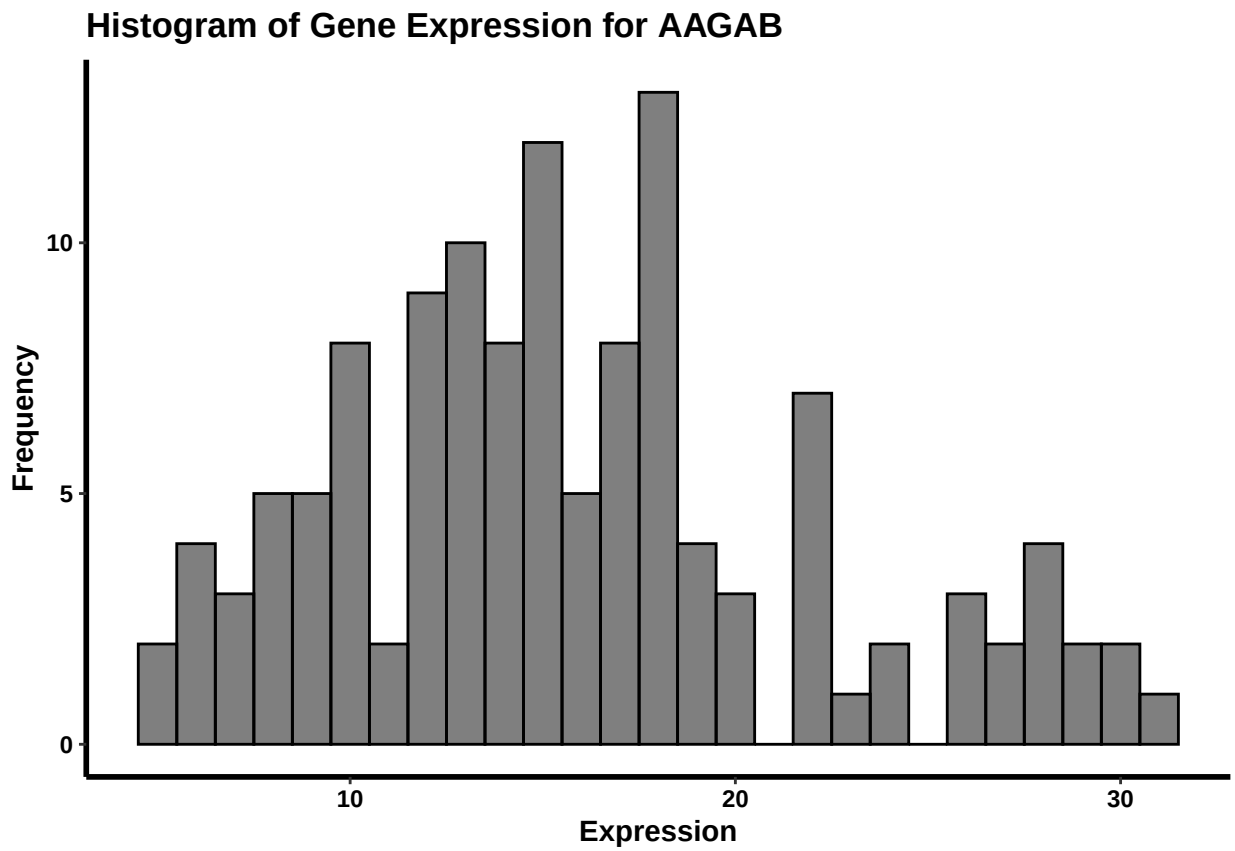
```
library(tidyverse)
```

```
## -- 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
```

