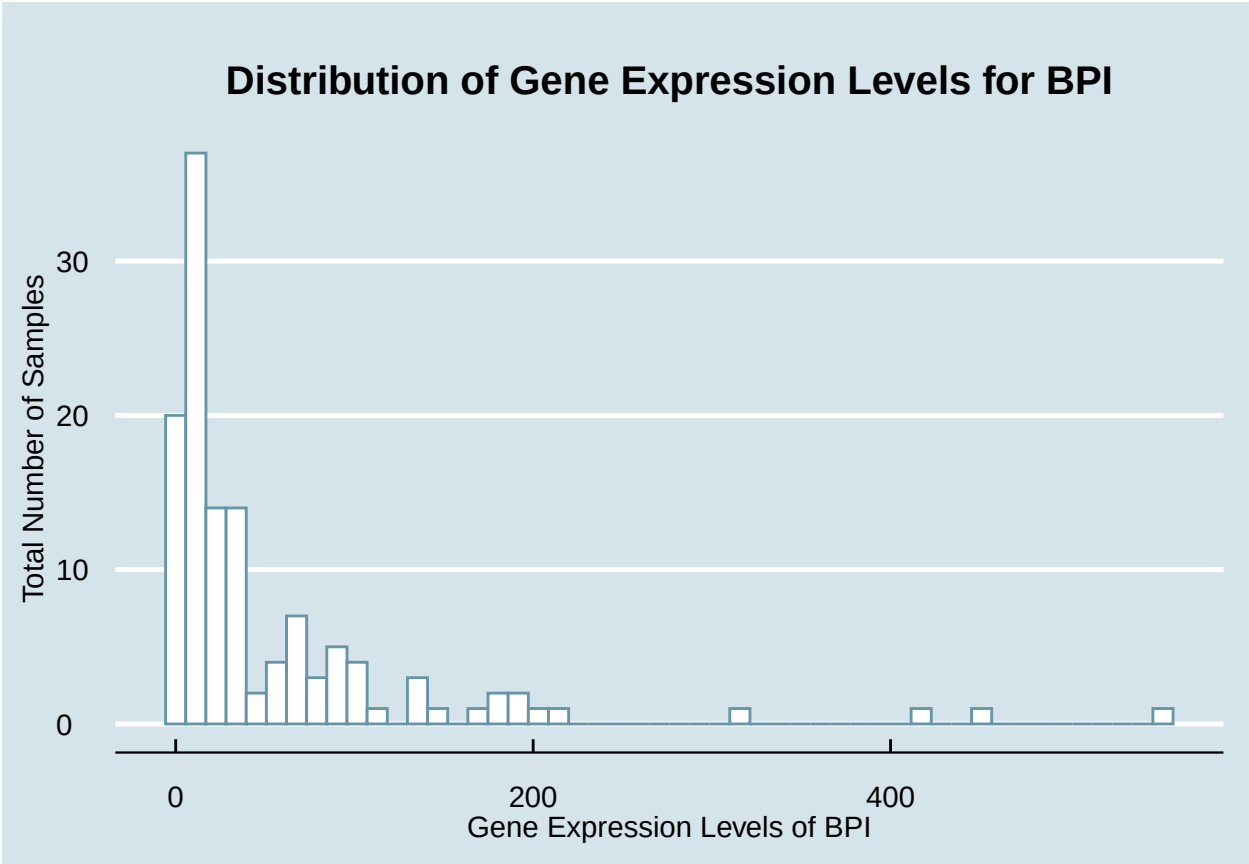
QBS103 Final Project

2023-07-23

```
metadata <- read.csv("QBS103_finalProject_metadata.csv", row.names = 1)
gene_exp <- read.csv("QBS103_finalProject_geneExpression.csv", row.names = 1)
```

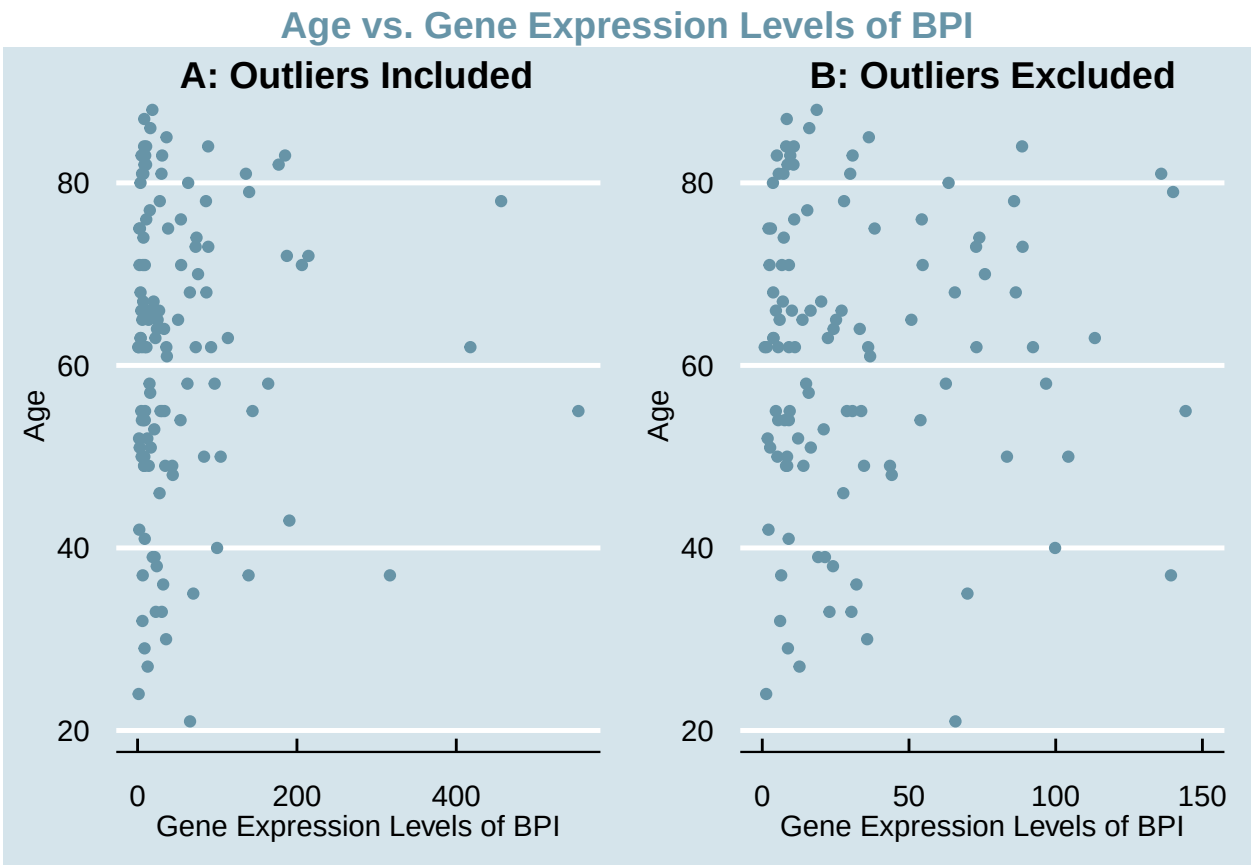
```
#chosen gene: BPI - bactericidal permeability increasing protein - Plays a role in the immune response

BPI <- gene_exp["BPI",]
BPI <- as.data.frame(t(BPI)) #transposes the matrix
metadata_BPI <- cbind(metadata, BPI) #creates one data frame with all BPI data
suppressWarnings(metadata_BPI$age <- as.integer(metadata_BPI$age))
```

```
#creates histogram of BPI

ggplot(metadata_BPI, aes(x = BPI)) + geom_histogram(color = "#6794a7", fill = "white", bins = 50) +
  labs(x = "Gene Expression Levels of BPI", y = "Total Number of Samples", title = "Distribution of Gene
  theme_economist() +
  scale_fill_economist() +
  theme(plot.title = element_text(size=15, face="bold", margin = margin(10, 0, 10, 0), hjust = (0.5)), a
```



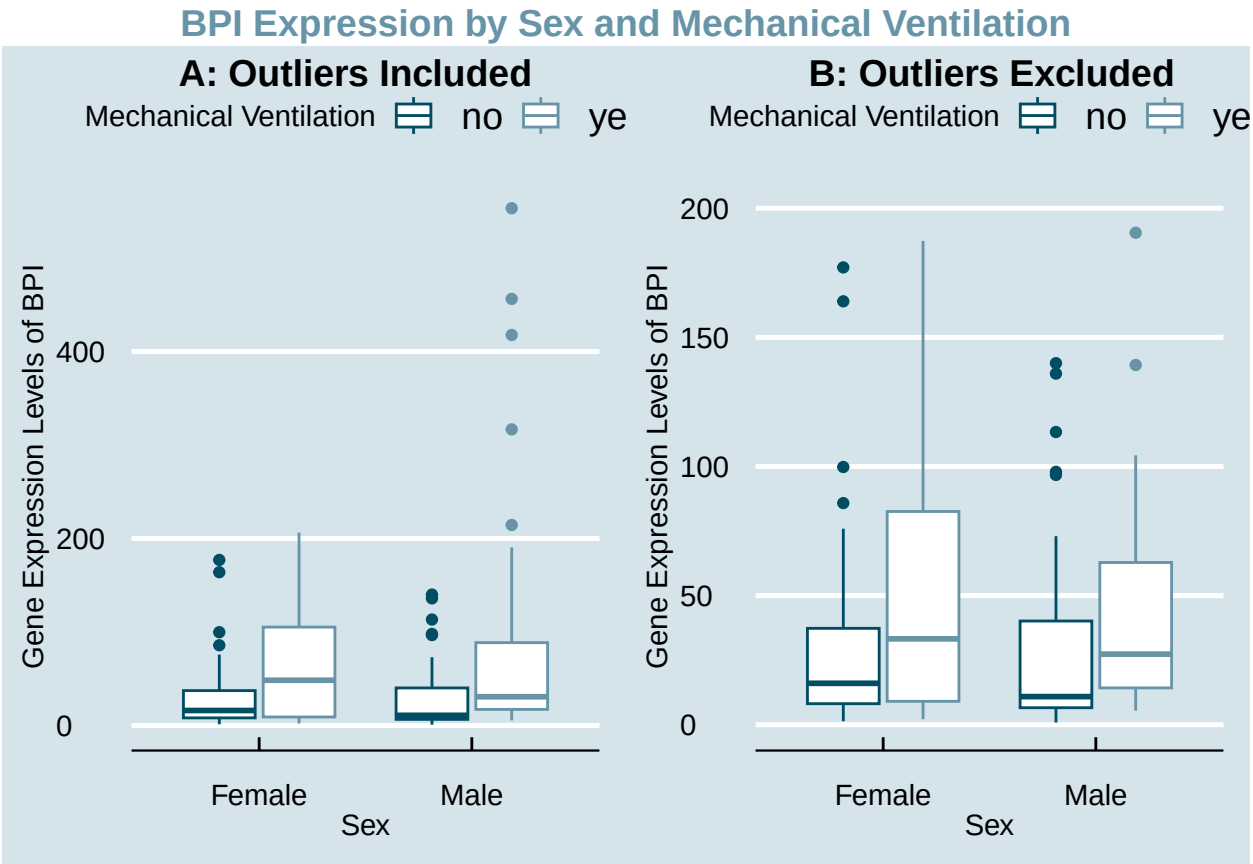

```
# interpretation - shows the distribution of BPI gene expression in the samples. there are more studies
```

```
#chosen continuous covariate: age

#creates scatterplot with BPI and age
suppressWarnings(scatter_outliers <- ggplot(metadata_BPI, aes(x = BPI, y = age)) +
  geom_point(color = "#6794a7", fill = "white") +
  labs(x = "Gene Expression Levels of BPI", y = "Age") +
  theme_economist() +
  scale_fill_economist() +
  theme(plot.title = element_text(size=12, face="bold", margin = margin(10, 0, 10, 0)), axis.title.x = 

#creates scatterplot with BPI and age and sets x-axis range from 0 to 150 for easier viewing
suppressWarnings(scatter_no_outliers <- ggplot(metadata_BPI, aes(x = BPI, y = age)) +
  geom_point(color = "#6794a7", fill = "white") +
  labs(x = "Gene Expression Levels of BPI", y = "Age") +
  theme_economist() +
  scale_fill_economist() +
  theme(plot.title = element_text(size=12, face="bold", margin = margin(10, 0, 10, 0)), axis.title.x = 
  xlim(0, 150))

#creates scatterplot of both previous plots combined
suppressWarnings(new_scatter <- ggarrange(scatter_outliers, scatter_no_outliers,ncol=2, labels = c("A: 

#adds title to the combined plot
suppressWarnings(annotate_figure(new_scatter, top = text_grob("Age vs. Gene Expression Levels of BPI", 
```




```
#interpretation - shows the relationship between BPI gene expression and ages of individuals sampled. t
```

```
#chosen categorical covariates: sex and mechanical ventilation

#create new dataset with the row of data where sex = "unknown"
unknown_removed <- metadata_BPI
unknown_removed <- metadata_BPI[metadata_BPI$sex != " unknown", ]

#creates scatterplot with BPI, sex, and mechanical ventilation
suppressWarnings(box_outliers <- ggplot(unknown_removed,aes(x = sex,y = BPI, color = mechanical_ventila

#creates scatterplot with BPI, sex, and mechanical ventilation and sets y-axis range from 0 to 200 for
suppressWarnings(box_no_outliers <- ggplot(unknown_removed,aes(x = sex,y = BPI, color = mechanical_vent

#creates boxplot of both previous plots combined
suppressWarnings(new_box <- ggarrange(box_outliers, box_no_outliers,ncol=2, labels = c("A: Outliers Incl

#adds title to the combined plot
annotate_figure(new_box, top = text_grob("BPI Expression by Sex and Mechanical Ventilation", color = "#
```




```
#interpretation - shows distribution of BPI gene expression across sexes and ventilation categories. fr
```

