Final Project

2023-07-24

R Markdown

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see http://rmarkdown.rstudio.com.

When you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document. You can embed an R code chunk like this:

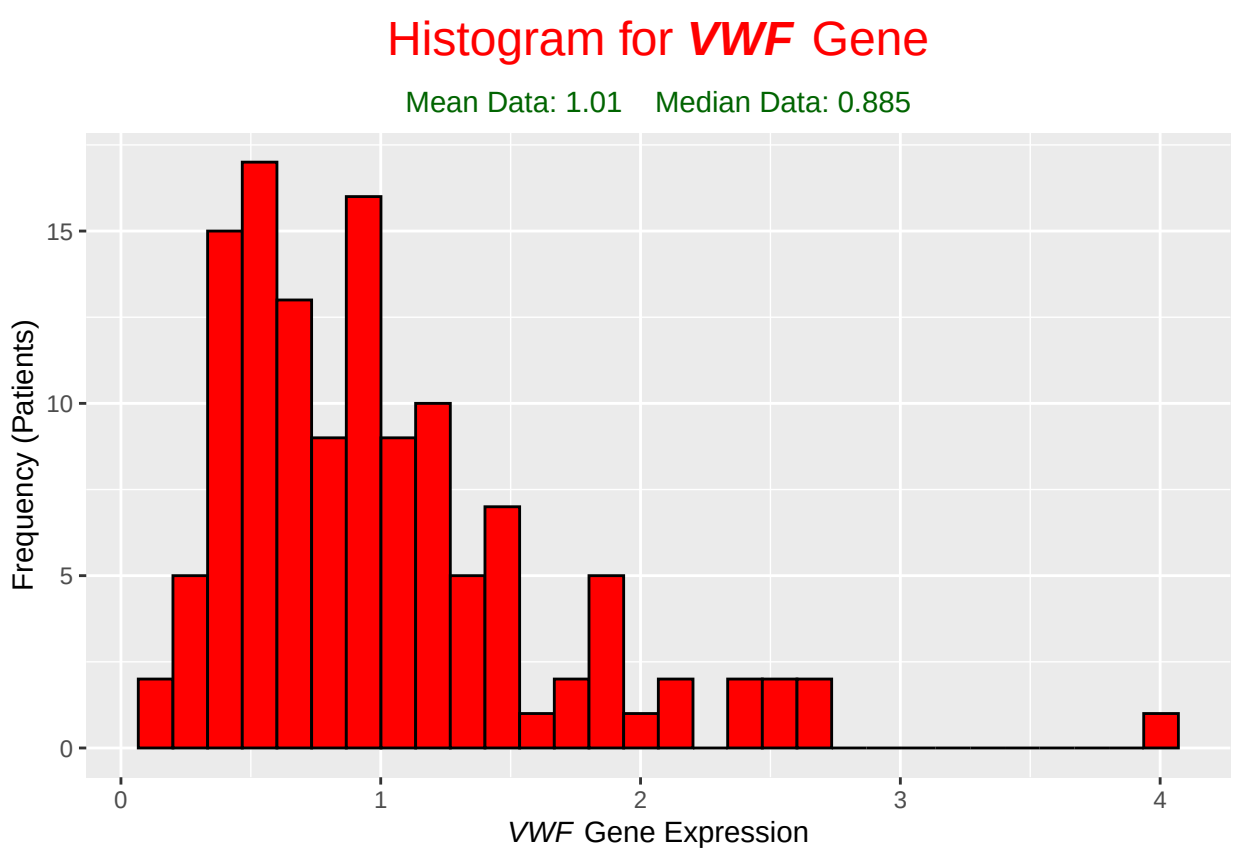
```
plot_creation <- function(data_gene, gene_list, covariate, categorical_covariate_list) {
    library(dplyr)
    library(ggplot2)
    library(ggpubr)
    meta_data <- read.csv(file = "QBS103_finalProject_metadata.csv")
    gene1 = toString(gene_list[1])
    category1 = tools::toTitleCase(gsub("_", " ", toString(categorical_covariate_list[1])))
    category2 = tools::toTitleCase(gsub("_", " ", toString(categorical_covariate_list[2])))
    linked_data1 <- data.frame(
        "Participant_ID" = meta_data$participant_id,
        "Age" = strtoi(meta_data[[covariate]]),
        "Categorical2" = meta_data[[toString(categorical_covariate_list[2])]],
        "Categorical" = meta_data[[toString(categorical_covariate_list[1])]],
        "Gene_Value" = as.numeric(data_gene[[gene1]])
      )
    # Histogram for gene expressions
    gene_hist <- ggplot(linked_data1, aes(x = Gene_Value)) +
      geom_histogram(fill = "red", color = "black") +
      labs(
        title = substitute(paste("Histogram for ", bolditalic(gene1), " Gene")),
        subtitle = paste("Mean Data:", round(mean(linked_data1$Gene_Value, na.rm = TRUE), 2), "   Median
                         median(linked_data1$Gene_Value, na.rm = TRUE)),
        x = substitute(paste(italic(gene1), " Gene Expression")),
        y = "Frequency (Patients)"
      ) +
      theme(plot.title = element_text(size = 18, color = "red", hjust = 1/2),
            plot.subtitle = element_text(color = "darkgreen", hjust = 1/2))
      # Added a mean and median value for the BPI gene expression and ignore all the NA values

    # Scatter plots
    linked_remove_null1 <- linked_data1[complete.cases(linked_data1),]
    gene_scatter <- ggplot(linked_remove_null1, aes(x = Age, y = Gene_Value)) +
      geom_point(color = "red") +
      labs(
        title = substitute(paste(bolditalic(gene1), " Gene Expression vs Continuous Covariat (Age)")),
        x = expression(bold("Age")),
```

```
      y = substitute(paste(italic(gene1), " Gene Expression"))
    ) +
    theme(plot.title = element_text(size = 18, color = "darkgreen", hjust = 1/2),
          panel.background = element_rect(fill = 'lightblue', color = 'black'),
          panel.grid.major = element_line(color = 'black', linetype = 'dotted')) +
    # There are no one outside the age of 10 or 100 so set the limit between these 2 values
    scale_x_continuous(limits = c(10, 100))
  # Box plot
  # Remove the unknown sex value to clean up the data
  linked_remove_unknown1 = linked_data1[which(linked_data1$Categorical2 != " unknown"), ]
  gene_icuplot <- ggplot(linked_remove_unknown1, aes(x = Categorical, y = Gene_Value, color = Categor
    geom_boxplot() +
    labs(title = substitute(paste(bolditalic(gene1), " Gene Expression vs ", bolditalic(category1))),
      x = substitute(paste(bold(category1))),
      y = substitute(paste(bolditalic(gene1), " Gene Expression"))
    ) +
    theme(plot.title = element_text(size = 16, color = "darkblue", hjust = 1/2),
          panel.background = element_rect(fill = 'lightblue', color = 'black'),
          panel.grid.major = element_line(color = 'black', linetype = 'dotted')) +
    # Add a jitter
    geom_jitter() +
    guides(color = guide_legend(title = category2)) +
    scale_color_manual(values = c("red","darkgreen"))


  gene_sexplot <- ggplot(linked_remove_unknown1, aes(x = Categorical2, y = Gene_Value, color = Catego
    geom_boxplot() +
    labs(title = substitute(paste(bolditalic(gene1), " Gene Expression vs ", bolditalic(category2))),
      x = substitute(paste(bold(category2))),
      y = substitute(paste(bolditalic(gene1), " Gene Expression"))
    ) +
    theme(plot.title = element_text(size = 18, color = "darkblue", hjust = 1/2),
          panel.background = element_rect(fill = 'lightblue', color = 'black'),
          panel.grid.major = element_line(color = 'black', linetype = 'dotted')) +
    # Add a jitter
    geom_jitter() +
    guides(color = guide_legend(title = category1)) +
    scale_color_manual(values = c("red","darkgreen"))

  print(list(gene_hist, gene_scatter, gene_icuplot, gene_sexplot))
  #return(list(gene_hist, gene_scatter, gene_icuplot, gene_sexplot))
}
```