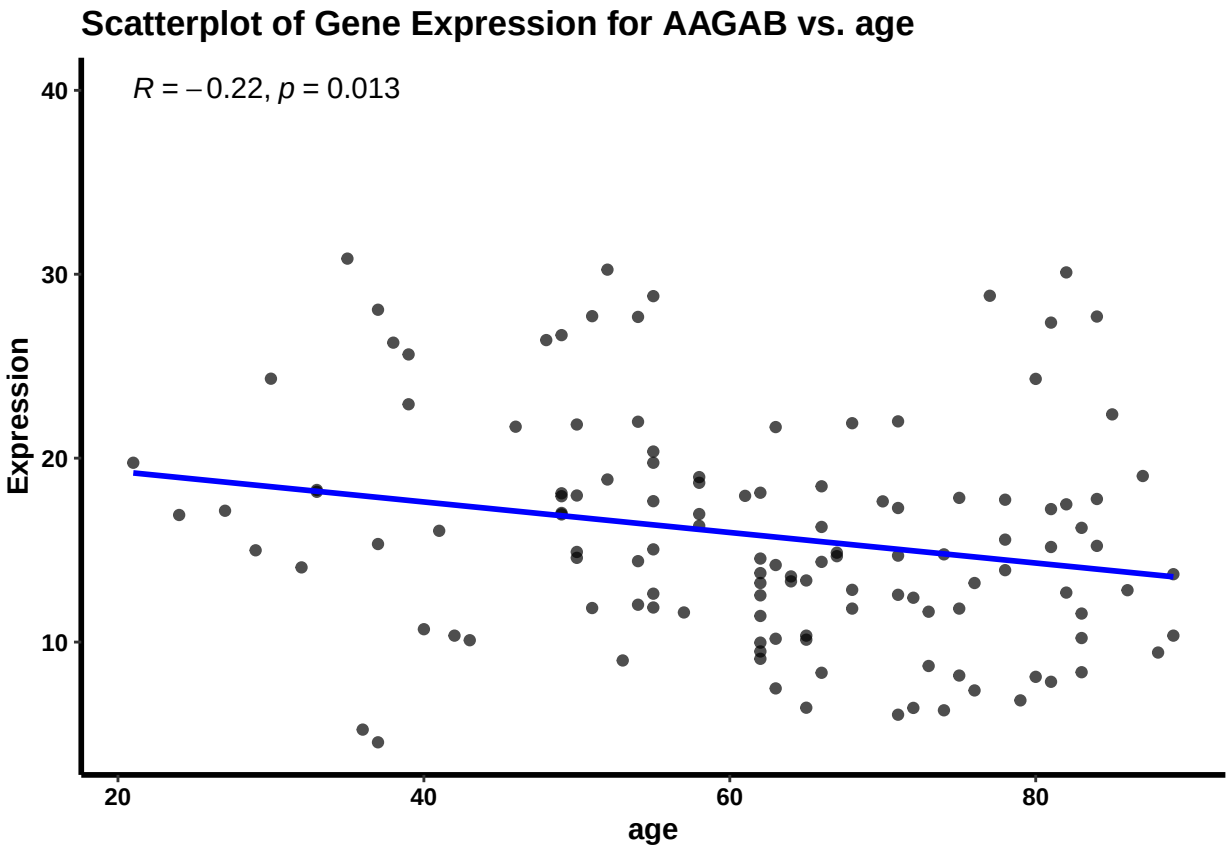
```
library(ggpubr)

# Set the working directory
setwd("/Users/wangdeyao/Downloads/QBS103")

# Load the data from CSV files
df1 <- read.csv("QBS103_GSE157103_genes.csv")
df2 <- read.csv("QBS103_GSE157103_series_matrix.csv")

# Define a custom theme for ggplot
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", color = "black"),
  axis.text = element_text(face = "bold", colour = "black"),
  legend.position = 'bottom'
)


# Filter and transform the data for each gene
data_long <- df1 %>%
```