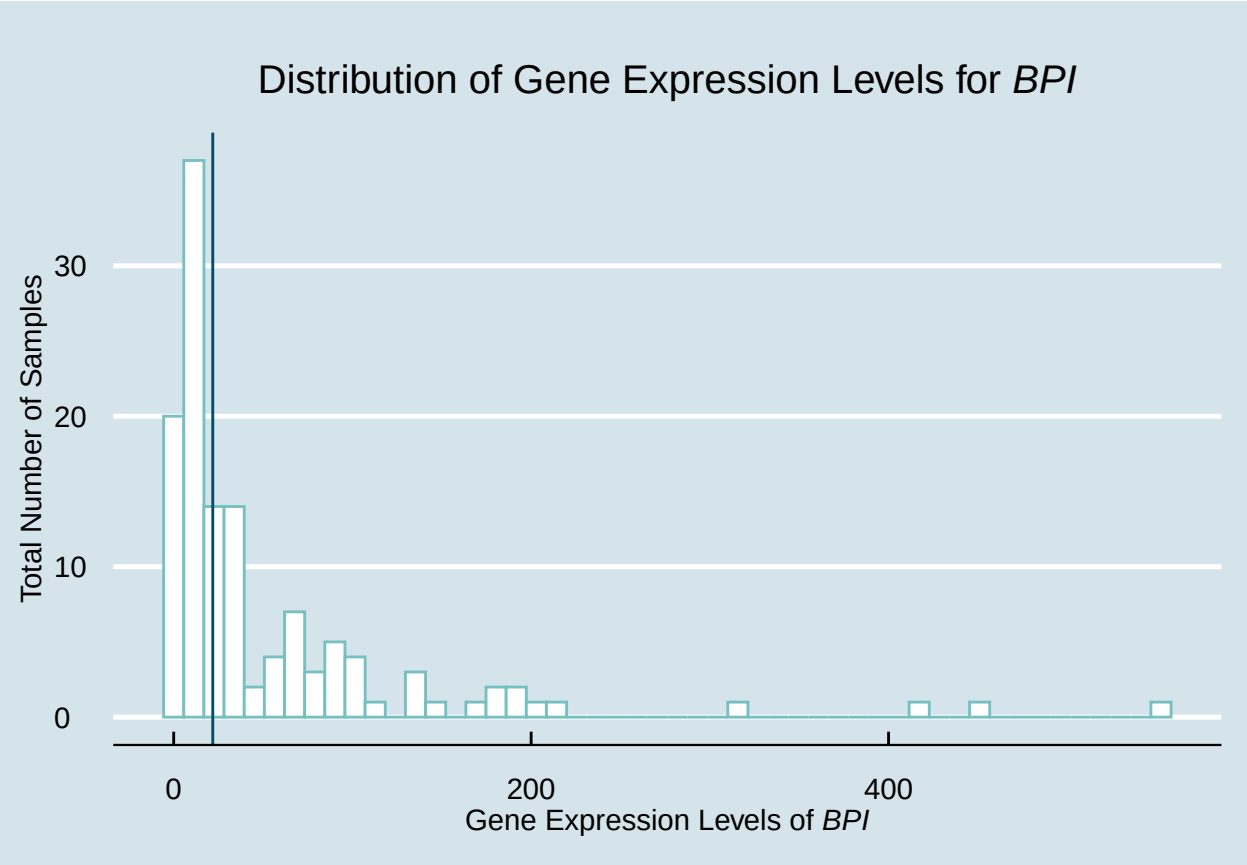
```
#Code for boxplot, scatter, and histogram
all_graphs <- function(data, genes, gene_exp, contcovariate, catcovariate1, catcovariate2, metadata) {
  graphs <- list()
  for (gene in genes) { #creates a loop to create each graph per gene
    gene_data <- as.numeric(t(gene_exp[gene, ]))
    metadata_gene <- cbind(metadata, gene_data)

    #relabels data within a few covariates for better keys in boxplots
    metadata_gene$icu_status <- ifelse(metadata_gene$icu_status == " yes", "in icu", "not in icu")
    metadata_gene$mechanical_ventilation <- ifelse(metadata_gene$mechanical_ventilation == " yes", "wit
    metadata_gene$disease_status <- ifelse(metadata_gene$disease_status == "disease state: COVID-19", "


    #creates histogram
    hist <- ggplot(metadata_gene, aes(x = gene_data)) +
      geom_histogram(color = "#76c0c1", fill = "white", bins = 50) +
      geom_vline(xintercept = median(gene_data), #adds mean line
             col = "#014d64",
             lwd = .5) +
      labs(x = substitute(paste("Gene Expression Levels of ", italic(gene)), list(gene = gene)), #subst
           y = "Total Number of Samples",
           title = substitute(paste("Distribution of Gene Expression Levels for ", italic(gene)), list(
      theme_economist() +
      scale_fill_economist() +
```

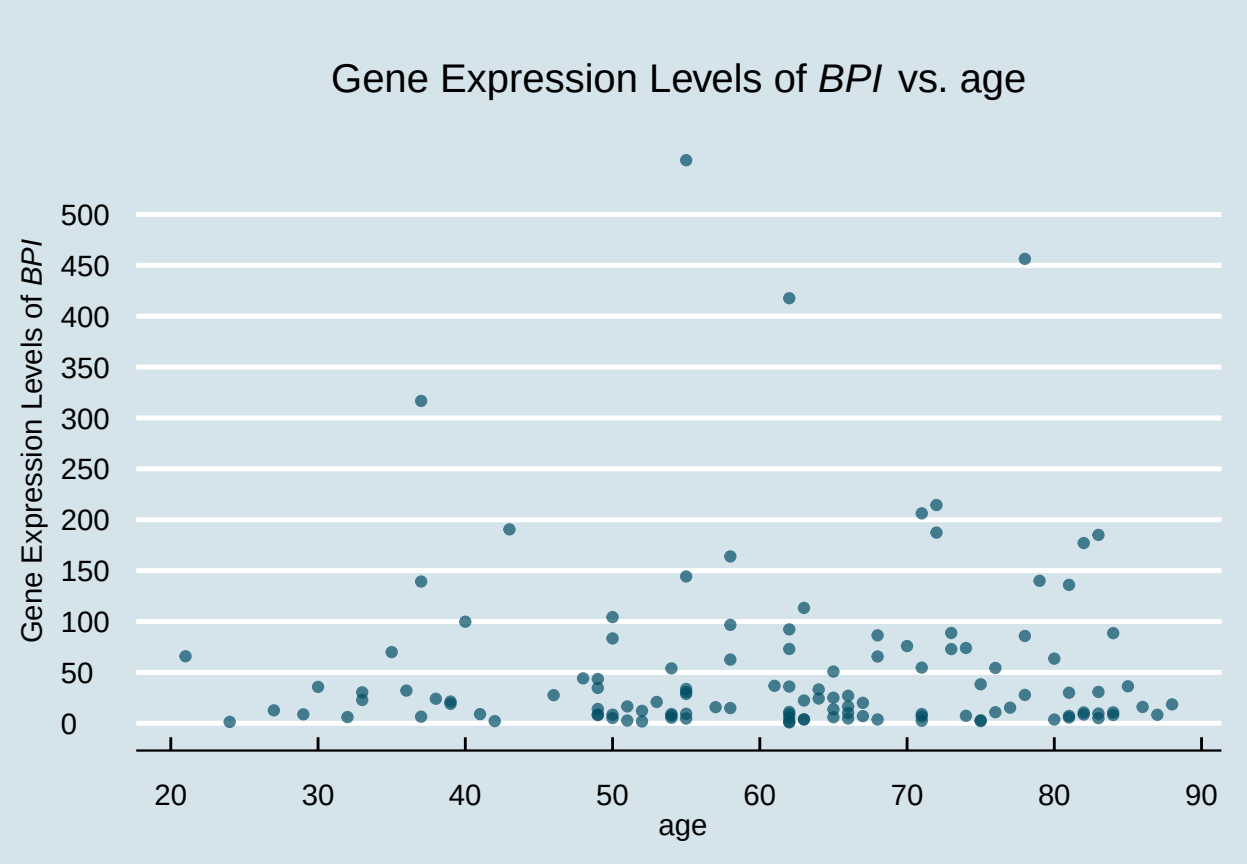
```r
      theme(plot.title = element_text(size = 15, face = "bold", margin = margin(10, 0, 10, 0), hjust = (
            axis.title.x = element_text(vjust = -0.4),
            axis.title.y = element_text(vjust = 2))


    #creates scatter plot
    metadata_gene$contcovariate <- as.integer(metadata_gene[[contcovariate]])
    metadata_gene <- metadata_gene[!is.na(metadata_gene$contcovariate), ]

    scatter <- ggplot(metadata_gene, aes(x = contcovariate, y = gene_data)) +
      geom_point(color = "#014d64", fill = "white", alpha = 0.7) +
      labs(x = paste("", contcovariate) , y = substitute(paste("Gene Expression Levels of ", italic(gene
      theme_economist() +
      scale_fill_economist() +
      theme(plot.title = element_text(size = 15, face = "bold", margin = margin(10, 0, 10, 0), hjust = (
      ggtitle(substitute(paste("Gene Expression Levels of ", italic(gene), " vs. ", contcovariate), lis


    #creates boxplot
    metadata_gene$catcovariate1 <- as.factor(metadata_gene[[catcovariate1]])
    metadata_gene$catcovariate2 <- as.factor(metadata_gene[[catcovariate2]])
    metadata_gene <- metadata_gene[!is.na(metadata_gene$catcovariate1) & !is.na(metadata_gene$catcovaria

    box <- ggplot(metadata_gene, aes(x = catcovariate1, y = gene_data, color = catcovariate2)) +
      geom_boxplot() +
      scale_x_discrete(labels = levels(metadata_gene$catcovariate1)) +
      labs(x = gsub("_", " ", paste("", catcovariate1)), y = substitute(paste("Gene Expression Levels o
      theme_economist() +
      scale_color_manual(values = c('#014d64','#76c0c1')) +
      theme(plot.title = element_text(size = 15, face ="bold", margin = margin(10, 0, 10, 0), hjust = (
      ggtitle(substitute(paste(italic(gene), " Expression by ", catcovariate1, " and ", "_", " ", catco

    plots <- list(hist = hist, scatter = scatter, box = box)
    graphs[[gene]] <- plots

  }
  return(graphs)
}

selected_genes <- c("BPI", "CD24")

multiple_graphs <- all_graphs(data = gene_exp, genes = selected_genes, contcovariate = "age", catcovaria
```