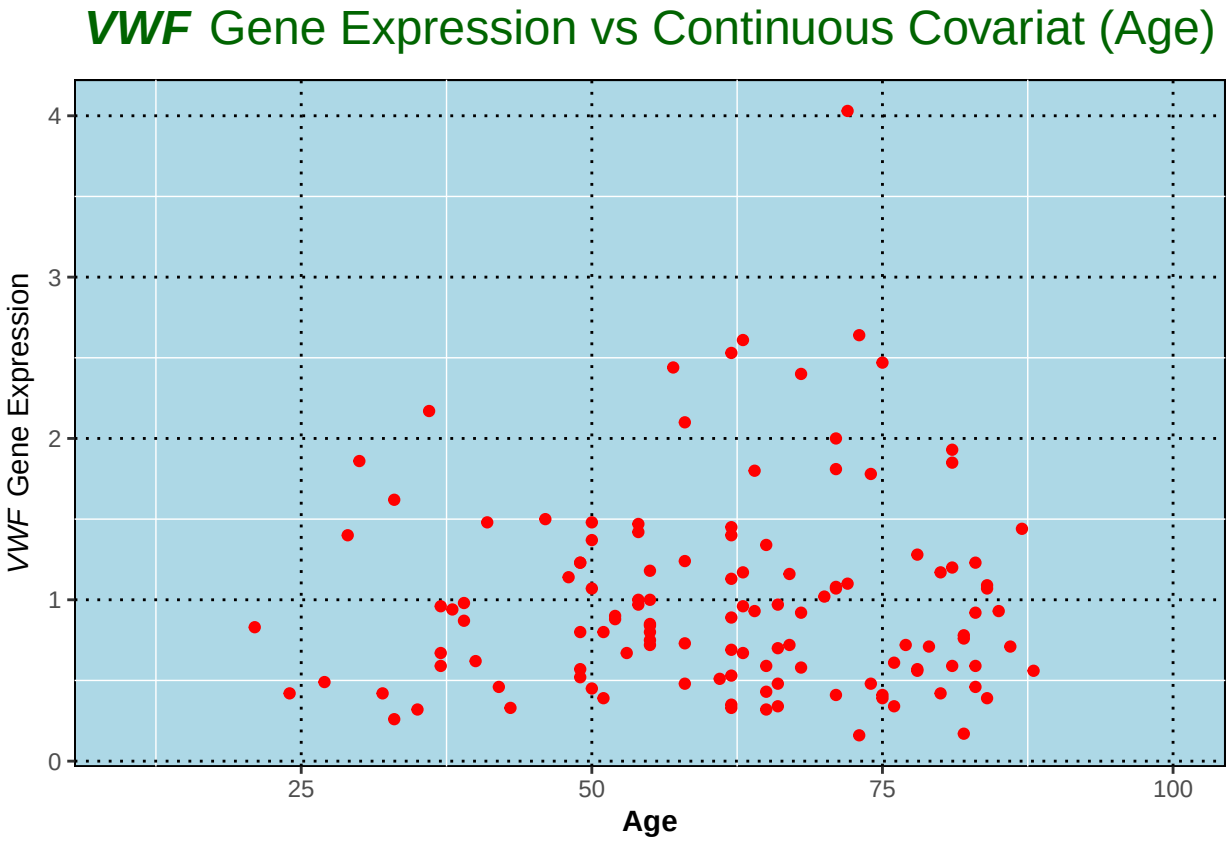
```
# This function is to convert the gene expression csv to dataframe to be the input dataframe
csv_to_dataframe <- function(gene_expression) {
  # Transpose the dataset in order to link the 2 sets of data
  gene_transposed <- data.frame(cbind(names(gene_expression), t(gene_expression)))
  rownames(gene_transposed) <- NULL
  colnames(gene_transposed) <- gene_transposed[1, ]
  # delete the first redundant row to match the entries
  gene_final <- gene_transposed[-1, ]
  # Create a data frame with the participant ID, gene BPI, one continuous covariate (age) and
  # two categorical covariates (sex and ICU status)
```

```
    return(gene_final)
}
# Calling csv_to_dataframe function to convert to dataframe
gene_expression <- read.csv(file = "QBS103_finalProject_geneExpression.csv")
gene_input <- csv_to_dataframe(gene_expression)

# Create the genes list and 2 categorical covariate list to be looked at
gene_list <- list("VWF", "AAAS", "BPI")
categorical_covariate_list <- list("mechanical_ventilation", "sex", "icu_status")
continuous_covariate <- "age"

# Calling the plot_creation function with four parameters
#plot_creation(gene_input, list("AAAS"), continuous_covariate, categorical_covariate_list)
for (x in 1:length(gene_list)) {
  plot_creation(gene_input, gene_list[x], continuous_covariate, categorical_covariate_list)
}
```