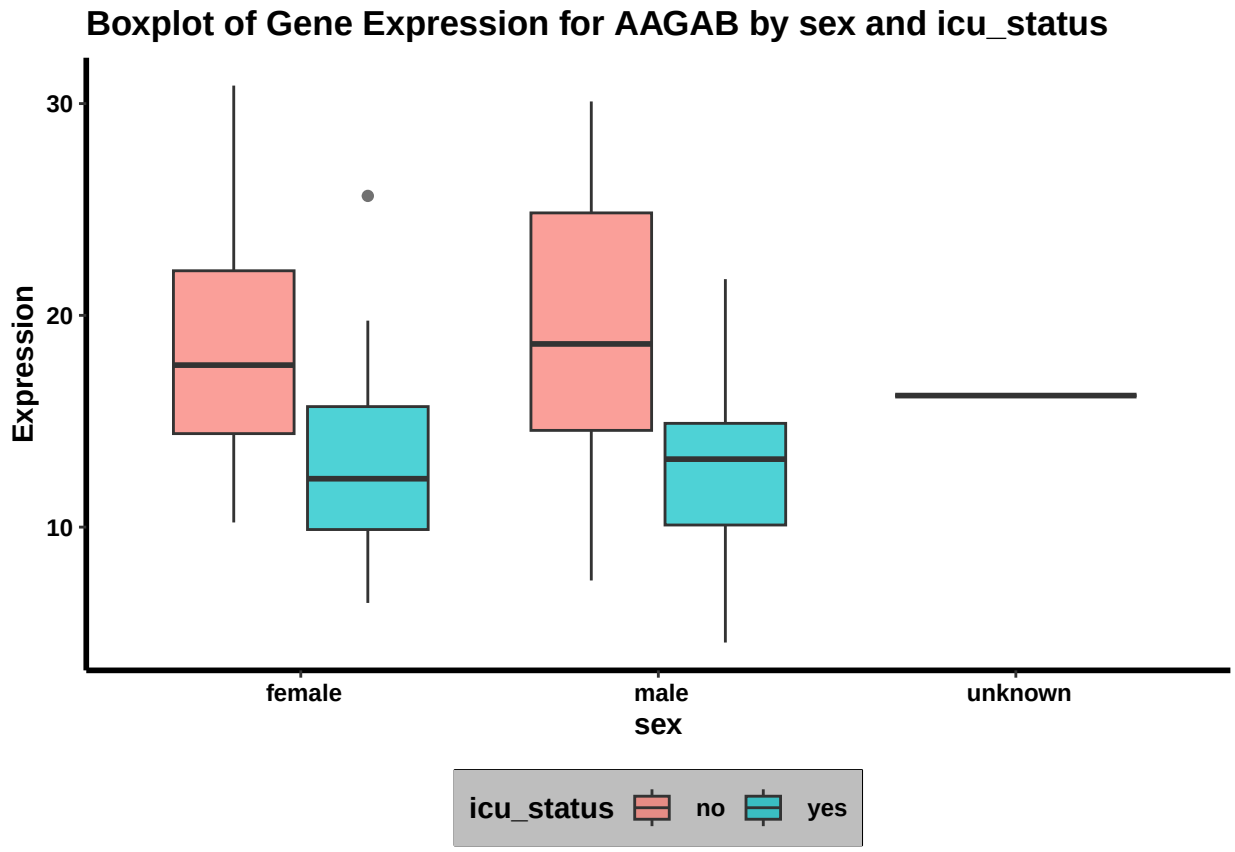
```
  gather(
    key = "Sample",
    value = "Expression",
    COVID_01_39y_male_NonICU:NONCOVID_26_36y_male_ICU
  )

# Merge the datasets on the 'Sample' and 'participant_id' columns
df <- merge(data_long, df2, by.x = "Sample", by.y = "participant_id")

# Function to create plots for a given list of genes
plots <- function(data_frame,
                  gene_list,
                  cont_covariate,
                  cat_covariate1,
                  cat_covariate2) {
  for (gene in gene_list) {
    # Filter and transform the data for each gene
    merged_data <- data_frame %>% filter(X == gene) %>%
      select("Sample",
             "Expression",
             cont_covariate,
             cat_covariate1,
             cat_covariate2)
    # Remove '>' characters and convert the continuous covariate to numeric
    merged_data[[cont_covariate]] <- gsub(">", "", merged_data[[cont_covariate]])
    merged_data[[cont_covariate]] <- as.numeric(
      as.character(merged_data[[cont_covariate]]))
    merged_data <- merged_data %>% drop_na()

    # Create a histogram of gene expression
    hist_plot <- ggplot(merged_data, aes(x = Expression)) +
      geom_histogram(
        binwidth = 1,
        fill = "black",
        color = "black",
        alpha = 0.5
      ) +
      labs(
        title = paste("Histogram of Gene Expression for", gene),
        x = "Expression",
        y = "Frequency"
      ) +
      newTheme
    print(hist_plot)

    # Create a scatter plot of gene expression vs. the continuous covariate
    scatter_plot <- ggplot(merged_data,
                           aes_string(x = cont_covariate, y = "Expression")) +
      geom_point(color = "black", alpha = 0.7) +
      geom_smooth(method = "lm",
                  color = "blue",
                  se = FALSE) +
      stat_cor(label.y = 40) +
```

