

Project Presentation

Sai Lakkireddy

2023-08-08

Chosen Gene: AAMP

AAMP stands for “Angio-Associated Migratory Cell Protein” gene. This gene is responsible for making the AAMP protein, which is involved in cell movement and angiogenesis, contributing to processes like wound healing and tissue repair. Genes and their corresponding proteins are crucial for the proper functioning of our bodies.

In this analysis, we study the association between AAMP’s gene expression, age, COVID Status and ICU Status

Importing and combining the Data from two csv files

Steps:

1. Import both the csv files
2. Convert the gene expression into long format
3. Inner join it with meta data

```
#set current working directory to the previous folder
setwd("../")

#import the geneExpression and metaData csv from the data folder
data_gene_Expression <- read.csv("data/QBS103_finalProject_geneExpression.csv", header=TRUE)
data_meta <- read.csv("data/QBS103_finalProject_metadata.csv", header=TRUE)

#We convert the geneExpression data from wide form into long form
data_gene_Expression.longForm <- data_gene_Expression %>%
  pivot_longer(cols = starts_with(c("COVID_", "NONCOVID_")),
    names_to = "participant_id",
    values_to = "gene_expression_value"
  )

#make a final data frame by combining the two data sets and making it a data frame
final_df <- as.data.frame(data_gene_Expression.longForm %>% inner_join( data_meta,
  by=c('participant_id'))

#head(final_df)
```

Pre-processing the data

Steps:

1. Remove “unknown” strings and prefixes
2. Convert the class the columns to their appropriate type

```
#rename with x column with gene
final_df <- rename(final_df, gene = X)

#remove all unknown strings and substitute it with NAs
final_df[, 16:27][final_df[, 16:27] == ' unknown' | final_df[, 16:27] == 'unknown'] <- NA

#format the disease status column to just include the status
final_df$disease_status <- sub('disease state: ', '', final_df$disease_status)

#convert the column type of disease_status, sex, icu_status and mechanical_ventilation to factor
final_df <- final_df %>%
  mutate_at(vars(disease_status, sex, icu_status, mechanical_ventilation), as.factor)

#convert the class of age, charlson_score
final_df <- final_df %>%
  mutate_at(vars(age, apacheii, ferritin.ng.ml.,
                 crp.mg.l., ddimer.mg.l_feu.,
                 procalcitonin.ng.ml., lactate.mmol.l., fibrinogen, sofa), as.integer)

#head(final_df)
```

Optional - handle missing values

```
# check all the numeric columns
num_cols <- names(select_if(final_df, is.numeric))

# Create an imputation model
imputation_model <- mice(final_df[num_cols], method = "pmm", printFlag = FALSE)

# Perform the imputation
imputed_data_final <- complete(imputation_model)
# update the final data frame with the imputed values
final_df[num_cols] <- imputed_data_final[num_cols]

#head(final_df)
```

Create a subset the AAMP Gene and the chosen covariates

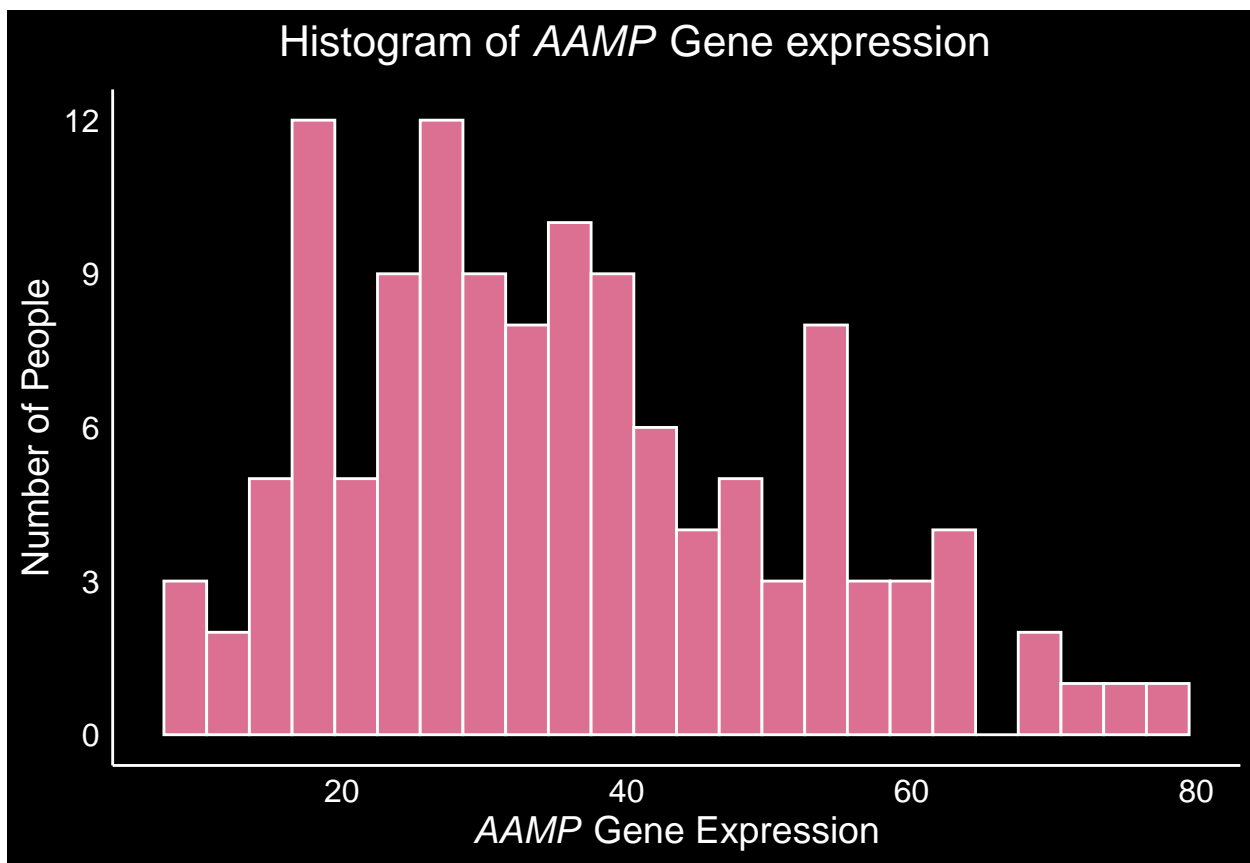
```
final_subset <- final_df[final_df$gene == 'AAMP',
  c('gene',
    'gene_expression_value', 'age',
    'icu_status', 'disease_status')]
```

```
#head(final_subset)
```