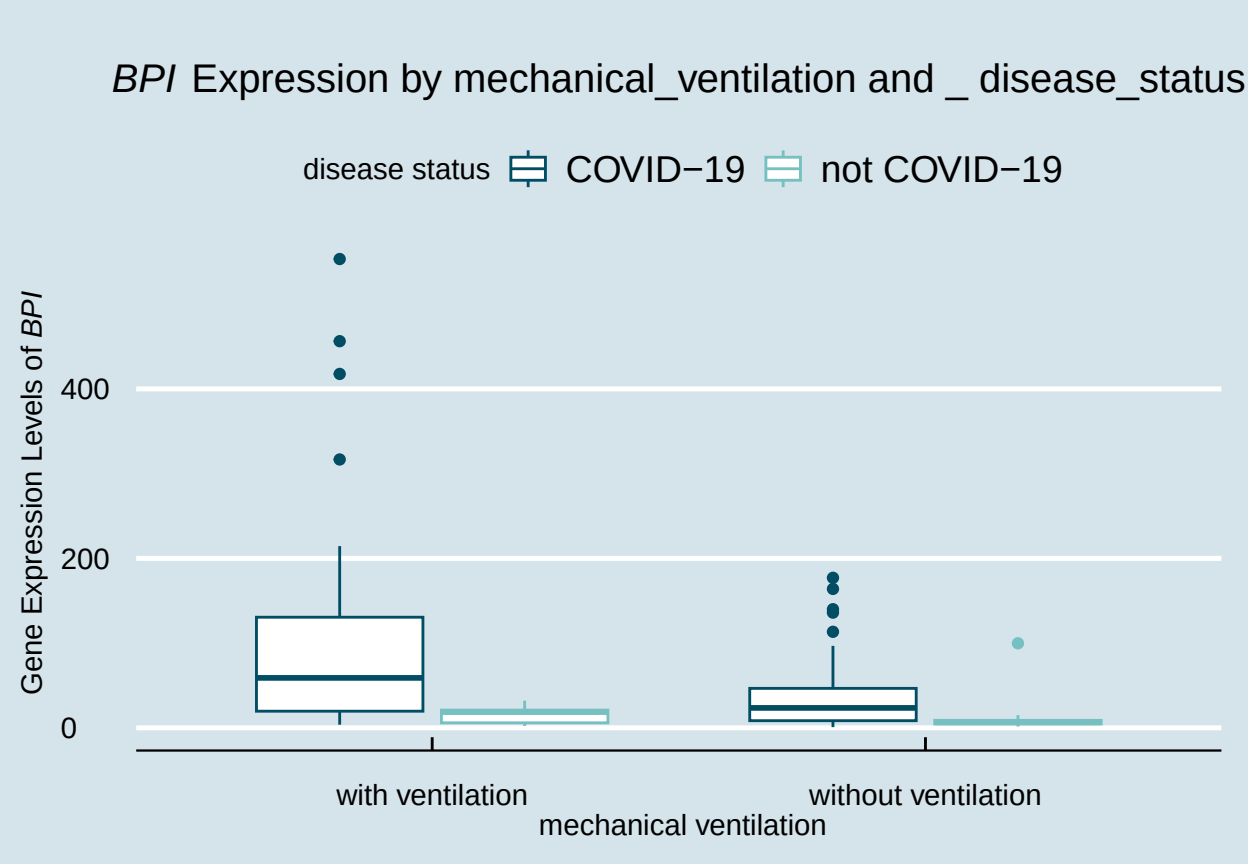
```
## Warning in all_graphs(data = gene_exp, genes = selected_genes, contcovariate =
## "age", : NAs introduced by coercion

## Warning in all_graphs(data = gene_exp, genes = selected_genes, contcovariate =
## "age", : NAs introduced by coercion
```

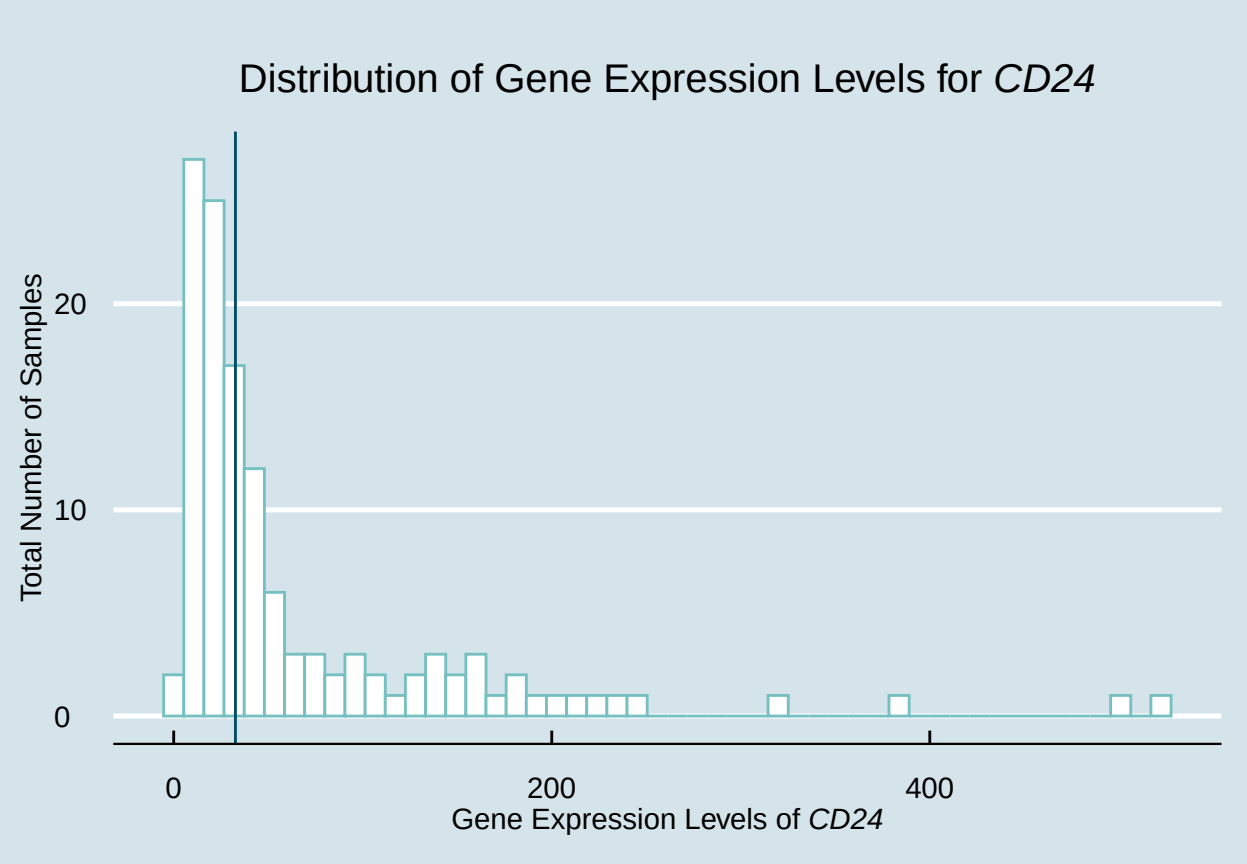
```r
for (gene in selected_genes) {
  print(multiple_graphs[[gene]]$hist)
  print(multiple_graphs[[gene]]$scatter)
  print(multiple_graphs[[gene]]$box)
```