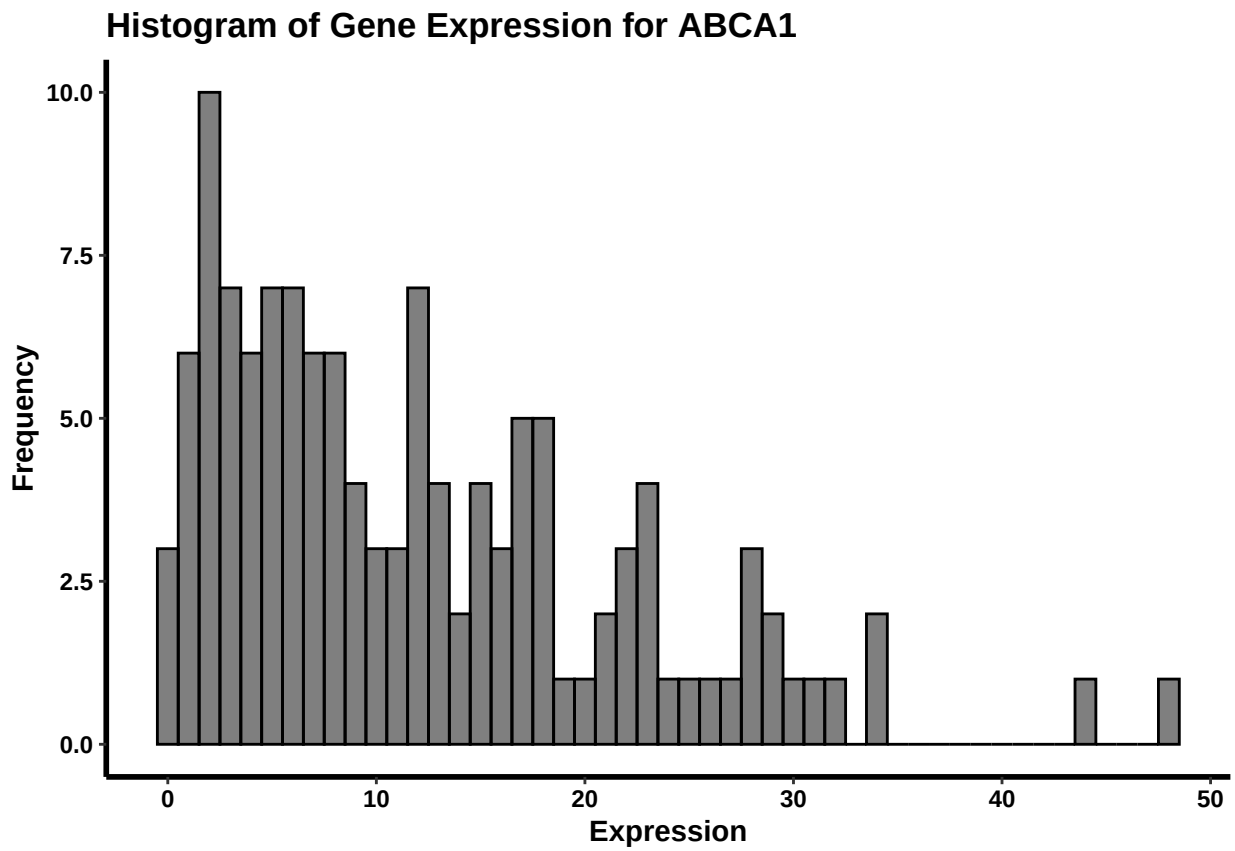
```
      labs(
        title = paste(
          "Scatterplot of Gene Expression for",
          gene,
          "vs.",
          cont_covariate
        ),
        y = "Expression",
        x = cont_covariate
      ) +
      newTheme
    print(scatter_plot)

    # Create a box plot of gene expression by the categorical covariates
    box_plot <- ggplot(
      merged_data,
      aes_string(x = cat_covariate1, y = "Expression", fill = cat_covariate2)
    ) +
      geom_boxplot(alpha = 0.7) +
      labs(
        title = paste(
          "Boxplot of Gene Expression for",
          gene,
          "by",
          cat_covariate1,
          "and",
          cat_covariate2
        ),
        x = cat_covariate1,
        y = "Expression"
      ) +
      newTheme
    print(box_plot)
  }
}

# List of genes to create plots for
gene_list <- list("AAGAB", "ABCA1", "AAAS")

# Call the function to create plots for the given list of genes
plots(df, gene_list, "age", "sex", "icu_status")%>% suppressWarnings()
```