Histogram for Gene Expression

```
breaks <- seq(0, 15, by = 3)

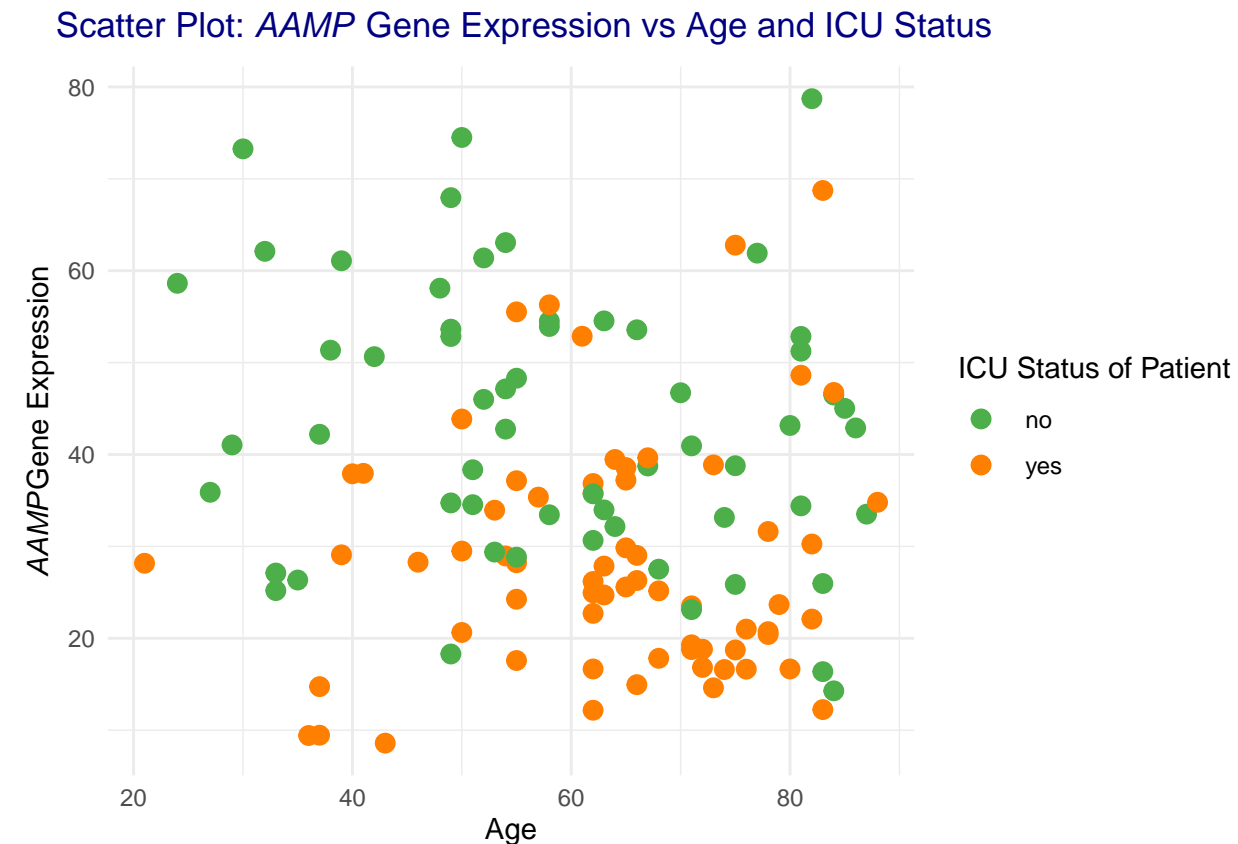
# Create the histogram with integer bins
ggplot(final_subset, aes(x = gene_expression_value)) +
  geom_histogram(binwidth = 3, color = "white", fill = "#DB7093") +
  scale_y_continuous(breaks = breaks) +
  theme_minimal() +
  theme(
    panel.background = element_rect(fill = "black"),
    plot.background = element_rect(fill = "black"),
    axis.line = element_line(color = "white"),
    axis.text = element_text(color = "white", size = 12),
    axis.title = element_text(color = "white", size = 14),
    panel.grid = element_blank(),
    plot.title = element_text(color = "white", size = 16, face = "bold", hjust = 0.4),
  ) +
  ggtitle(expression(paste("Histogram of ", italic("AAMP"), " Gene expression")))) +
  xlab(expression(paste(italic("AAMP"), " Gene Expression")))) +
  ylab("Number of People")
```



Scatter plot: Age vs Gene Expression factoring for ICU status

```
my_colors_1 <- c("#4DAF4A", "#FF7F00")

# Create the scatter plot with custom color scheme
ggplot(final_subset, aes(x = age, y = gene_expression_value, color = icu_status)) +
  geom_point(size = 3) +
  scale_color_manual(values = my_colors_1, name = "ICU Status of Patient") +
  theme_minimal() +
  theme(
    plot.title = element_text(color = "navy", size = 13, face = "bold", hjust = 0.4)
  ) +
  ggtitle(expression(paste("Scatter Plot: ", italic("AAMP"), " Gene Expression vs Age and ICU Status"))) +
  xlab("Age") +
  ylab(expression(paste(italic("AAMP"), "Gene Expression")))
```