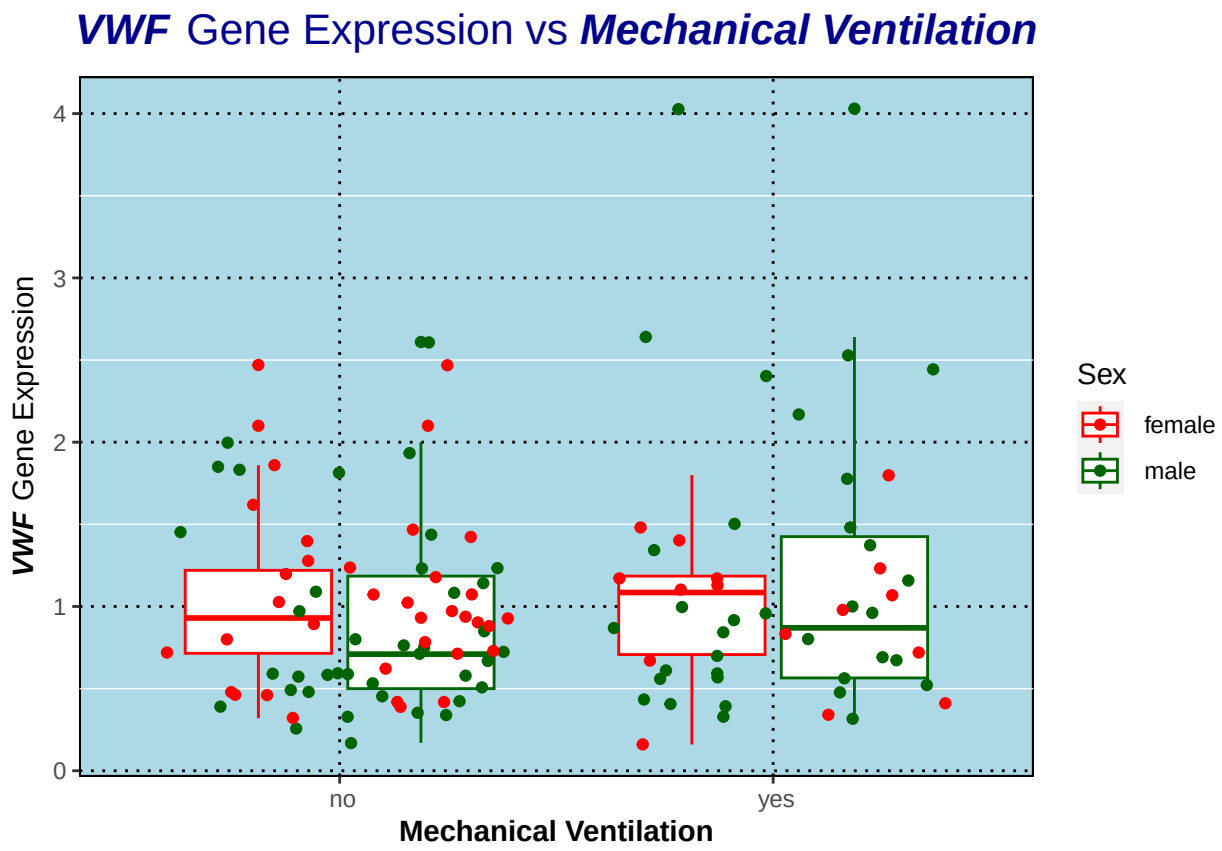
```
##
## Attaching package: 'dplyr'

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

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

##
## Attaching package: 'ggpubr'

## The following object is masked _by_ '.GlobalEnv':
##
##     gene_expression

## [[1]]

## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```


Histogram for VWF Gene
Mean Data: 1.01 Median Data: 0.885
Frequency (Patients)
15
10
5
0
0
1
2
3
4
VWF Gene Expression

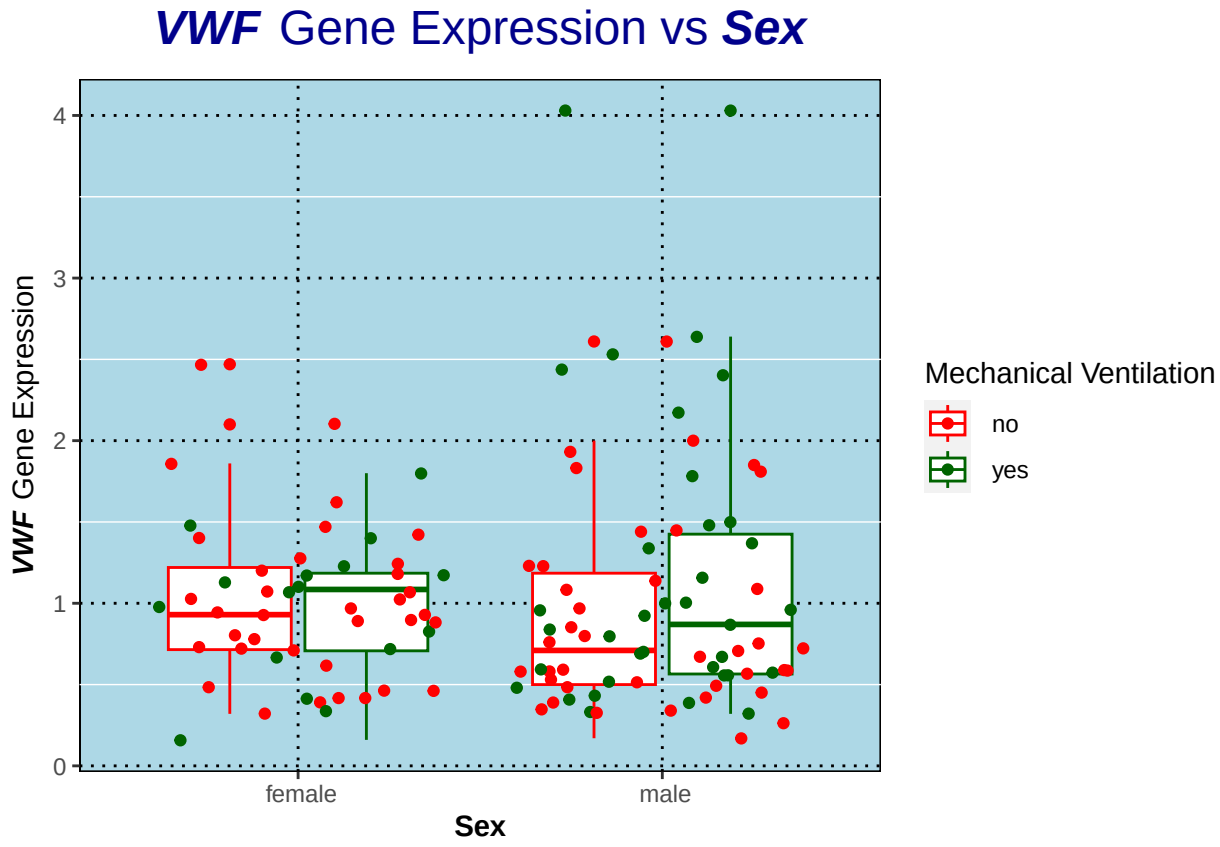
```
##
## [[2]]
```


VWF Gene Expression vs Continuous Covariat (Age)
VWF Gene Expression
Age