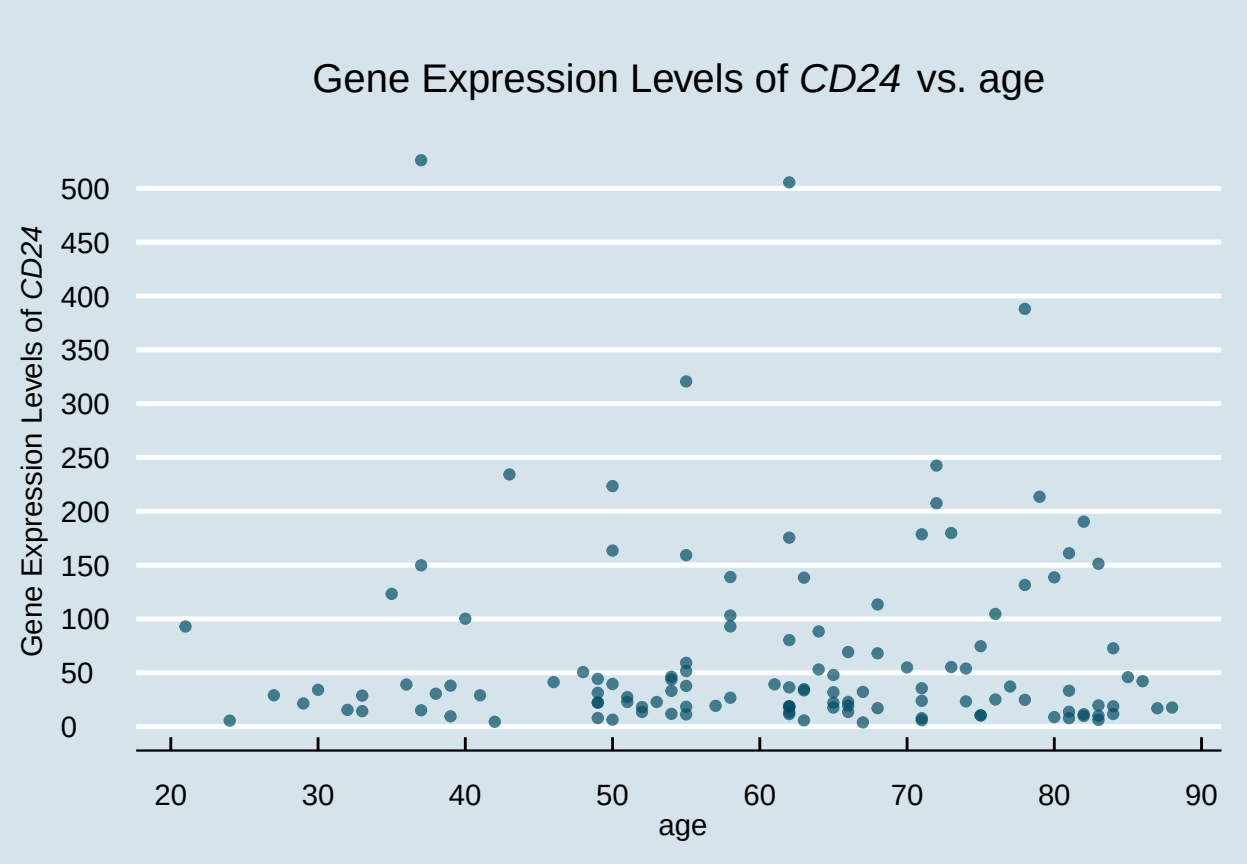
```
}
```

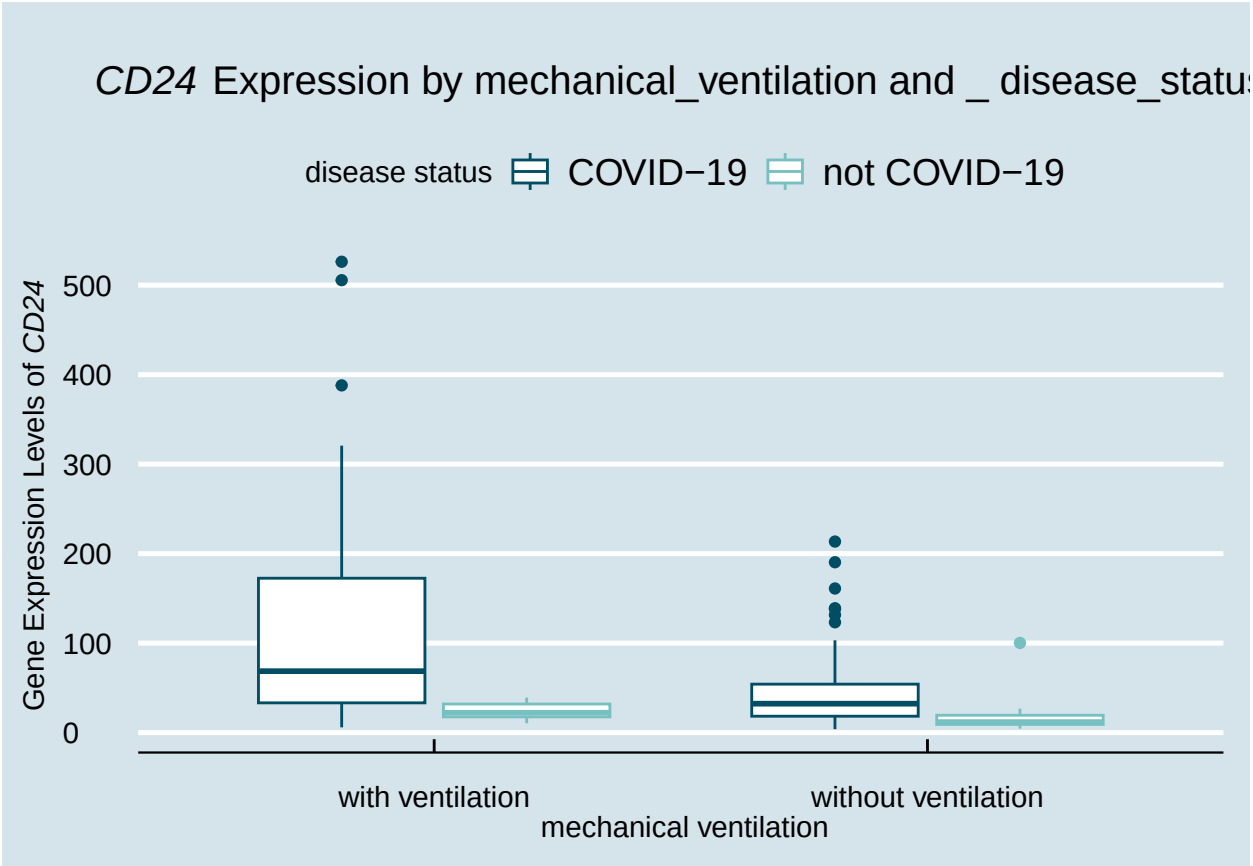

Distribution of Gene Expression Levels for *BPI*
Total Number of Samples
30
20
10
0
0
200
400
Gene Expression Levels of *BPI*


Gene Expression Levels of *BPI* vs. age
Gene Expression Levels of *BPI*
500
450
400
350
300
250
200
150
100
50
0
20
30
40
50
60
70
80
90
age


BPI Expression by mechanical_ventilation and _ disease_status
disease status
COVID−19
not COVID−19
Gene Expression Levels of *BPI*
400
200
0
with ventilation
without ventilation
mechanical ventilation


Distribution of Gene Expression Levels for *CD24*
Total Number of Samples
20
10
0
0
200
400
Gene Expression Levels of *CD24*


Gene Expression Levels of *CD24* vs. age
Gene Expression Levels of *CD24*
500
450
400
350
300
250
200
150
100
50
0
20
30
40
50
60
70
80
90
age

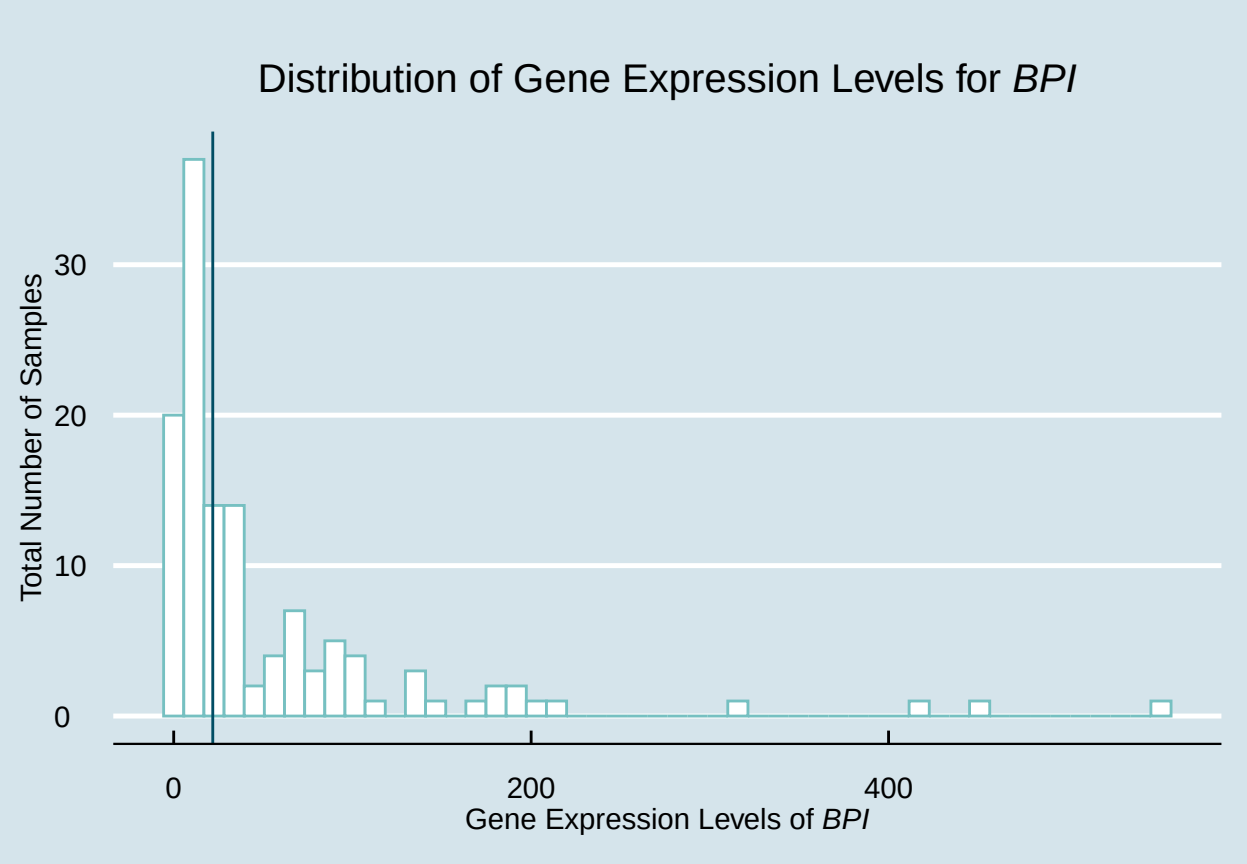



```
#calling the function
additional_genes <- c("GPLD1", "AAK1")
selected_genes <- c("BPI", "MPO", "CD24")
multiple_graphs <- all_graphs(data = gene_exp, genes = selected_genes, contcovariate = "age", catcovari
```

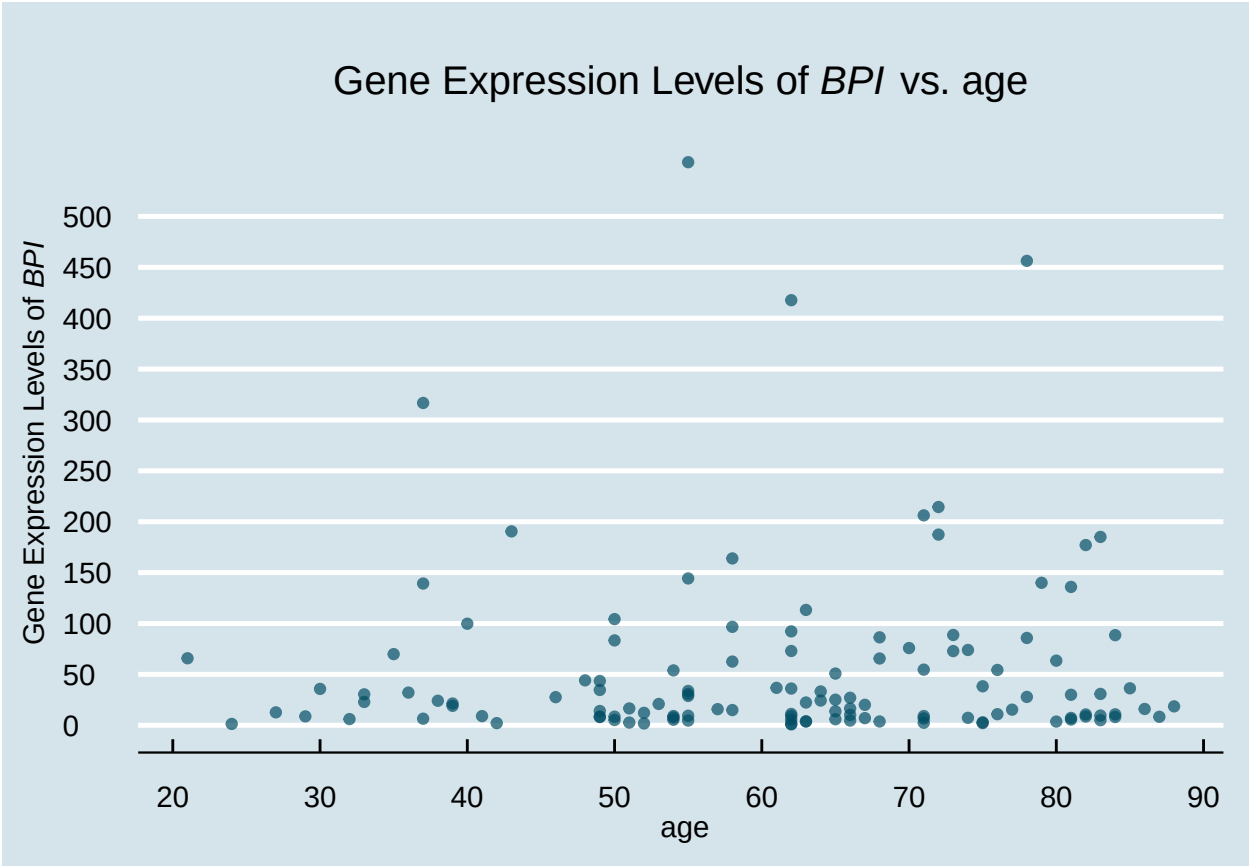
```
## Warning in all_graphs(data = gene_exp, genes = selected_genes, contcovariate =
## "age", : NAs introduced by coercion

## Warning in all_graphs(data = gene_exp, genes = selected_genes, contcovariate =
## "age", : NAs introduced by coercion

## Warning in all_graphs(data = gene_exp, genes = selected_genes, contcovariate =
## "age", : NAs introduced by coercion
```

```
print(multiple_graphs)
```