Box plot: Gene Expression by COVID and ICU Status

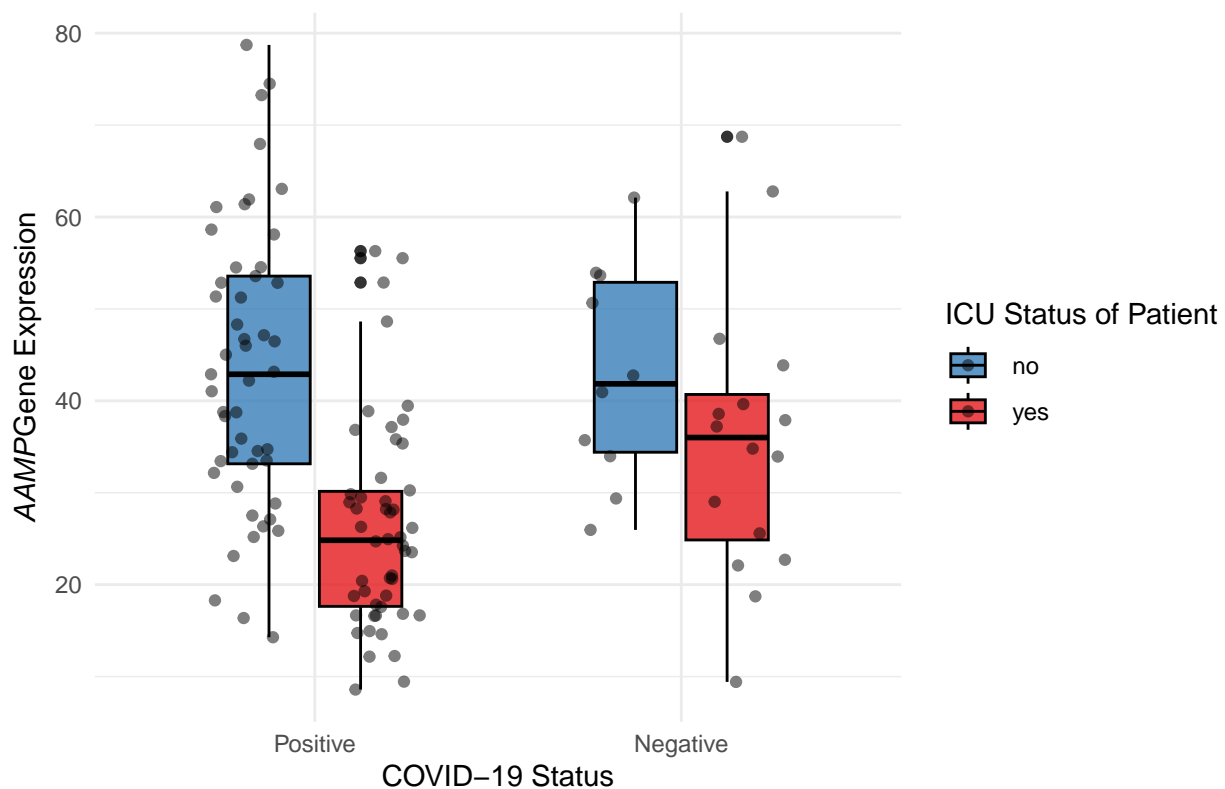
```
my_colors_2 <- c("#377eb8", "#e41a1c")
ggplot(final_subset, aes(x = disease_status, y = gene_expression_value, fill = icu_status)) +
  geom_boxplot(color = "black", width = 0.5, alpha = 0.8) +
```

```

scale_fill_manual(values = my_colors_2, name = "ICU Status of Patient") +
geom_jitter(position = position_jitterdodge(), alpha = 0.5) +
scale_x_discrete(labels = c("COVID-19" = "Positive", "non-COVID-19" = "Negative")) +
theme_minimal() +
  theme_minimal() +
  theme(
    plot.title = element_text(color = "darkgreen", size = 13, face = "bold", hjust = 0.4)
  )+
ggtitle(expression(paste("Box Plot: ",italic("AAMP")," Gene Expression by COVID and ICU Status")))+
xlab("COVID-19 Status") +
ylab(expression(paste(italic("AAMP"),"Gene Expression")))

```

Box Plot: *AAMP* Gene Expression by COVID and ICU Status



Plots generated by a function for genes ABHD18, AAMP and ABHD17C

```

my_plots_function <- function(dataFrame, genes.list, cont.covariate, cat.covariates) {
  #breaks <- seq(min(freque), 15, by = 3)
  my_colors_1 <- c("#4DAF4A", "#FF7F00")
  my_colors_2 <- c("#377eb8", "#e41a1c")
  all.plots.list <- list()

  for (gene in genes.list) {
    gene_subset <- final_df[final_df$gene == gene,
      c('gene','gene_expression_value',paste(cont.covariate), paste(cat.covariates