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

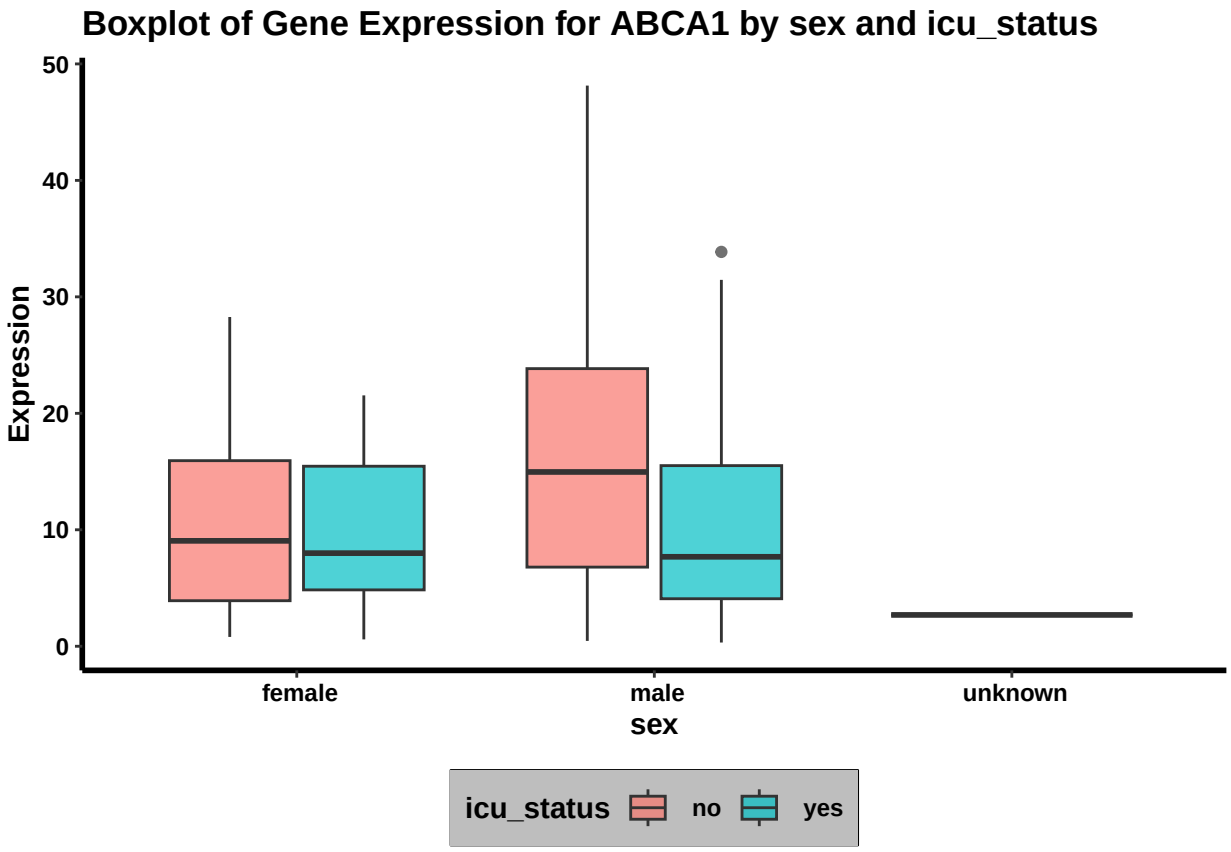

Scatterplot of Gene Expression for AAGAB vs. age
R = −0.22, p = 0.013
Expression
40
30
20
10
20
40
60
80
age