0
1
2
3
4
25
50
75
100

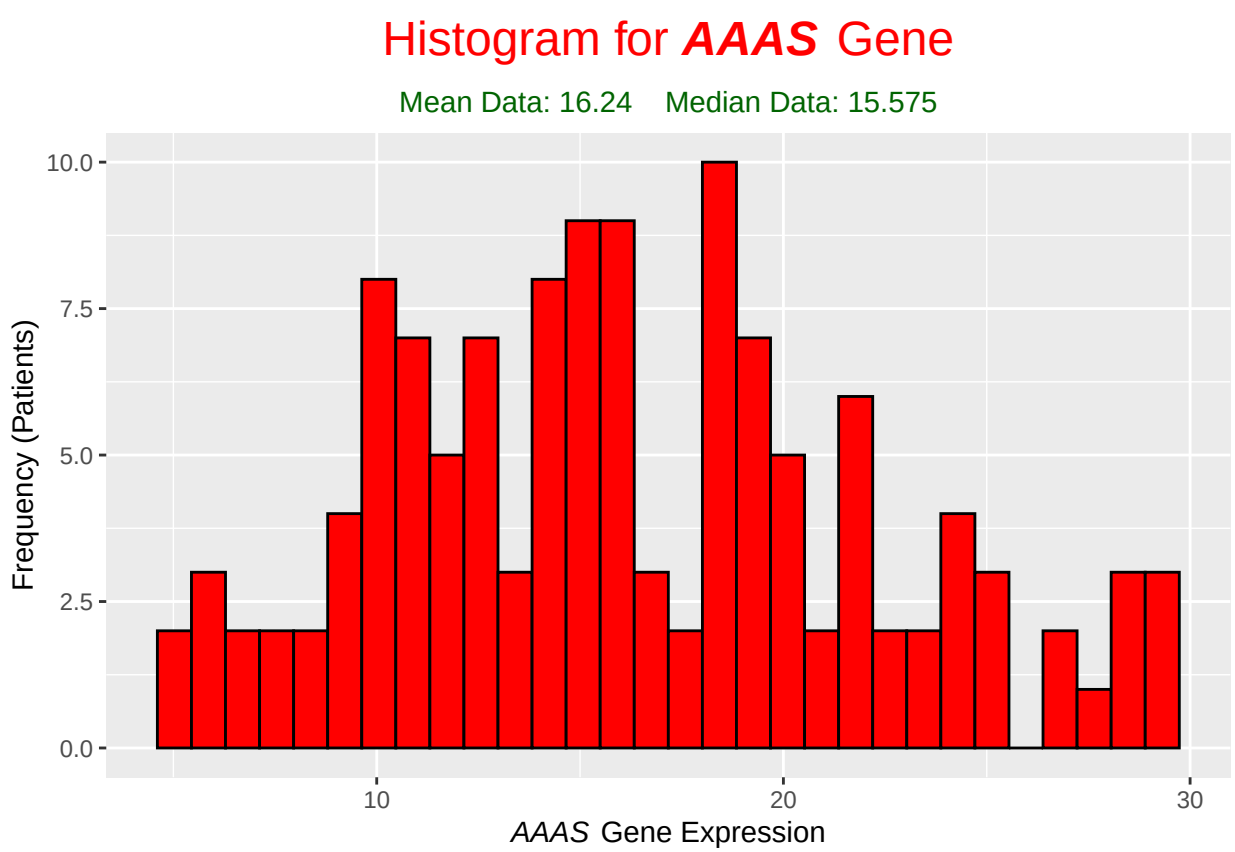
```
##
## [[3]]
```


VWF Gene Expression vs Mechanical Ventilation
VWF Gene Expression
0
1
2
3
4
no
yes
Mechanical Ventilation
Sex
female
male

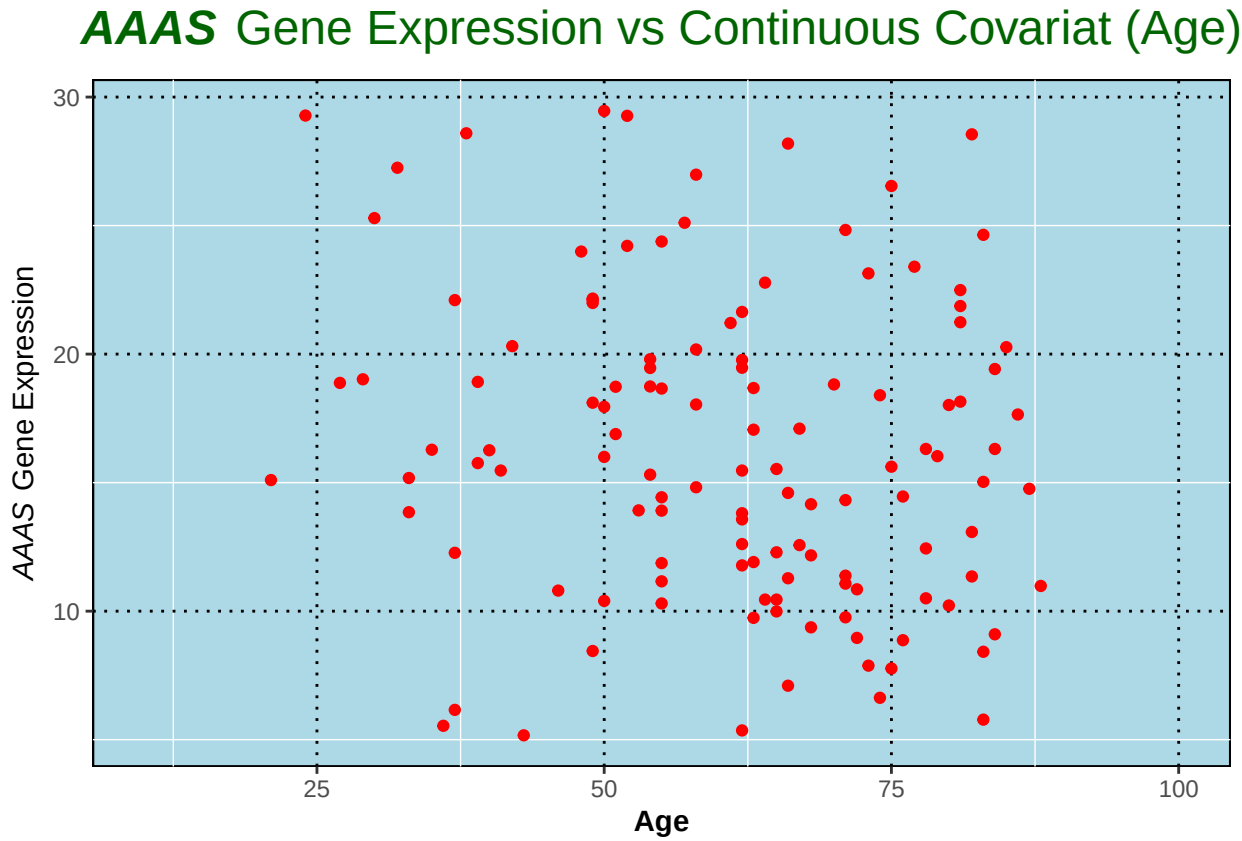
```
##
## [[4]]
```



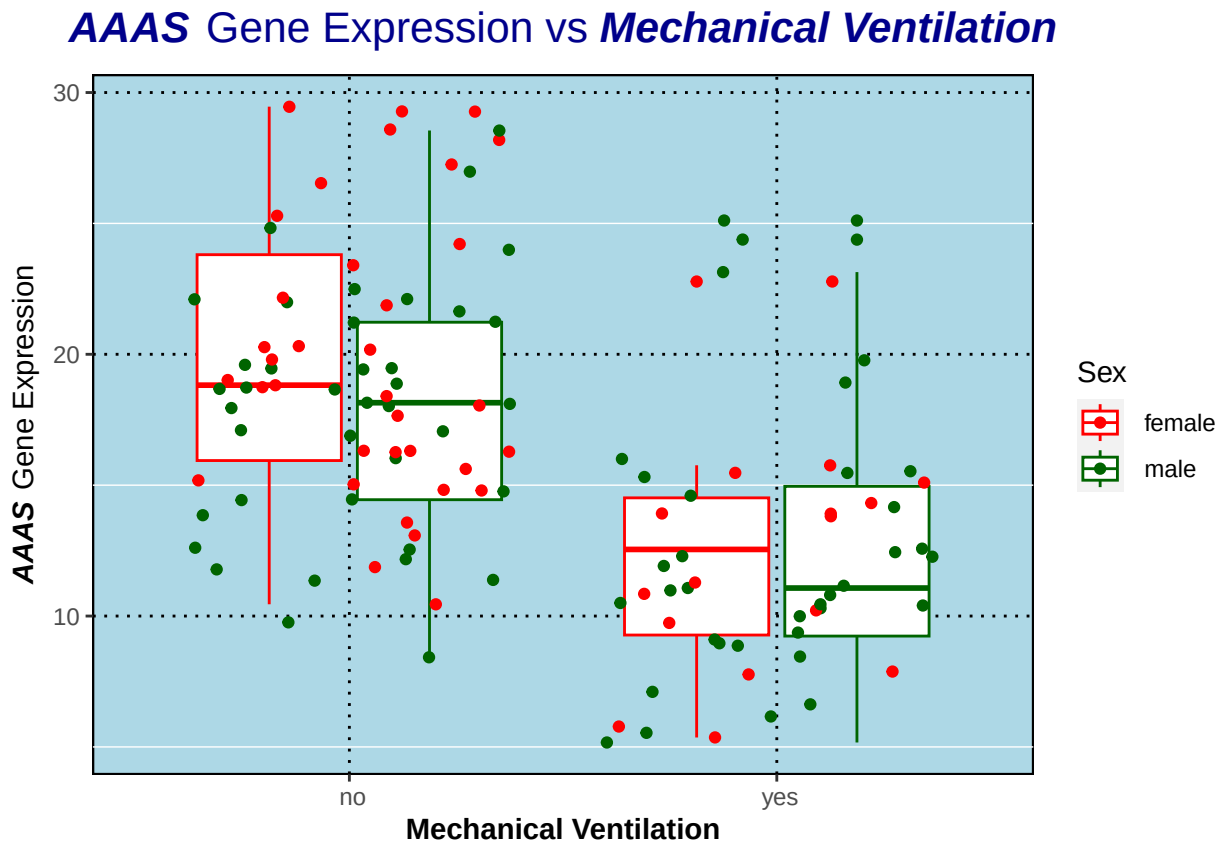

