R Projects

2024-07-25

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.

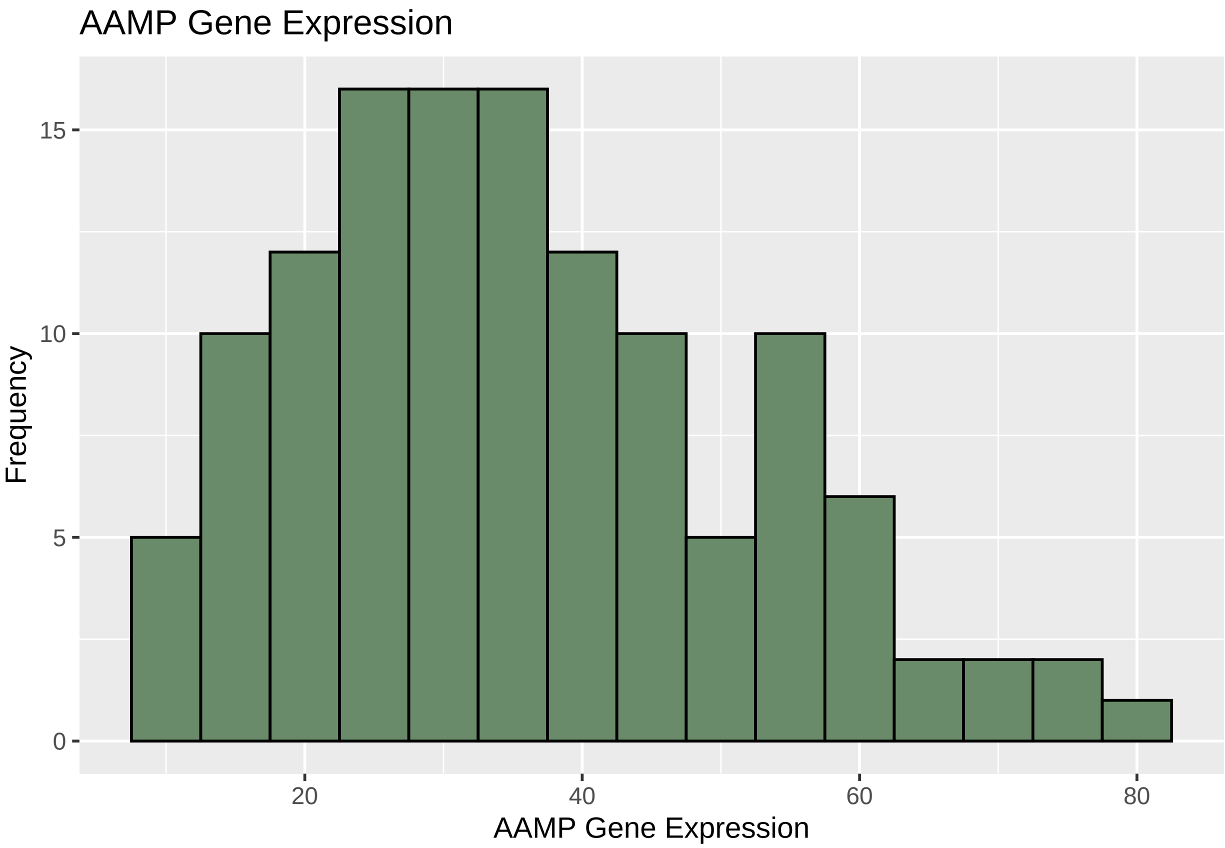
When you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document. You can embed an R code chunk like this:

```
# READ IN CSV FILES

setwd("/Users/sarahmirza/Documents/GitHub/QBS103_Repository/") # set the working directory
genes.df <- read.csv(file = "QBS103_GSE157103_genes.csv")
#head(genes.df) # see if it worked



matrix.df <- read.csv(file="QBS103_GSE157103_series_matrix.csv")
#head(matrix.df)
```

```
# GENES DATA FRAME - transpose

# remove column names and rows before transposing then add them back
genes_transpose <-t(genes.df)
genes.df <- as.data.frame(genes_transpose) # transpose, convert rows to columns and columns to rows
names(genes.df) <- genes.df[1,] #set the first row in the data frame and set it to the column names for
genes.df <- genes.df[-1,] # remove the first row in the data frame so that the names are no longer a ro


# FROM TUTORIALSPOINT - the [] maintains the data frame as a data frame because the lapply works on a l
# has been converted to a numeric because transpose makes them character
genes.df[] <- lapply(genes.df, function(x) as.numeric(as.character(x)))

genes.df <- na.omit(genes.df) # get rid of NA
```

```
genes.df$participant_id <- row.names(genes.df) # make new row called participant_id and with the gene.d
merged_matrix <- merge(genes.df,matrix.df,by = "participant_id") # merge the two data frames using part
#head(merged_matrix) # see if it worked
```

```
# Load required package
library(ggplot2)

# Convert AAMP to numeric to read into histogram easier
merged_matrix$AAMP <- as.numeric(merged_matrix$AAMP)

# Create histogram using the new data frame, the gene chosen was AAMP - changed binwidth per suggestion
histo <- ggplot(merged_matrix, aes(x = AAMP)) +
  geom_histogram(binwidth = 5, fill = "darkseagreen4", color = "black") +
  labs(title = "AAMP Gene Expression",
       x = "AAMP Gene Expression",
       y = "Frequency")
plot(histo)
```