```

```

    histogram <- ggplot(gene_subset, aes(x = gene_expression_value)) +
    geom_histogram(binwidth = 1, color = "white", fill = "#DB7093") +
    #scale_y_continuous(breaks = breaks) +
    theme_minimal() +
    theme(
      panel.background = element_rect(fill = "black"),
      plot.background = element_rect(fill = "black"),
      axis.line = element_line(color = "white"),
      axis.text = element_text(color = "white", size = 12),
      axis.title = element_text(color = "white", size = 14),
      panel.grid = element_blank(),
      plot.title = element_text(color = "white", size = 16, face = "bold", hjust = 0.4),
    ) +
    ggtitle(substitute(Histogram ~ of ~ italic(gene) ~ Gene ~ expression, list(gene = gene))) +
    xlab(substitute(italic(gene) ~ Gene ~ Expression, list(gene = gene))) +
    ylab("Number of People")

    scatter.plot <- ggplot(gene_subset, aes(y = gene_expression_value, x = gene_subset[[cont.covariate]]),
    geom_point(size = 3) +
    scale_color_manual(values = my_colors_1, name = "ICU Status of Patient") +
    theme_minimal() +
    theme(
      plot.title = element_text(color = "navy", size = 13, face = "bold", hjust = 0.4)
    ) +
    ggtitle(substitute(Scatter ~ Plot ~ italic(gene) ~ Gene ~ Expression ~ vs ~ Age ~ and ~ ICU ~ Status))
    xlab(substitute(cont.covariate)) +
    ylab(substitute(italic(gene) ~ Gene ~ Expression))

    box.plot <- ggplot(gene_subset, aes(x = gene_subset[[cat.covariates[1]]], y = gene_expression_value, fill =
    geom_boxplot(color = "black", width = 0.5, alpha = 0.8) +
    scale_fill_manual(values = my_colors_2, name = "ICU Status of Patient") +
    geom_jitter(position = position_jitterdodge(), alpha = 0.5) +
    scale_x_discrete(labels = c("COVID-19" = "Positive", "non-COVID-19" = "Negative")) +
    theme_minimal() +
    theme(
      plot.title = element_text(color = "darkgreen", size = 13, face = "bold", hjust = 0.4)
    ) +
    ggtitle(substitute(Box ~ plot ~ of ~ italic(gene) ~ Gene ~ Expression ~ by ~ COVID ~ and ~ ICU ~ Status))
    xlab("COVID-19 Status") +
    ylab(substitute(italic(gene) ~ Gene ~ Expression))

    all.plots.list[[gene]] <- list(histogram = histogram, scatter.plot = scatter.plot, box.plot = box.plot)
  }

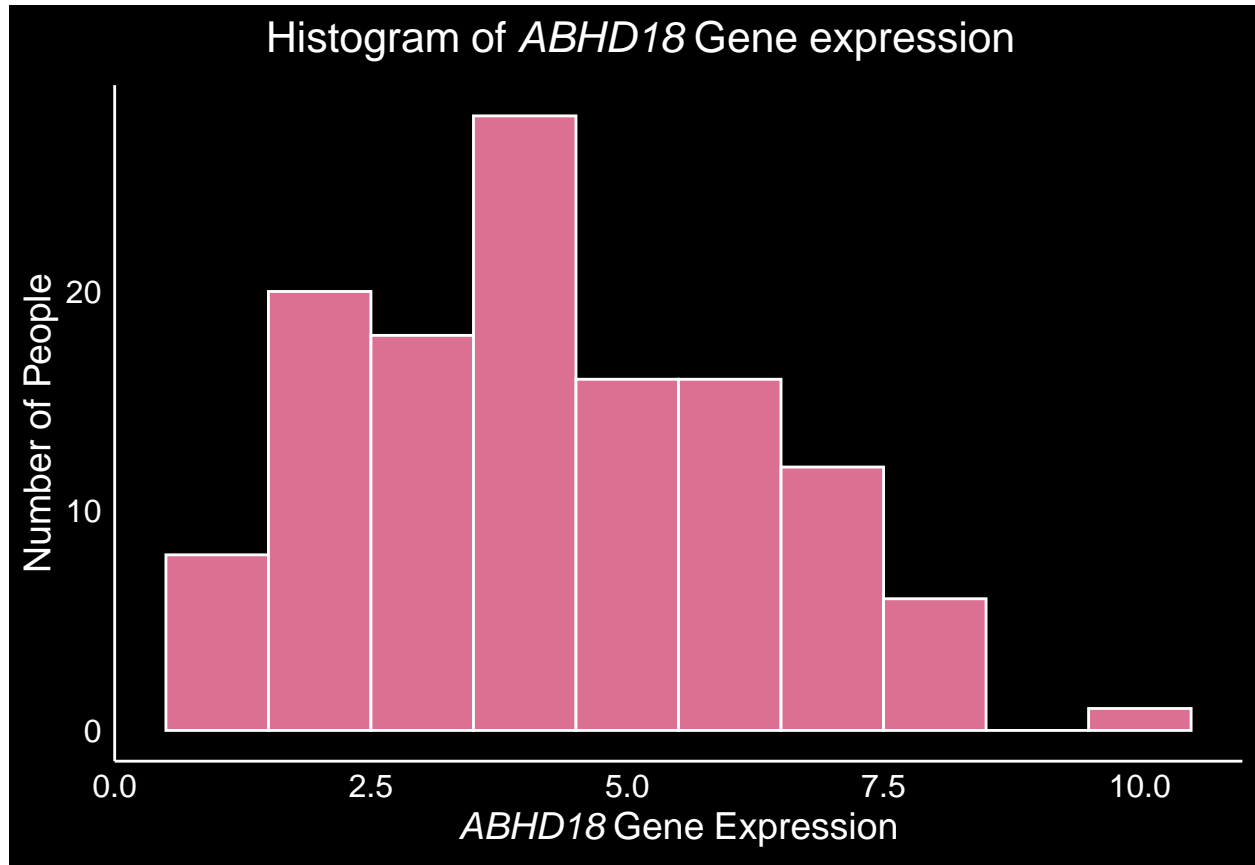
  return(all.plots.list)
}

all.plots.list <- my_plots_function(final_subset, c('ABHD18', 'AAMP', 'ABHD17C'), 'age', c('disease_status', 'icu_status'))

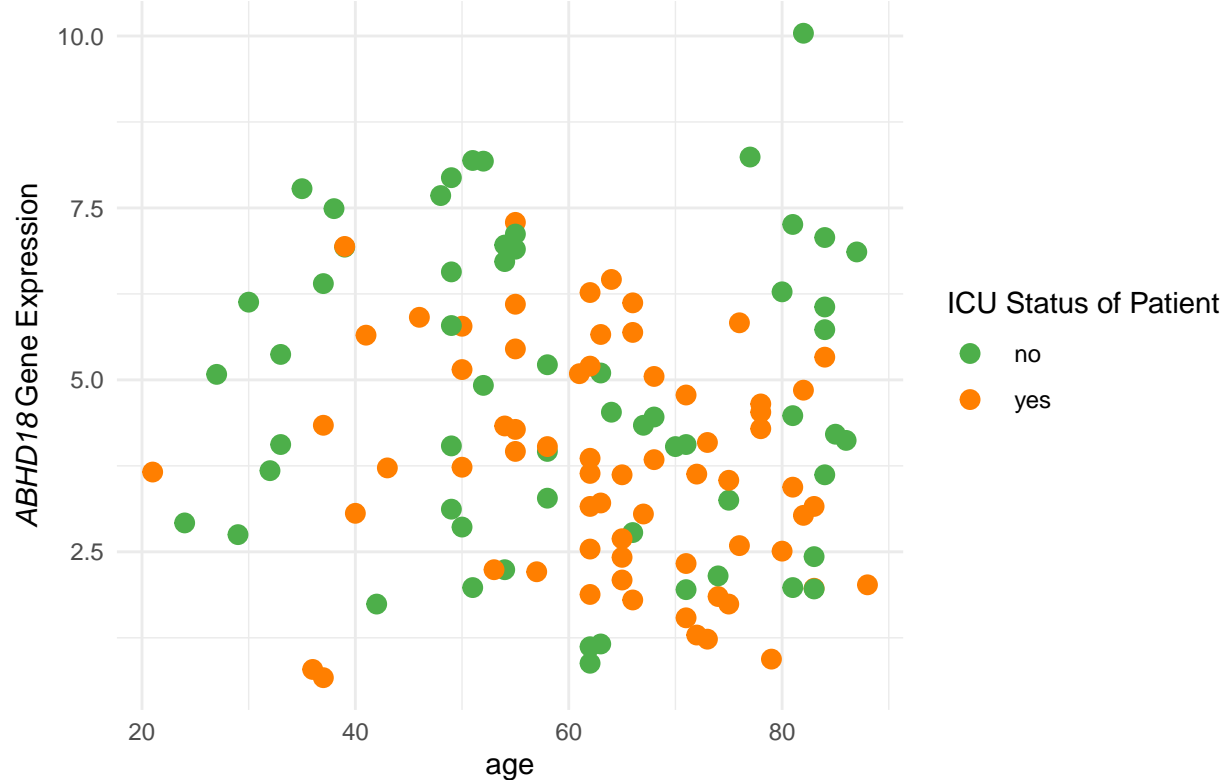
for (gene in names(all.plots.list)) {