```
##
## [[1]]

## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```


Histogram for AAAS Gene
Mean Data: 16.24 Median Data: 15.575
Frequency (Patients)
10.0
7.5
5.0
2.5
0.0
10
20
30
AAAS Gene Expression

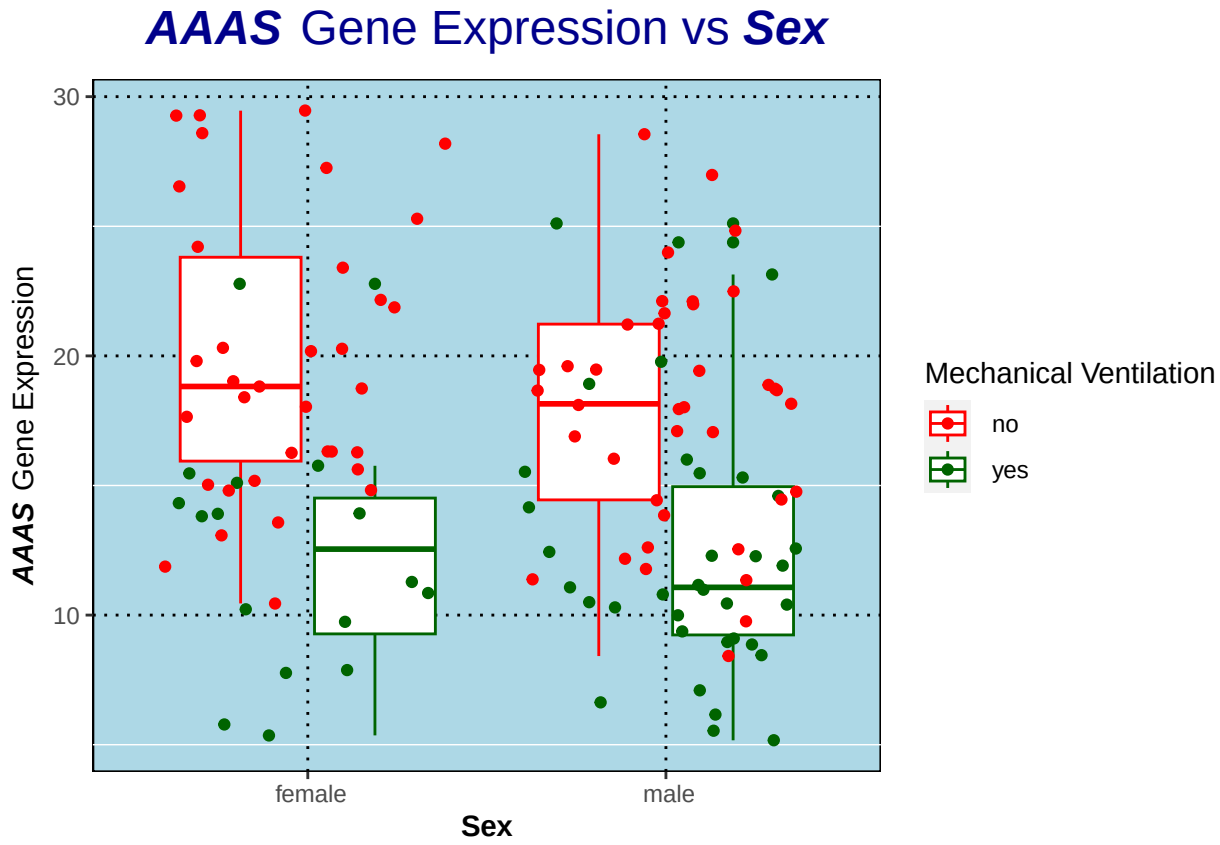
```
##
## [[2]]
```


AAAS Gene Expression vs Continuous Covariat (Age)
AAAS Gene Expression
30
20
10
25
50
75
100
Age