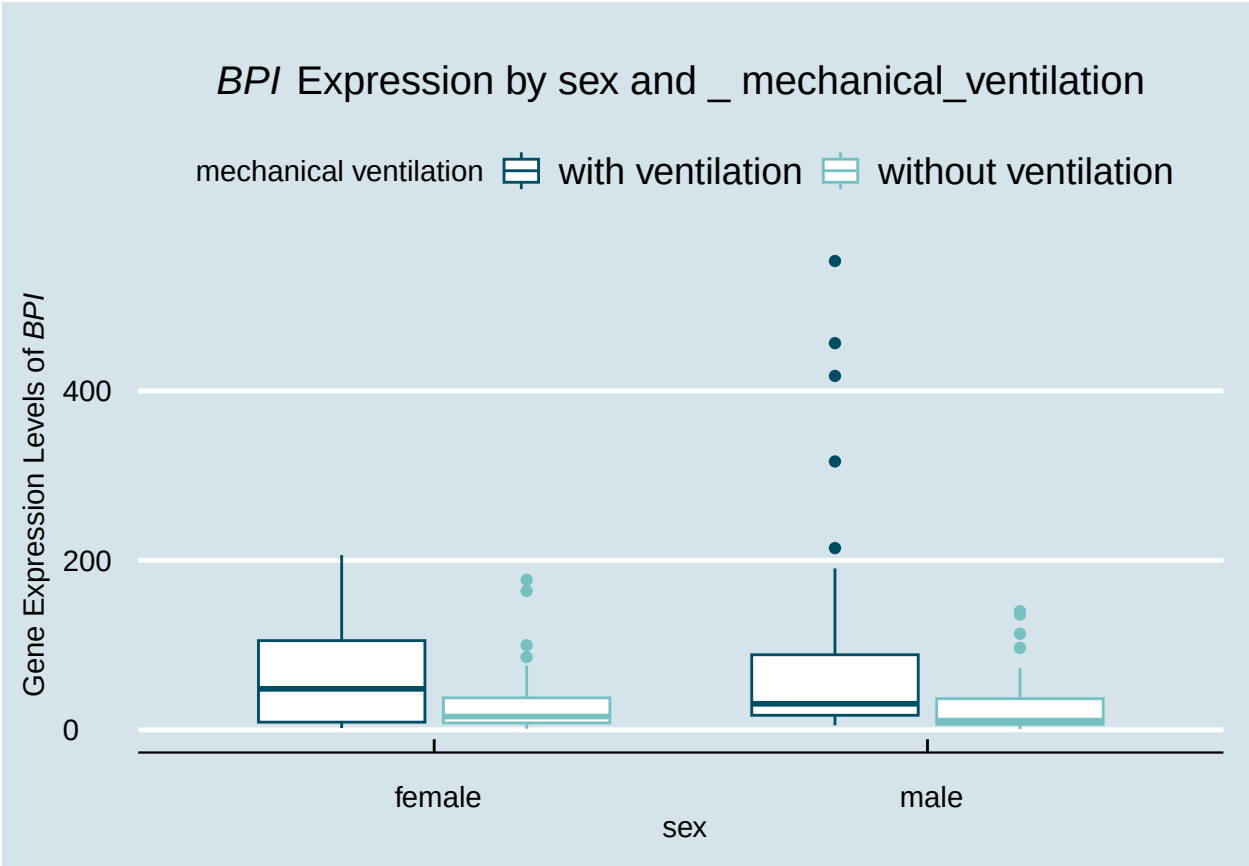
```
## $BPI
## $BPI$hist
```


Distribution of Gene Expression Levels for *BPI*
Total Number of Samples
30
20
10
0
0
200
400
Gene Expression Levels of *BPI*

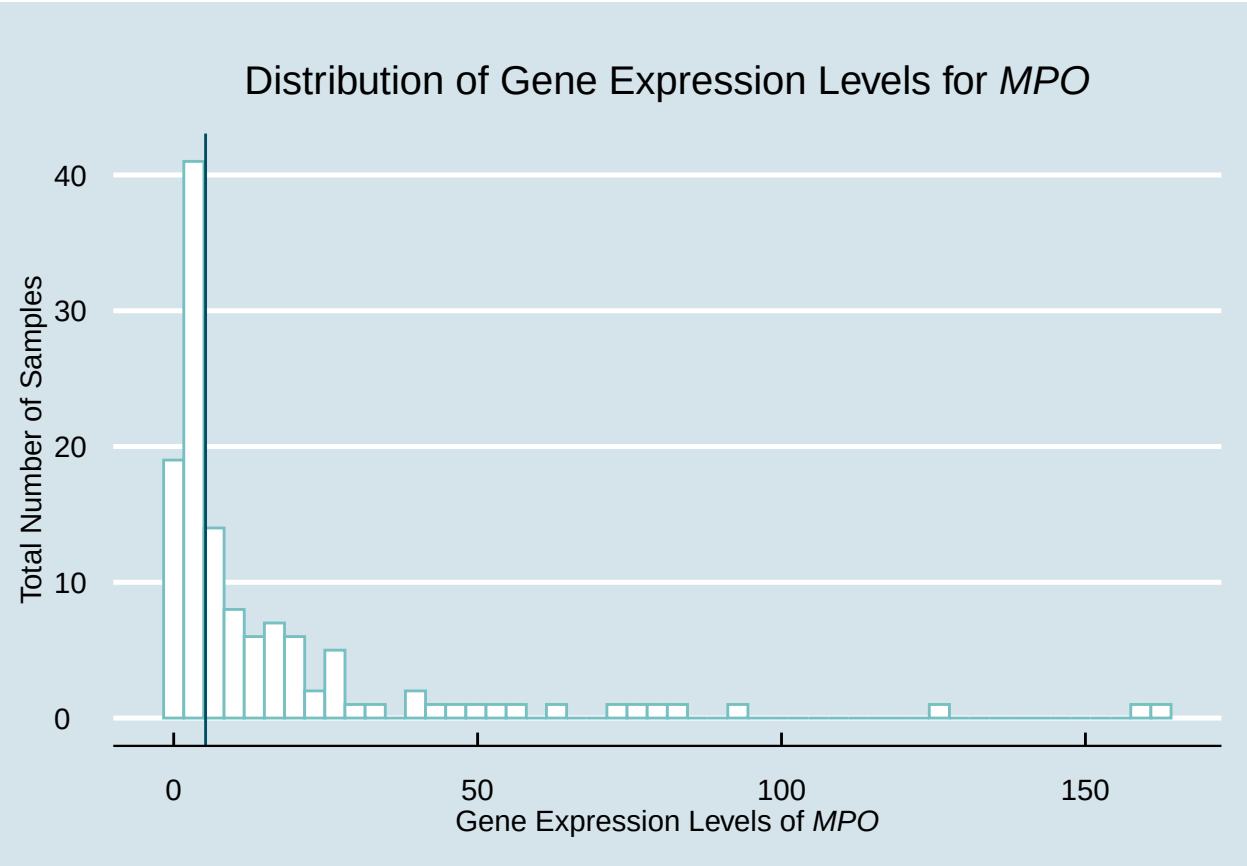
```
##
## $BPI$scatter
```


Gene Expression Levels of *BPI* vs. age
Gene Expression Levels of *BPI*
500
450
400
350
300
250
200
150
100
50
0
20
30
40
50
60
70
80
90
age

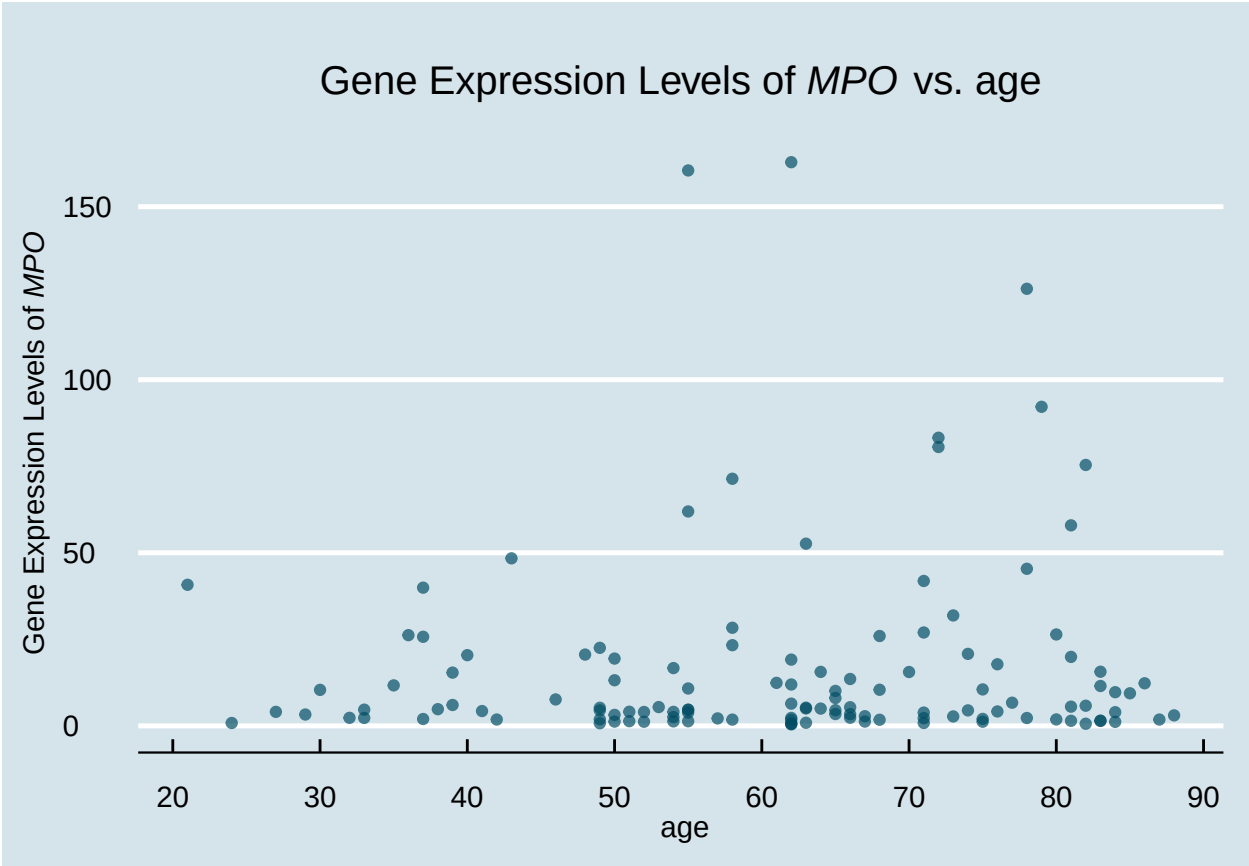
```
##
## $BPI$box
```


BPI Expression by sex and _ mechanical_ventilation
mechanical ventilation
with ventilation
without ventilation
Gene Expression Levels of BPI
400
200
0
female
male
sex