```

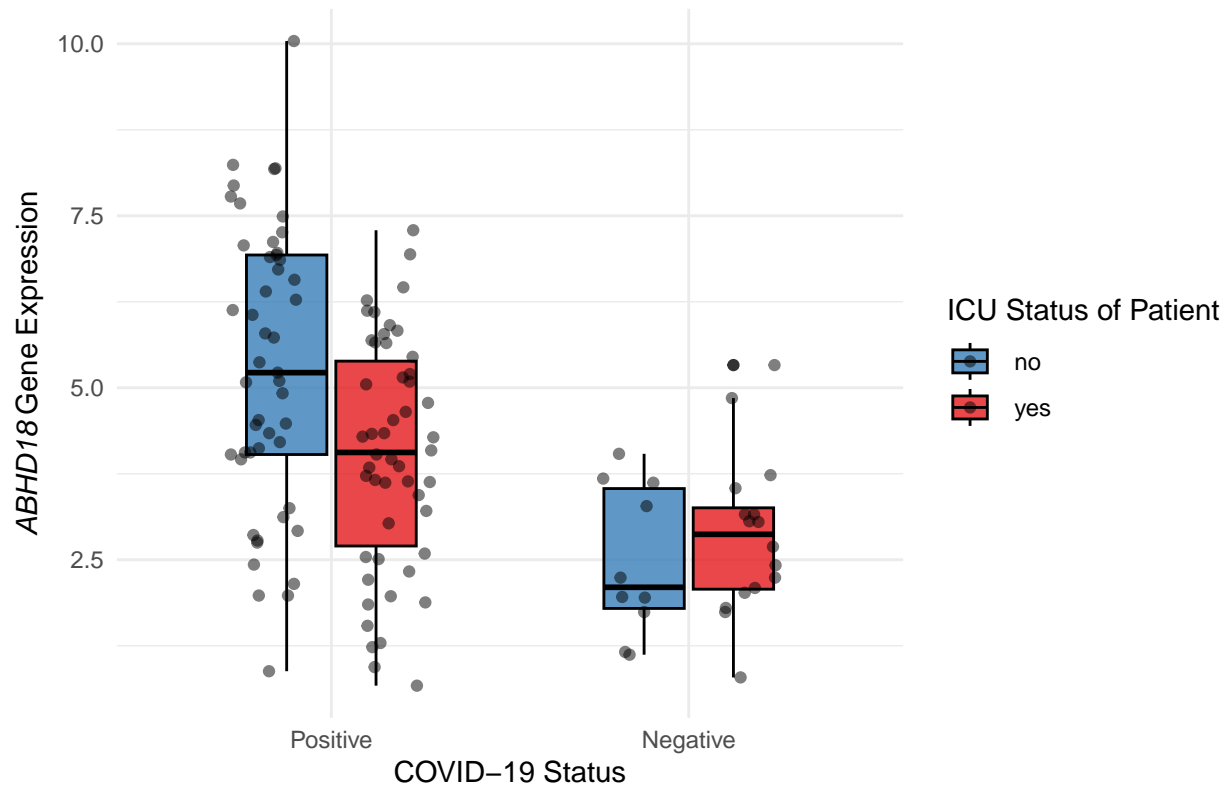
```
print(all.plots.list[[gene]]$histogram)
print(all.plots.list[[gene]]$scatter.plot)
print(all.plots.list[[gene]]$box.plot)
}
```