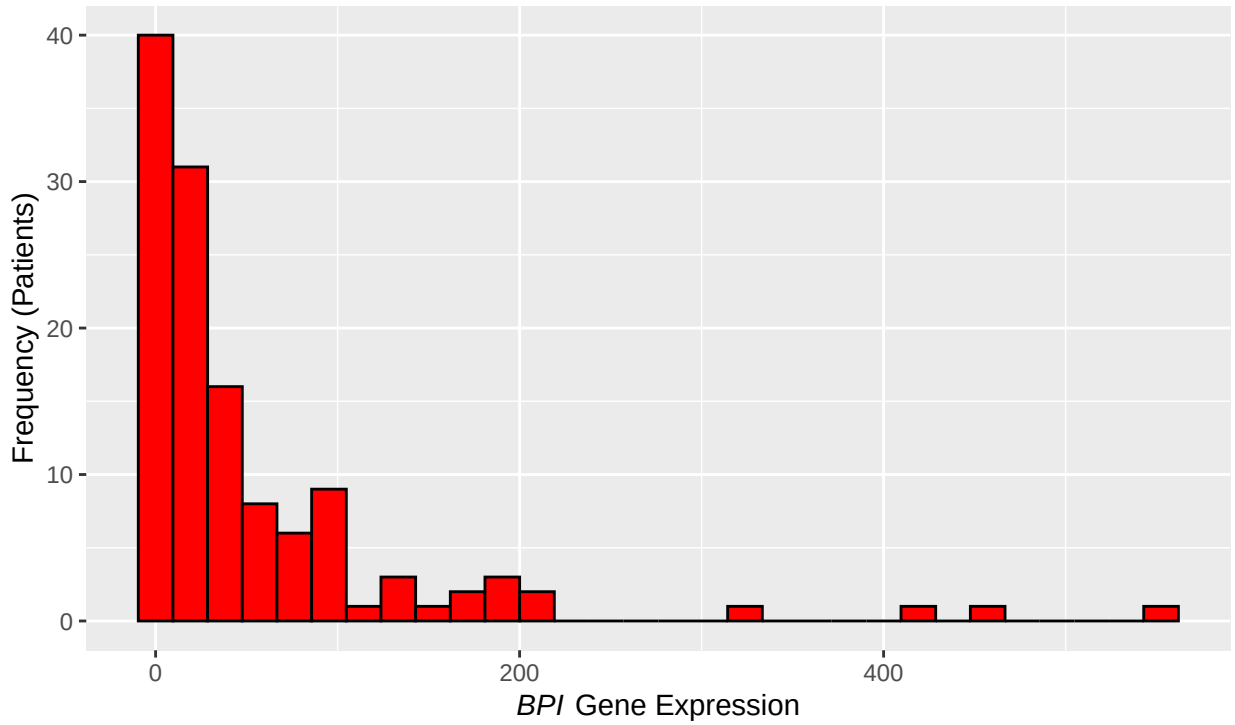
```
##
## [[3]]
```


AAAS Gene Expression vs Mechanical Ventilation
AAAS Gene Expression
30
20
10
no
yes
Mechanical Ventilation
Sex
female
male