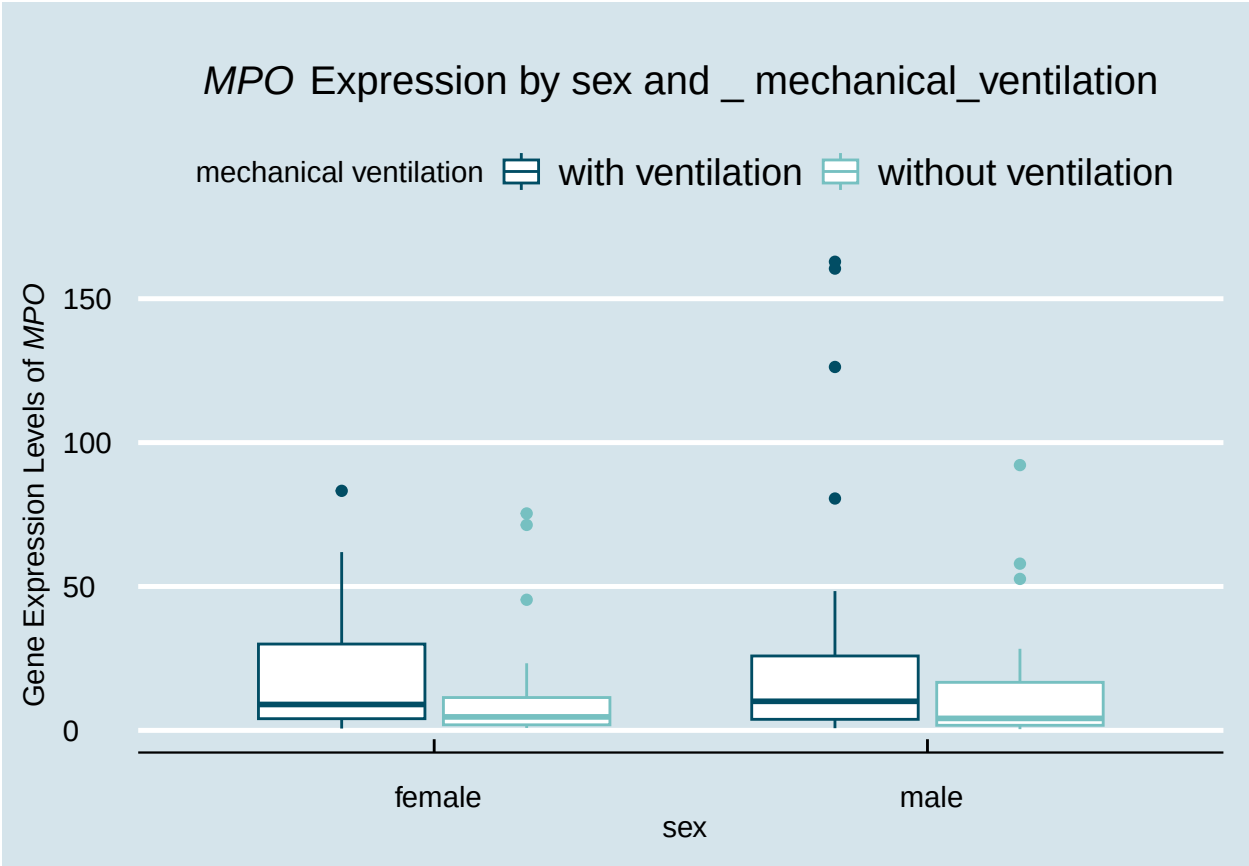
```
##
##
## $MPO
## $MPO$hist
```


Distribution of Gene Expression Levels for *MPO*
Total Number of Samples
40
30
20
10
0
0
50
100
150
Gene Expression Levels of *MPO*

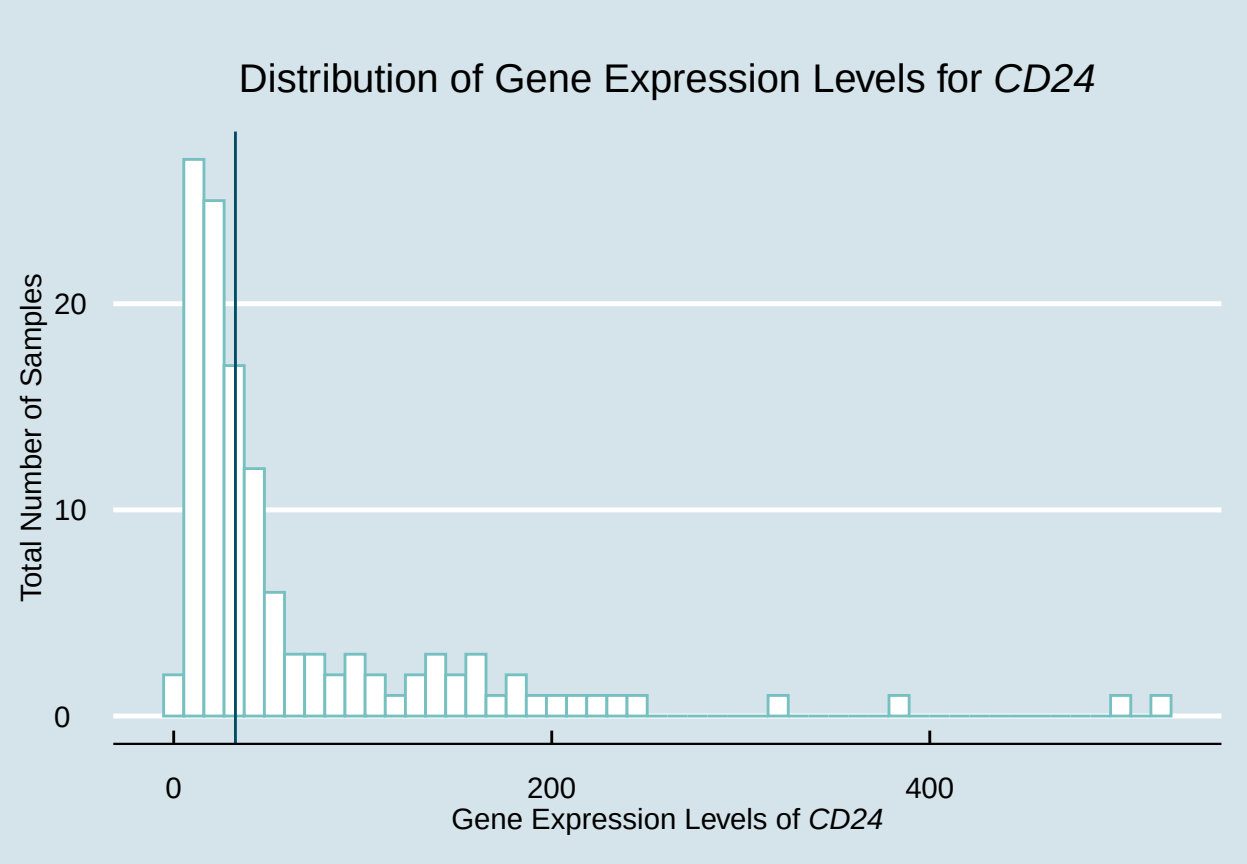
```
##
## $MPO$scatter
```


Gene Expression Levels of *MPO* vs. age
Gene Expression Levels of *MPO*
150
100
50
0
20
30
40
50
60
70
80
90
age

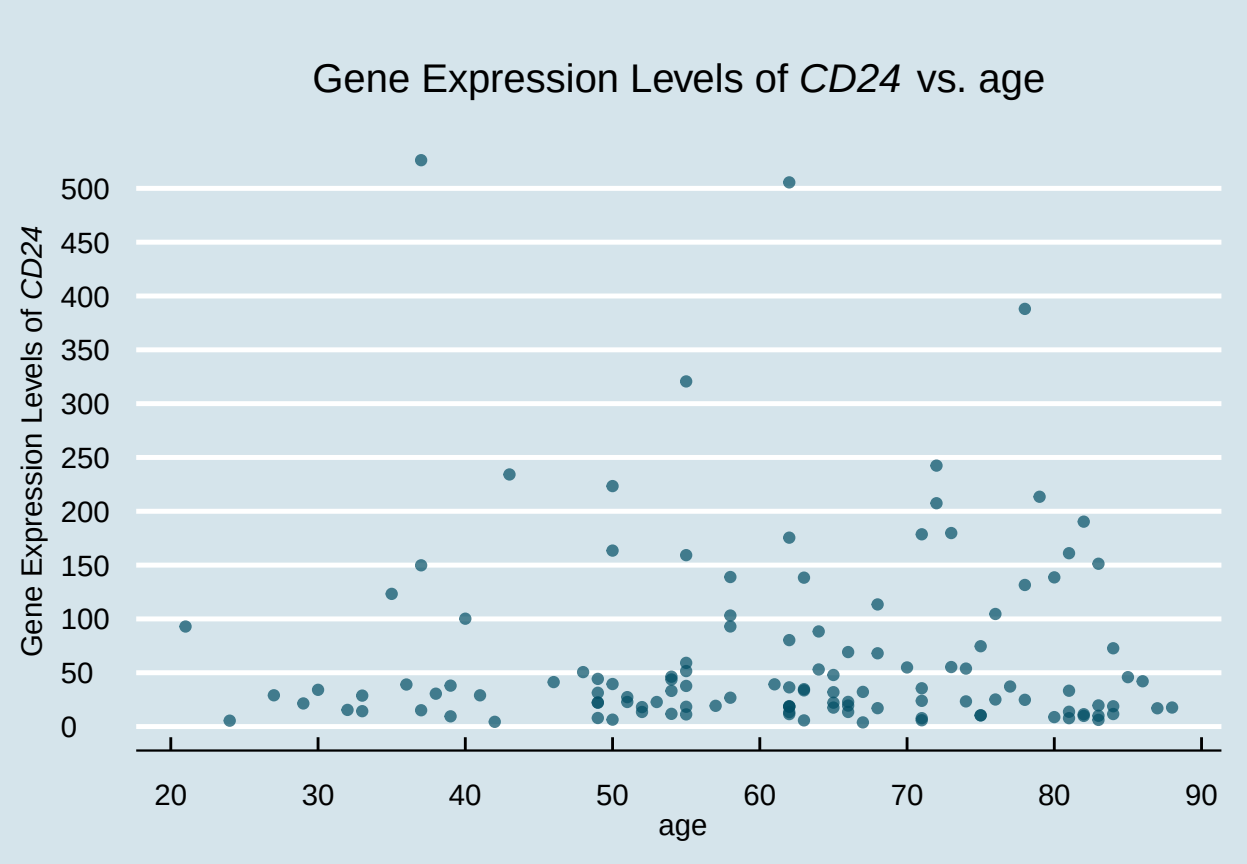
```
##
## $MPO$box
```



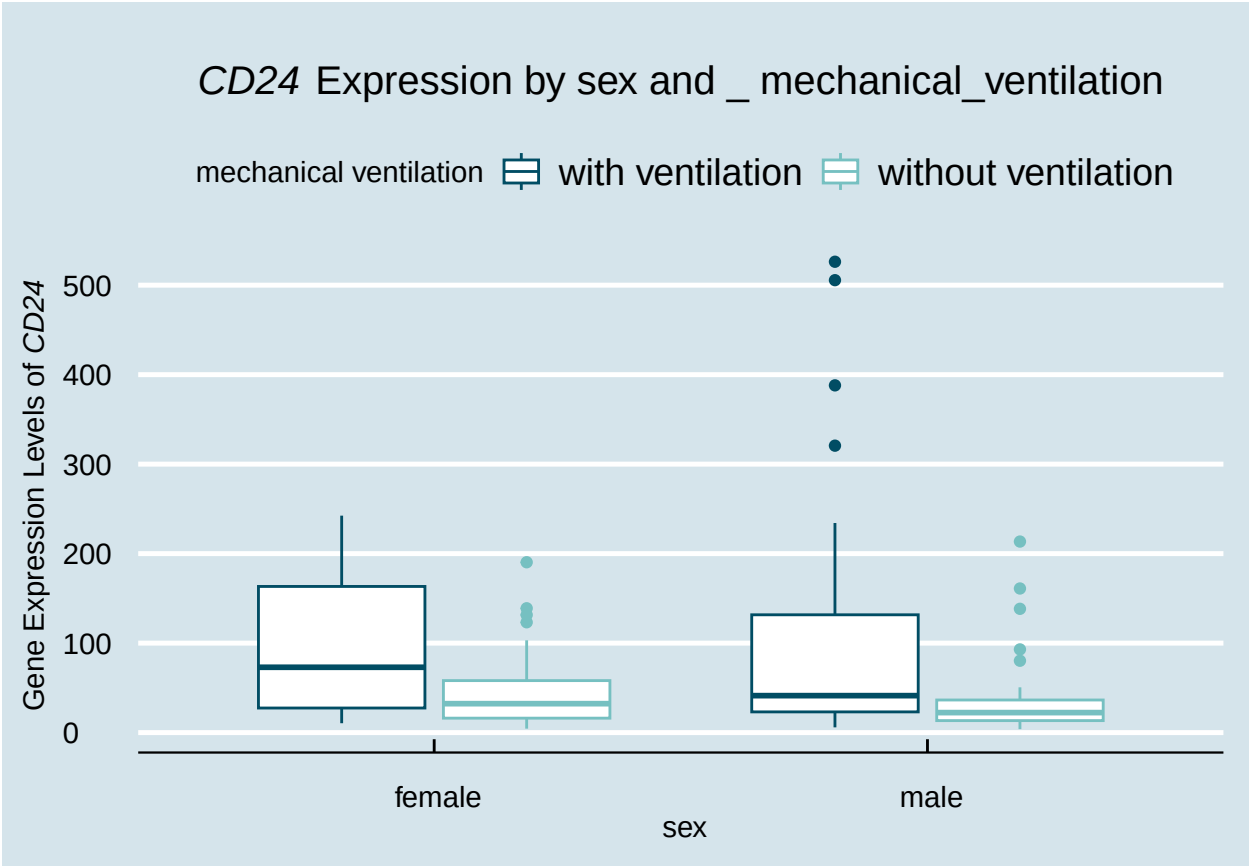

```
##
##
## $CD24
## $CD24$hist
```




```
##
## $CD24$scatter
```