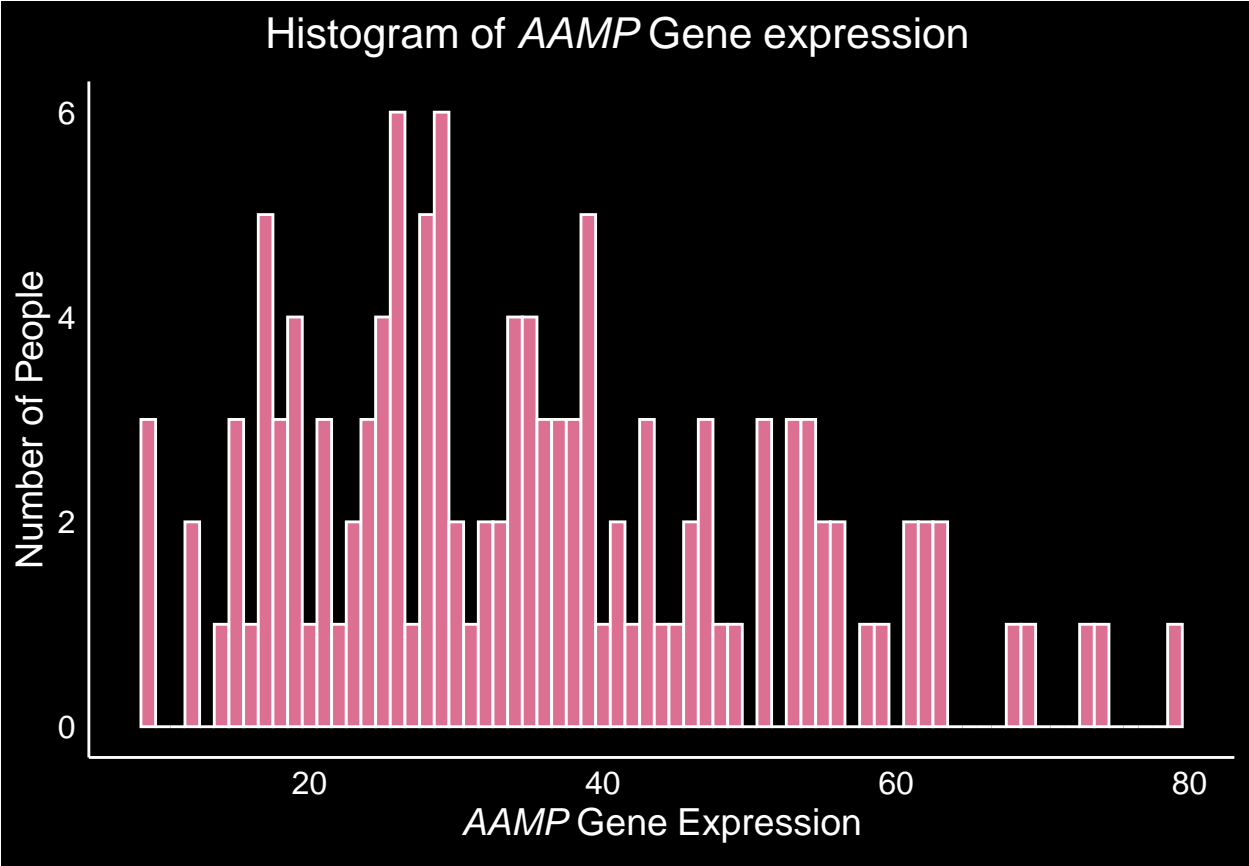


Scatter Plot *ABHD18* Gene Expression vs Age and ICU Status

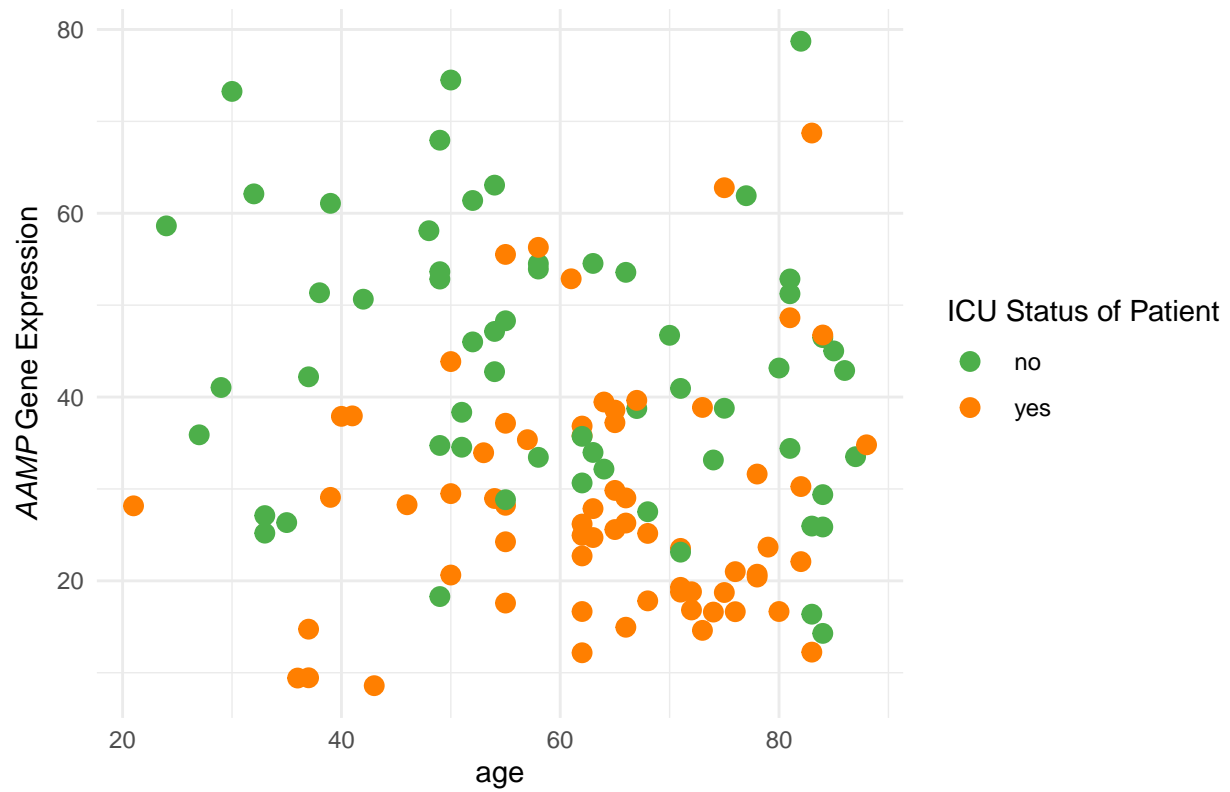


Box plot of *ABHD18* Gene Expression by COVID and ICU Status