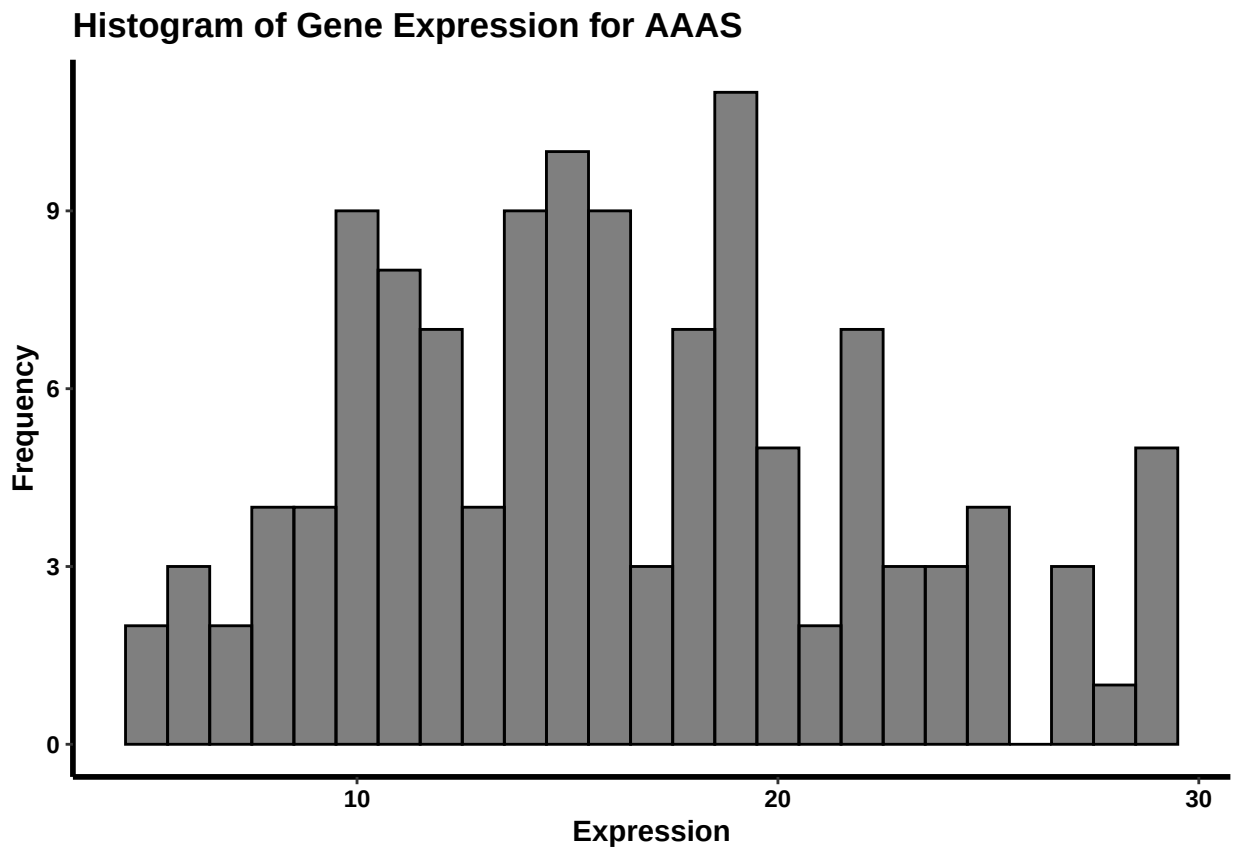

Boxplot of Gene Expression for AAGAB by sex and icu_status
Expression
30
20
10
female
male
unknown
sex
icu_status
no
yes