Gene Expression Levels of *CD24* vs. age
Gene Expression Levels of *CD24*
500
450
400
350
300
250
200
150
100
50
0
20
30
40
50
60
70
80
90
age

```
##
## $CD24$box
```




```
#code adapted from class
library(knitr)
buildTableOne <- function(data,varList,nonnormVars = '',catVars = '') {

  # Define an empty table
  table1 <- matrix(nrow = 0,ncol = 2)
  # Keep track of rows to indent
  indentRows <- c()

  # Loop through all variables
  for (var in varList) {

    # Define vector of variable values
    x <- data[,var]

    # Identify if non-normal
    if (var %in% nonnormVars) {

      # Calculate individual values
      myMedian <- round(median(x))
      myIQR1 <- round(quantile(x,1/4),digits = 2)
      myIQR2 <- round(quantile(x,3/4),digits = 2)
      # Combine values
      value <- paste0(myMedian,' [',myIQR1,', ',myIQR2,']')

      # Define new row
```

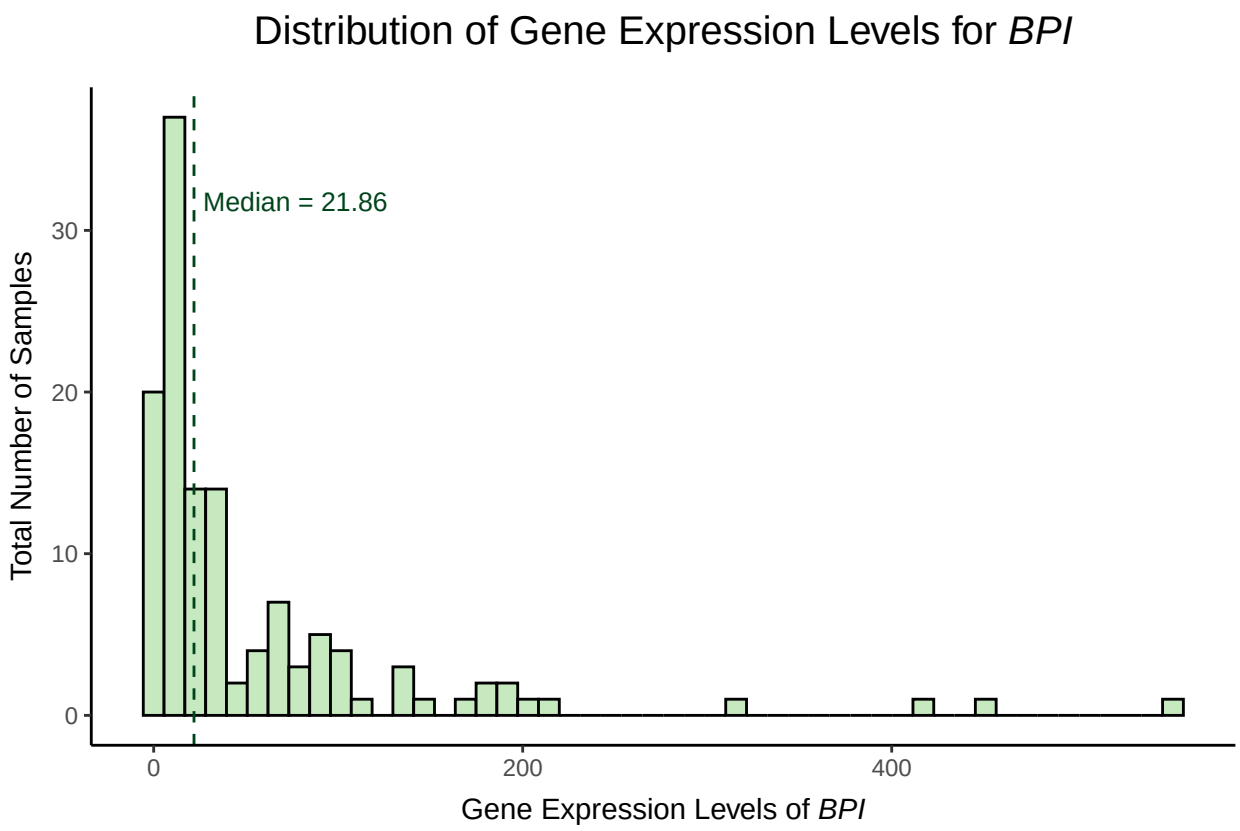
```r
    newRow <- c(paste0('**',var,'** Median [IQR]'),value)

    # Add row to data frame
    table1 <- rbind(table1,newRow)

  }

  # Identify if categorical
  if (var %in% catVars) {

    # Define new row for overall variable
    newRow <- c(paste0('**',var,'** n (%)'),'')
    # Add row to data frame
    table1 <- rbind(table1,newRow)

    # Loop through levels of variable
    for (level in levels(x)) {
      # Calculate n and perc
      n <- sum(x == level)
      perc <- round(n/(length(x)) * 100,digits = 2)

      # Combine values
      value <- paste0(n,' (',perc,')')

      # Define new row
      newRow <- c(level,value)
      # Add row to data frame
      table1 <- rbind(table1,newRow)
      # Add index to indented rows
      indentRows <- c(indentRows,nrow(table1))


    }
  }

  # Otherwise treat as normally distributed
  if (!(var %in% c(nonnormVars,catVars))) {

    # Calculate individual values
    myMean <- round(mean(x),2)
    mySD <- round(sd(x),2)
    # Combine values
    value <- paste0(myMean,' (',mySD,')')

    # Define new row
    newRow <- c(paste0('**',var,'** Mean (sd)'),value)

    # Add row to data frame
    table1 <- rbind(table1,newRow)

  }
}
write.csv(table1, file = "table.csv")
```

```
}
```

```
#cleans data further
clean_data <- metadata_BPI[(metadata_BPI$sex != " unknown") & (metadata_BPI$procalcitonin.ng.ml.. != "un
clean_data$procalcitonin.ng.ml.. <- as.numeric(clean_data$procalcitonin.ng.ml..)
clean_data$age <- as.numeric(clean_data$age)
clean_data$icu_status <- as.factor(clean_data$icu_status)
clean_data$sex <- as.factor(clean_data$sex)
clean_data$mechanical_ventilation <- as.factor(clean_data$mechanical_ventilation)

#creates a table for BPI and saves as a csv
buildTableOne(data = clean_data, varList = c('age','ventilator.free_days','procalcitonin.ng.ml..','sex'
```

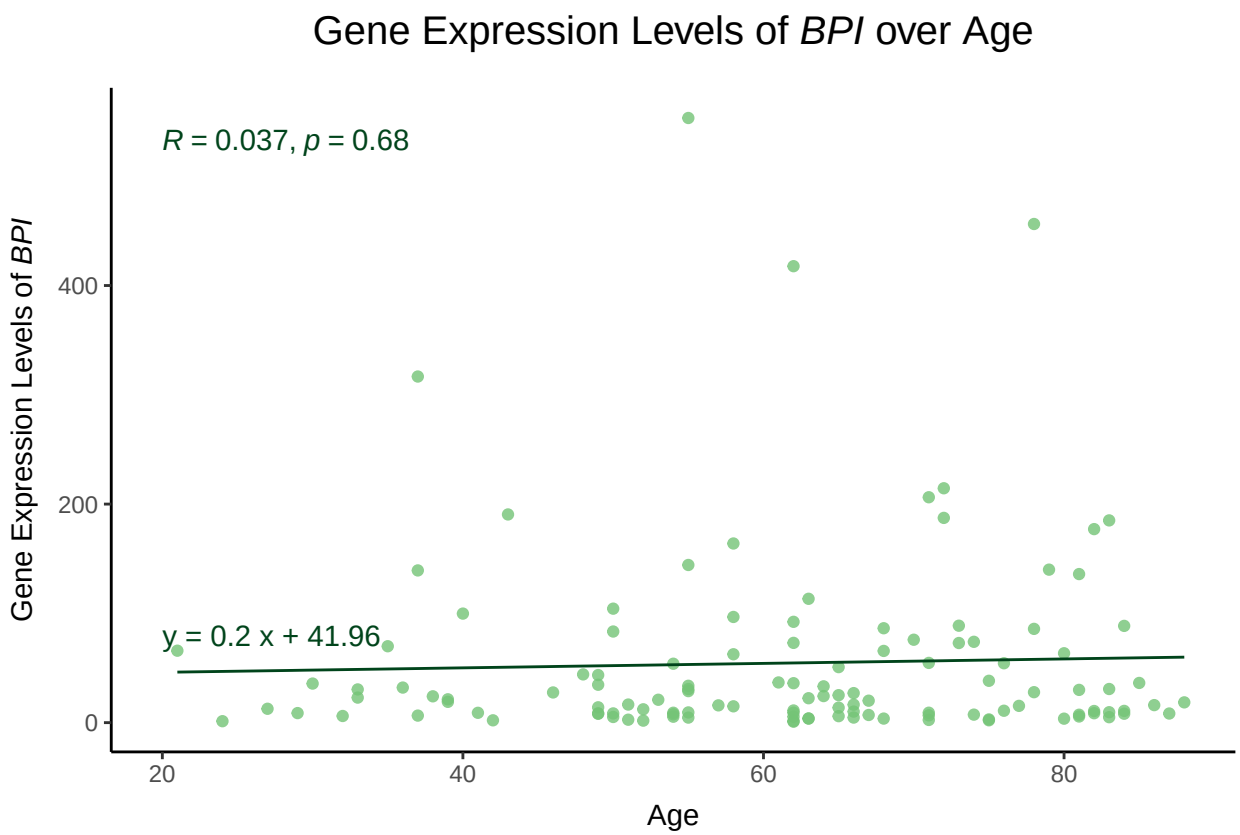
```
#sets new color palettes
greens <- brewer.pal(9, "Greens")
blues <- brewer.pal(9, "Blues")
yellows <- brewer.pal(9, "Oranges")

#cleans data further for boxplot
unknown_removed$MechanicalVentilation <- unknown_removed$mechanical_ventilation
unknown_removed$MechanicalVentilation <- as.factor(unknown_removed$MechanicalVentilation)
unknown_removed$MechanicalVentilation <- gsub(" yes", "Required", unknown_removed$MechanicalVentilation)
unknown_removed$MechanicalVentilation <- gsub(" no", "Not Required", unknown_removed$MechanicalVentilati

#calculates the median for histogram
median_value <- median(metadata_BPI$BPI)

#creates histogram
ggplot(metadata_BPI, aes(x = BPI)) +
  geom_histogram(color = "black", fill = greens[3], bins = 50) +
  geom_vline(aes(xintercept = median_value), color = greens[9], linetype = "dashed", size = .5) +  #add
  annotate("text", x = median_value + 55, y = 30, label = paste("Median =", median_value), color = green
  labs(x = expression(paste("Gene Expression Levels of ", italic("BPI"))), y = "Total Number of Samples"
  theme_classic() +
  scale_fill_manual(values = c(greens[3])) +
  scale_color_manual(values = c(greens[5])) +
  theme(plot.title = element_text(size = 15, face = "bold", margin = margin(10, 0, 10, 0), hjust = (0.5)
```

```
## Warning: Using ‘size‘ aesthetic for lines was deprecated in ggplot2 3.4.0.
## i Please use ‘linewidth‘ instead.
## This warning is displayed once every 8 hours.
## Call ‘lifecycle::last_lifecycle_warnings()‘ to see where this warning was
## generated.
```