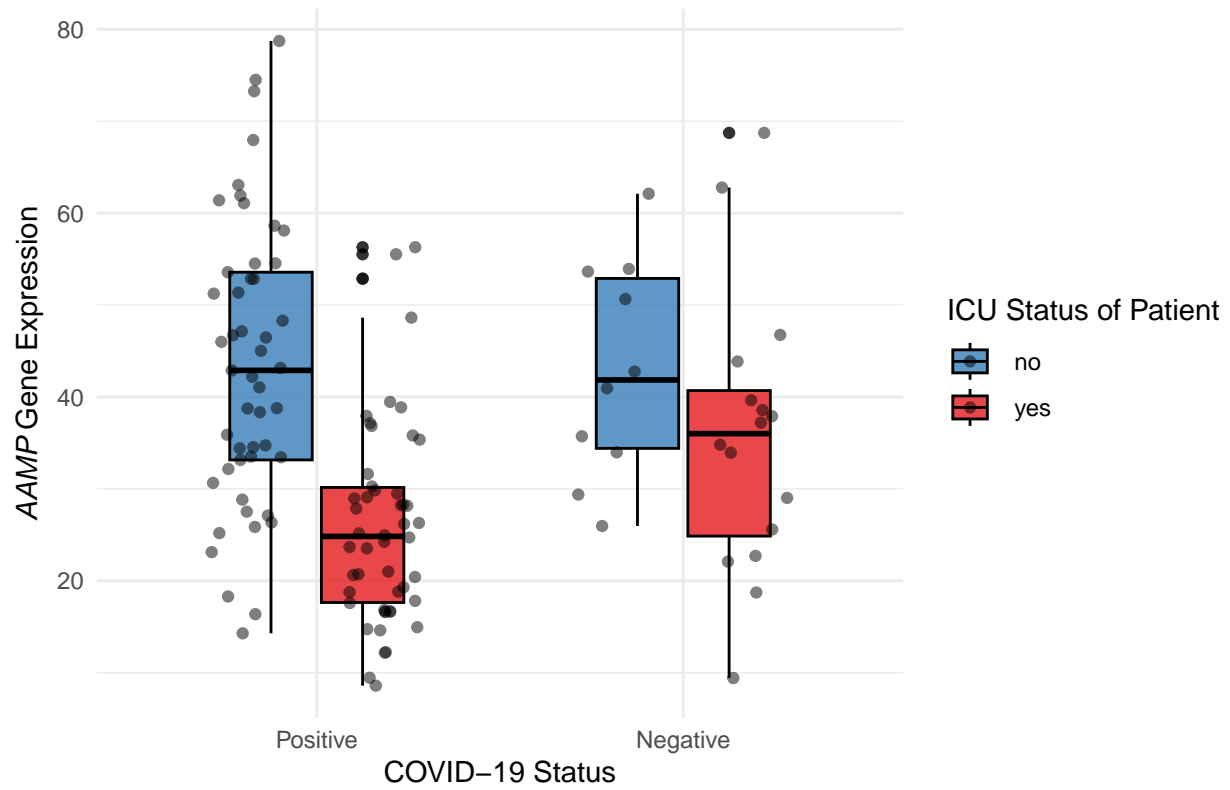


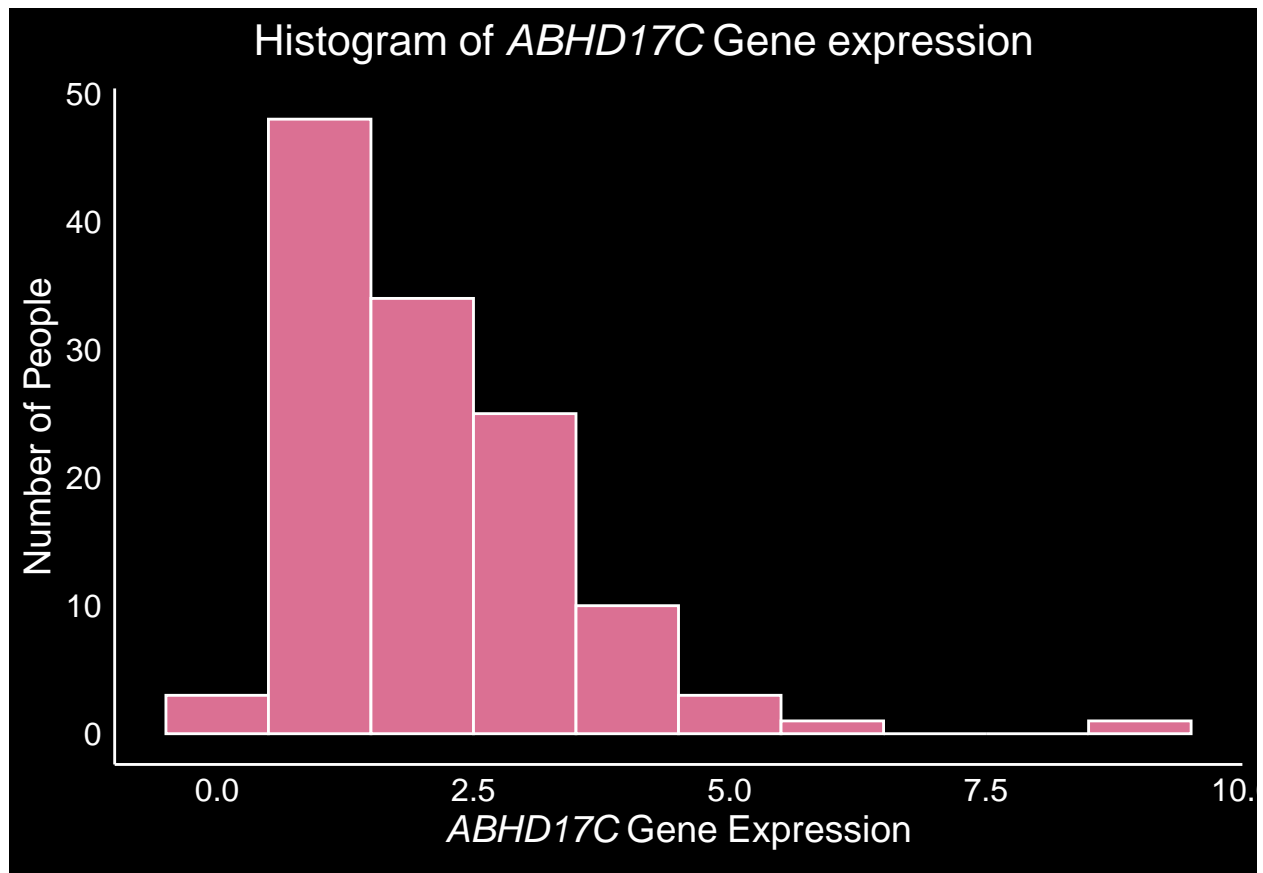


Scatter Plot AAMP Gene Expression vs Age and ICU Status

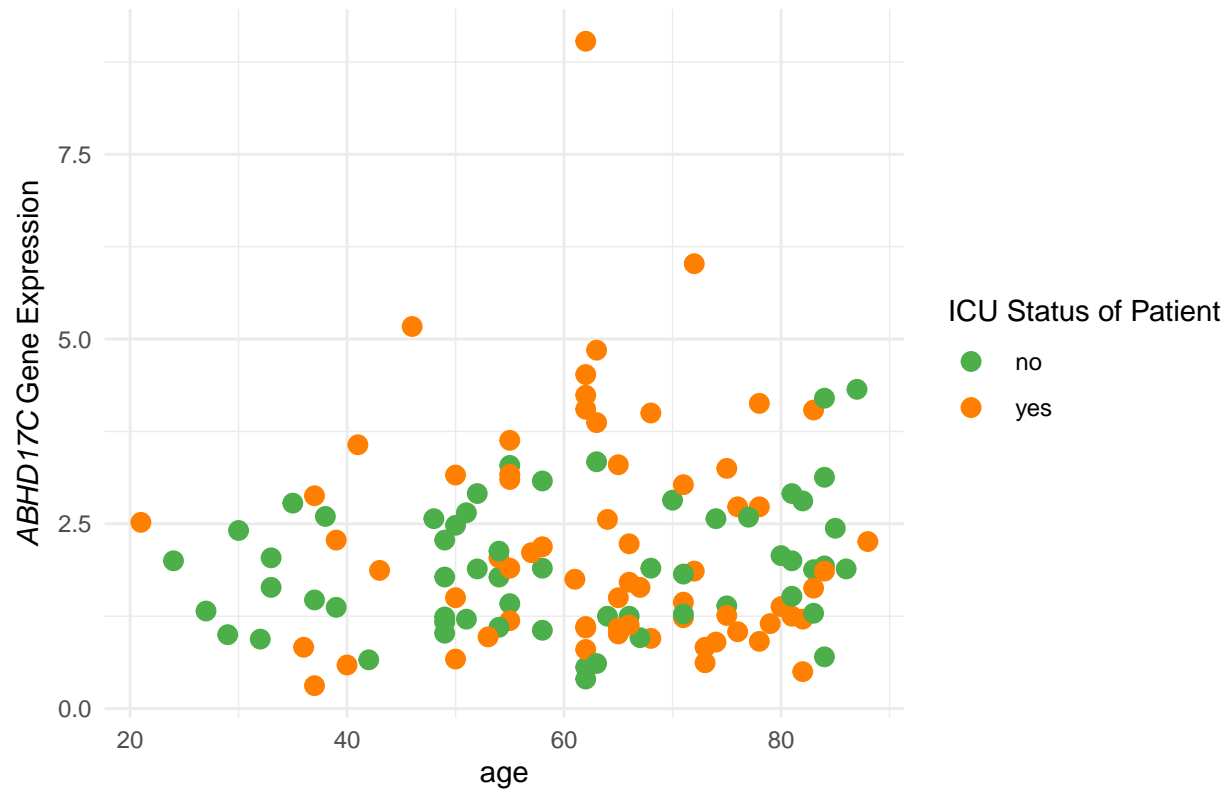


Box plot of *AAMP* Gene Expression by COVID and ICU Status





Scatter Plot *ABHD17C* Gene Expression vs Age and ICU Status



Box plot of *ABHD17C* Gene Expression by COVID and ICU Status