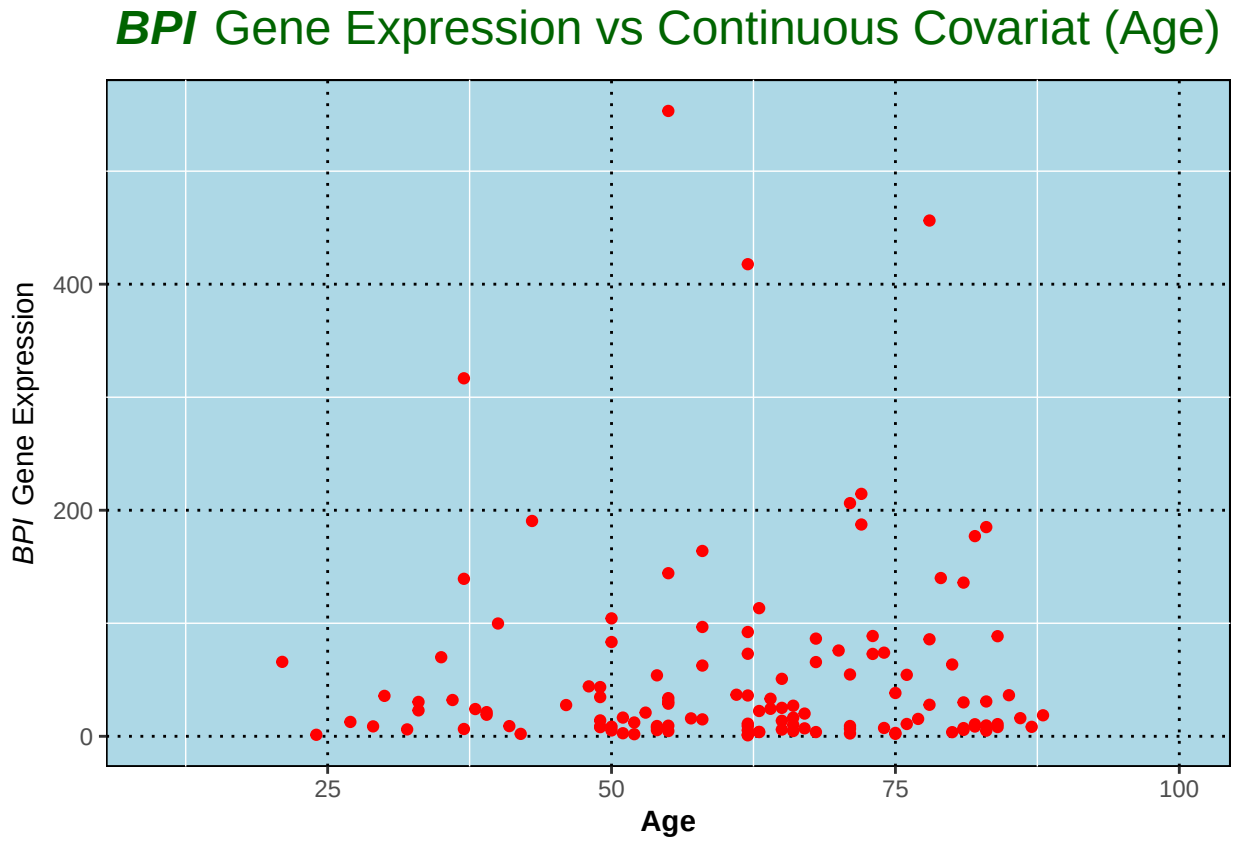
```
##
## [[4]]
```



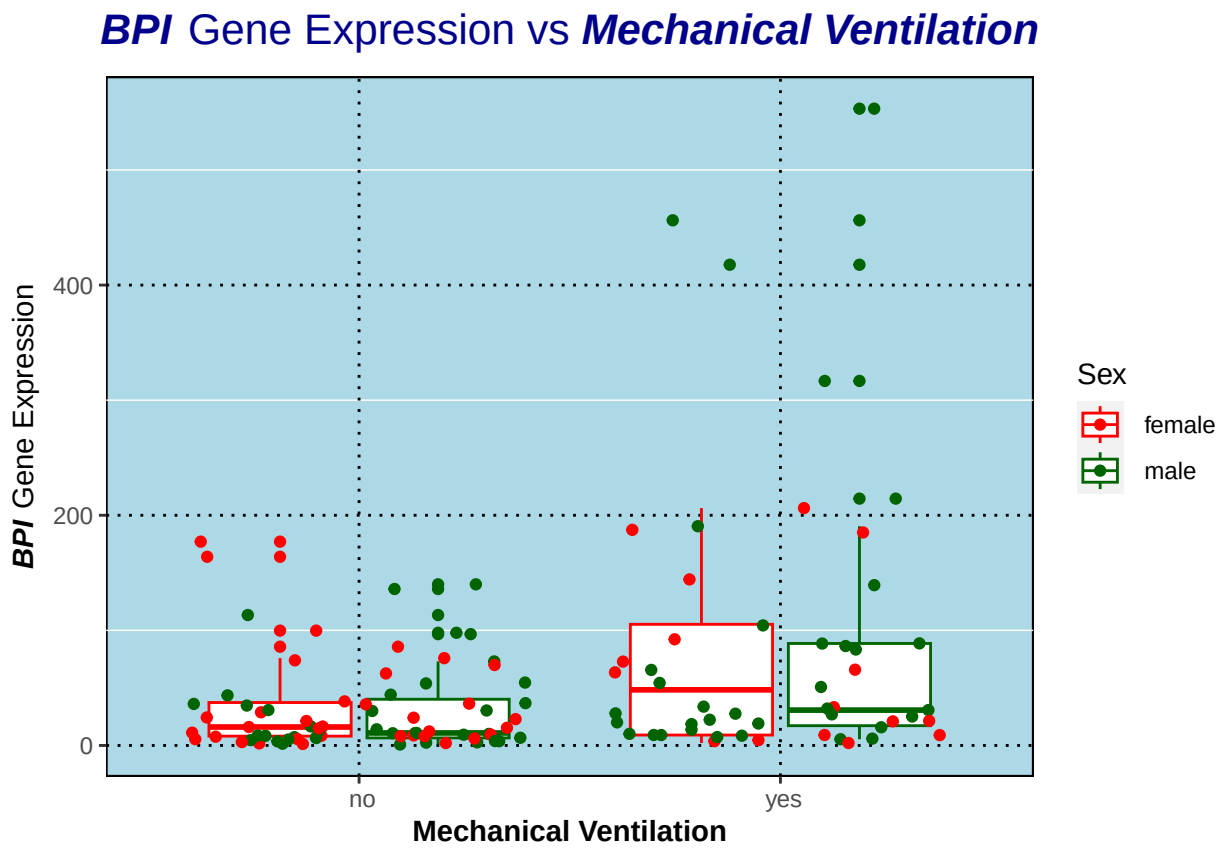

```
##
## [[1]]

## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```


Histogram for BPI Gene
Mean Data: 54.2 Median Data: 21.86
Frequency (Patients)
40
30
20
10
0
0
200
400
BPI Gene Expression

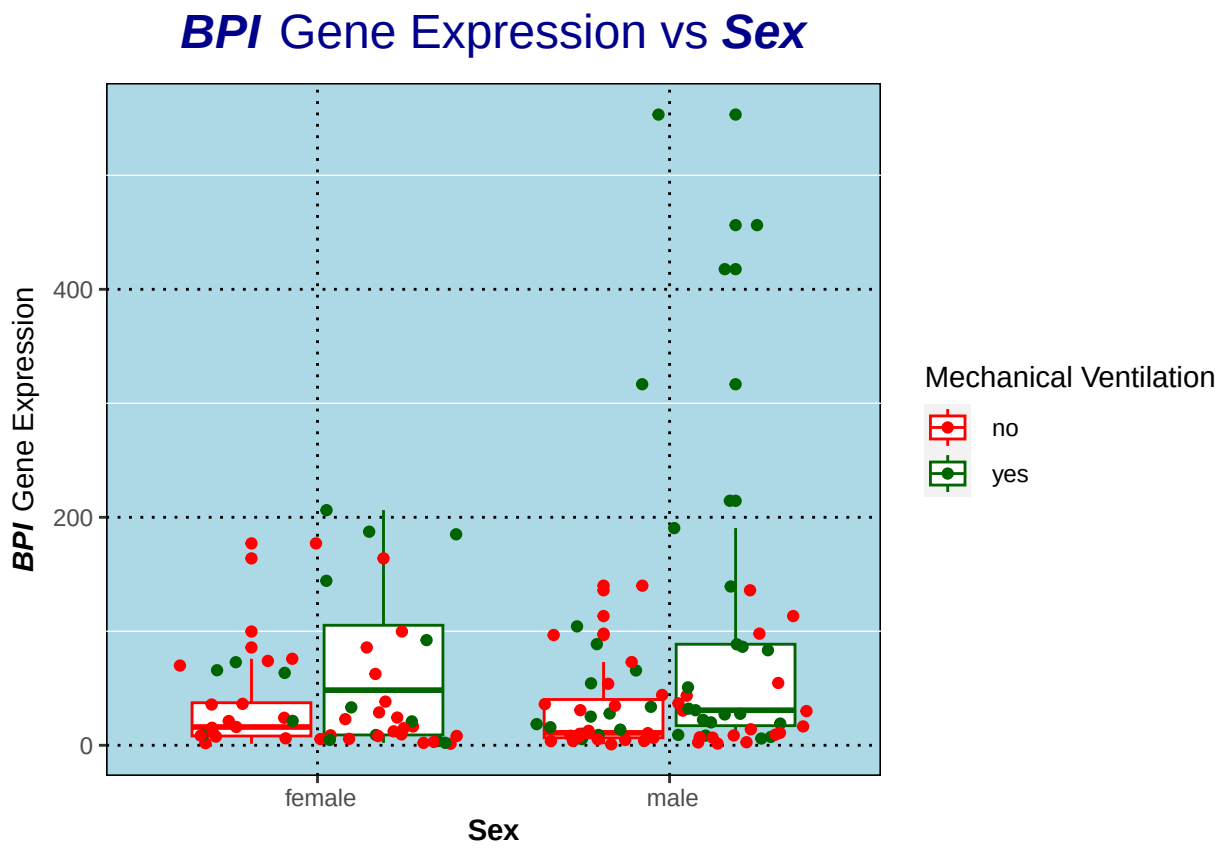
```
##
## [[2]]
```


BPI Gene Expression vs Continuous Covariat (Age)
BPI Gene Expression
400
200
0
25
50
75
100
Age

```
##
## [[3]]
```


BPI Gene Expression vs Mechanical Ventilation
BPI Gene Expression
400
200
0
no
yes
Mechanical Ventilation
Sex
female
male

```
##
## [[4]]
```


BPI Gene Expression vs Sex
BPI Gene Expression
400
200
0
female
male
Sex
Mechanical Ventilation
no
yes