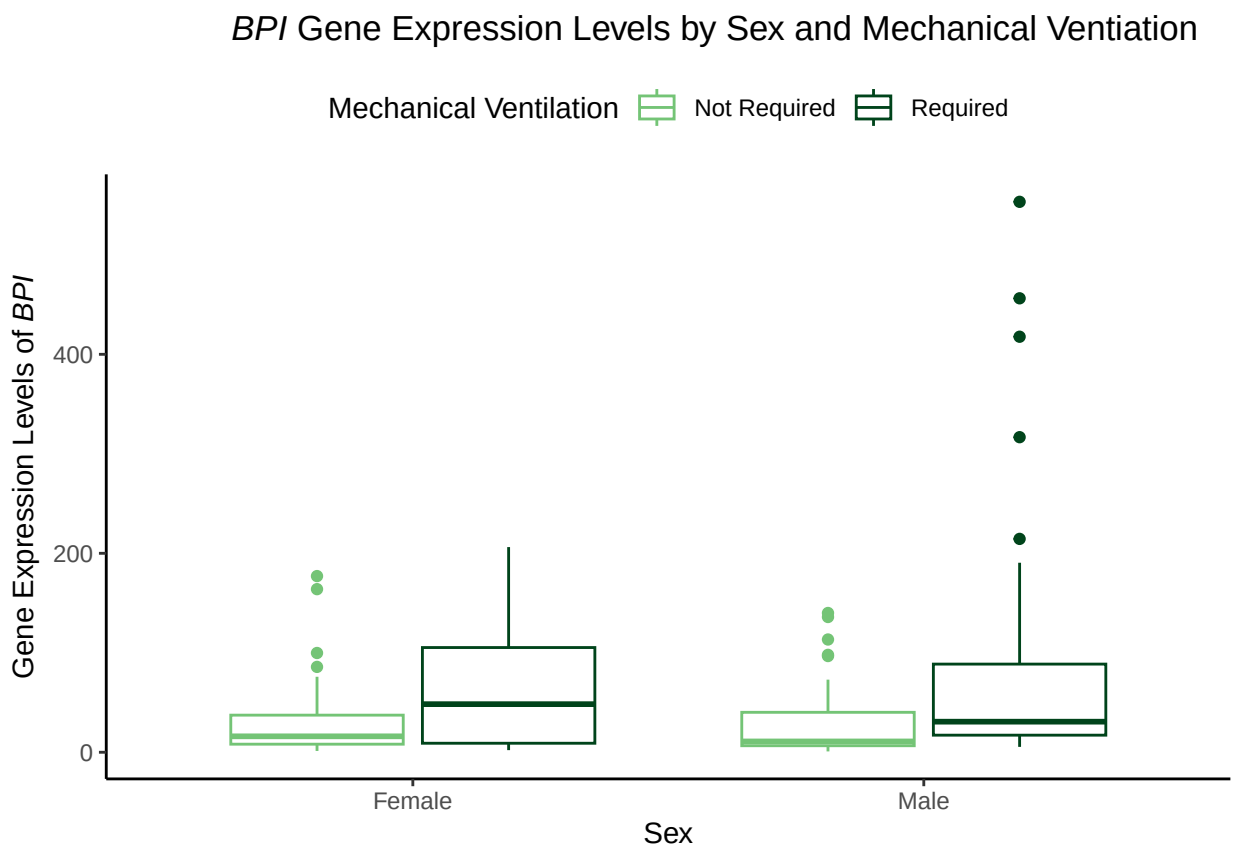



```
#creates scatterplot
ggplot(metadata_BPI, aes(x = age, y = BPI)) +
  geom_point(color = greens[5], fill = "white", alpha = 0.8) +
  geom_smooth(method = "lm", se = FALSE, color = greens[9], size = .5) +
  annotate("text", x = 20, y = 80, label = paste("y =", round(coef(lm(BPI ~ age, data = metadata_BPI))[
  stat_cor(method = "pearson", label.x = 20, color = greens[9]) +
  labs(x = "Age", y = expression(paste("Gene Expression Levels of ", italic("BPI"))), title = expressio
  theme_classic() +
  scale_color_manual(values = c(greens[5])) +
  scale_fill_manual(values = "white") +
  theme(plot.title = element_text(size = 15, face ="bold", margin = margin(10, 0, 10, 0), hjust = 0.5),
```

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

## Warning: Removed 3 rows containing non-finite values (`stat_smooth()`).

## Warning: Removed 3 rows containing non-finite values (`stat_cor()`).

## Warning: Removed 3 rows containing missing values (`geom_point()`).
```

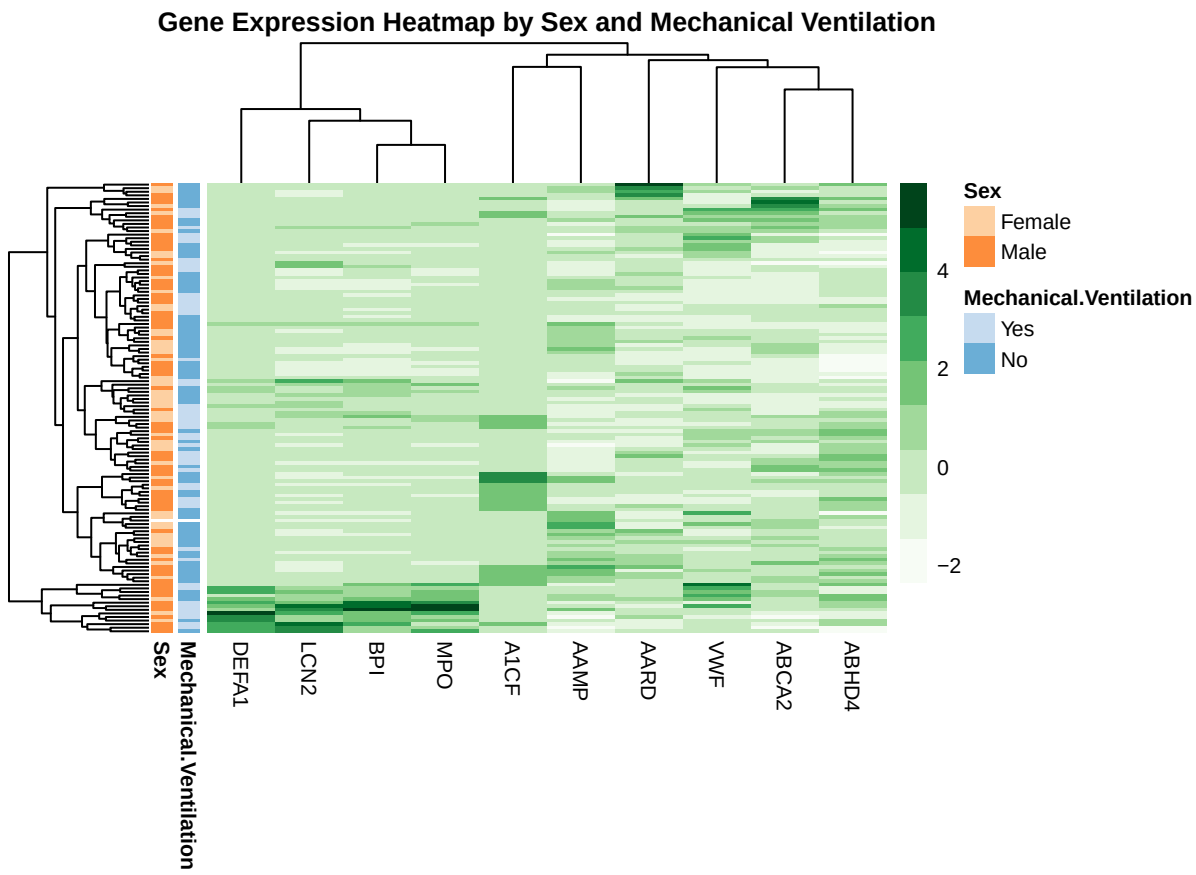



```
#creates boxplot
ggplot(unknown_removed, aes(x = sex, y = BPI, color = MechanicalVentilation)) +
  geom_boxplot() +
  scale_x_discrete(labels = c("Female", "Male")) +
  theme(plot.title = element_text(size = 15, face = "bold", margin = margin(10, 0, 10, 0), hjust = (0.5
  labs(x = "Sex", y = expression(paste("Gene Expression Levels of ", italic("BPI"))), color = "Mechanica
  theme_classic() +
  scale_color_manual(values = c(greens[5], greens[9])) +
  scale_fill_manual(values = "white") +
  theme(legend.position = "top")
```




```
chooseten <- gene_exp[c("BPI", "MPO", "VWF", "A1CF", "AAMP", "ABCA2", "ABHD4", "AARD", "DEFA1", "LCN2")
chooseten <- as.data.frame(t(chooseten))

#creates one data frame with all ten gene data
ten_genes <- cbind(metadata, chooseten)

#recodes to fix labeling in legend
ten_genes$sex <- gsub(" male", "Male", ten_genes$sex)
ten_genes$sex <- gsub(" female", "Female", ten_genes$sex)
ten_genes$mechanical_ventilation <- gsub(" no", "No", ten_genes$mechanical_ventilation)
ten_genes$mechanical_ventilation <- gsub(" yes", "Yes", ten_genes$mechanical_ventilation)

ten_genes_exp <- ten_genes[,-c(1:24)]

#euclidean scaling
scaled <- scale(ten_genes_exp)

#filtering unknown in sex
ten_genes <- ten_genes %>% filter(!grepl('unknown', sex))

#converts variables to factors
ten_genes$sex <- as.factor(ten_genes$sex)
ten_genes$mechanical_ventilation <- as.factor(ten_genes$mechanical_ventilation)

#creates annotations
annotationData <- data.frame(
```

```r
  row.names = rownames(ten_genes),
  'Mechanical.Ventilation' = ten_genes$mechanical_ventilation,
  'Sex' = ten_genes$sex
)

#creates color palettes for annotations
annotationColors <- list(
  'Mechanical.Ventilation' = c('Yes' = blues[3], 'No' = blues[5]),
  'Sex' = c('Female' = yellows[3], 'Male' = yellows[5])
)

#creates heatmap
pheatmap(
  scaled,
  color = greens,
  cluster_rows = TRUE,
  cluster_cols = TRUE,
  fontsize = 8,
  main = "Gene Expression Heatmap by Sex and Mechanical Ventilation",
  show_rownames = FALSE,
  annotation_row = annotationData,
  annotation_colors = annotationColors
)
```



```r
#creates new dataframe and cleans and recodes data
hex <- metadata_BPI[(metadata_BPI$procalcitonin.ng.ml.. != "unknown") & (metadata_BPI$lactate.mmol.l. !=
hex$procalcitonin.ng.ml.. <- as.numeric(hex$procalcitonin.ng.ml..)
hex$lactate.mmol.l. <- as.numeric(hex$lactate.mmol.l.)
```

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

```r
hex$ferritin.ng.ml. <- as.numeric(hex$ferritin.ng.ml.)
hex$disease_status <- gsub("disease state: COVID-19", "COVID-19", hex$disease_status)
hex$disease_status <- gsub("disease state: non-COVID-19", "Non-COVID-19", hex$disease_status)

#creates hexbin plot
ggplot(hex, aes(x = age, y = BPI)) +
  geom_hex(binwidth = c(1.8, 40)) +
  labs(x = "Age", y = "Gene Expression", title = expression(paste("              Hexbin Plot of Age and
  theme_classic() +
  theme(legend.position = "top") +
  scale_fill_gradientn(colors = greens[3:9], name = "Counts")
```

```
## Warning: Removed 3 rows containing non-finite values (`stat_binhex()`).
```