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

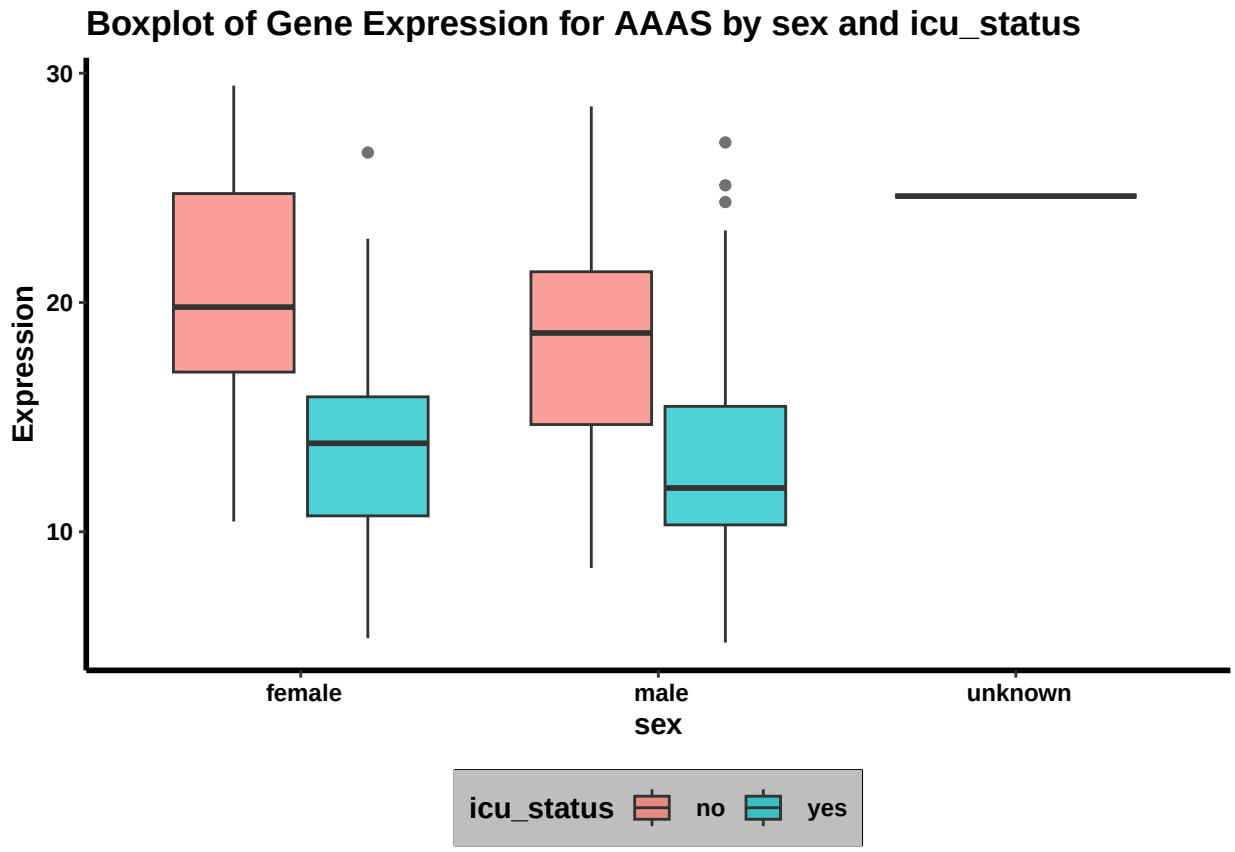

Scatterplot of Gene Expression for ABCA1 vs. age
R = –0.11, p = 0.22
Expression
age
50
40
30
20
10
0
20
40
60
80


Boxplot of Gene Expression for ABCA1 by sex and icu_status
Expression
50
40
30
20
10
0
female
male
unknown
sex
icu_status
no
yes




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


Scatterplot of Gene Expression for AAAS vs. age
R = – 0.19, p = 0.036
Expression
40
30
20
10
20
40
60
80
age


Boxplot of Gene Expression for AAAS by sex and icu_status
Expression
30
20
10
female
male
unknown
sex
icu_status
no
yes