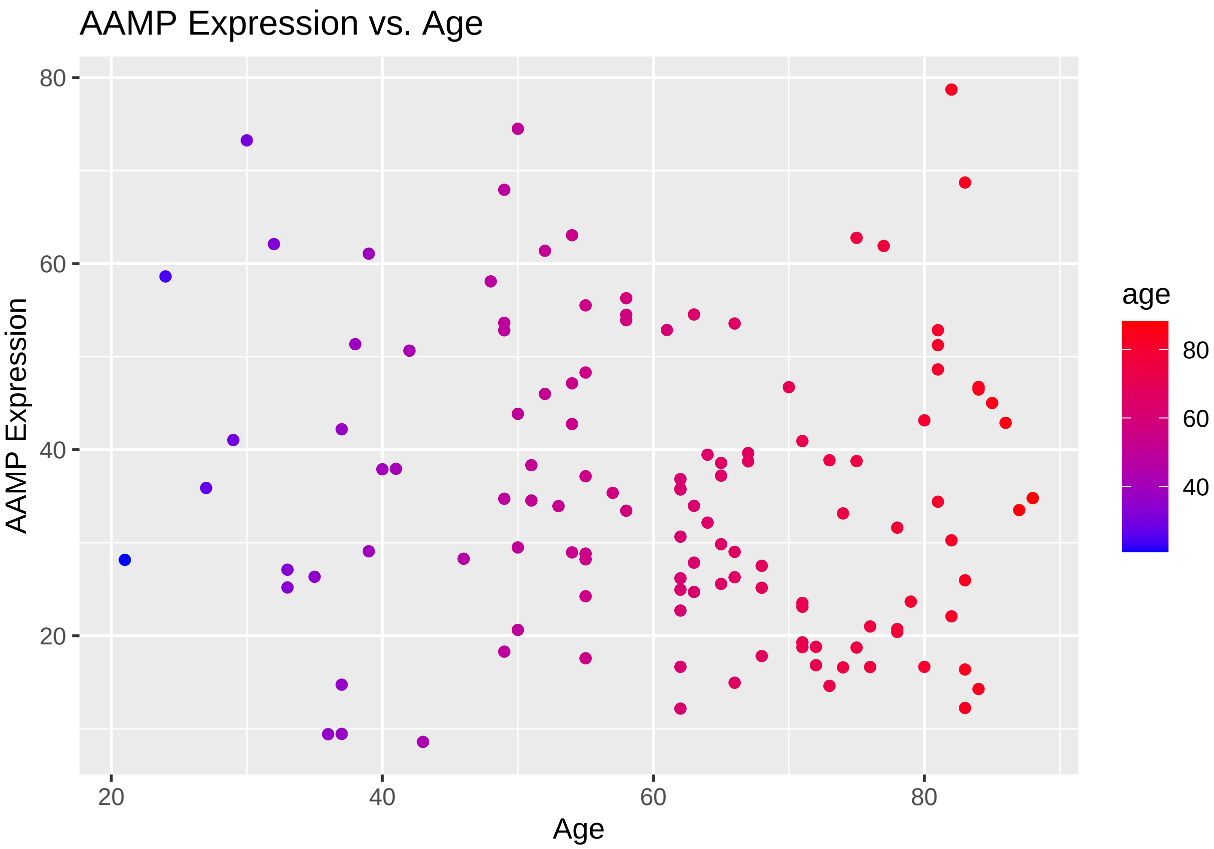


```
library(ggplot2)
setwd("/Users/sarahmirza/Documents/GitHub/QBS103_Repository/")

# Convert columns to numeric for a gradient label
merged_matrix$AAMP <- as.numeric(merged_matrix$AAMP)
merged_matrix$age <- as.numeric(merged_matrix$age)
```

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

```
# Create scatterplot of AAMP expression vs. age
scatter <- ggplot(merged_matrix, aes(x = age, y = AAMP,color = age)) + # color = age gives the color ba
  geom_point() +
  scale_color_gradient(low = "blue", high = "red") +
  labs(title = "AAMP Expression vs. Age",
       x = "Age",
       y = "AAMP Expression")
plot(scatter)
```

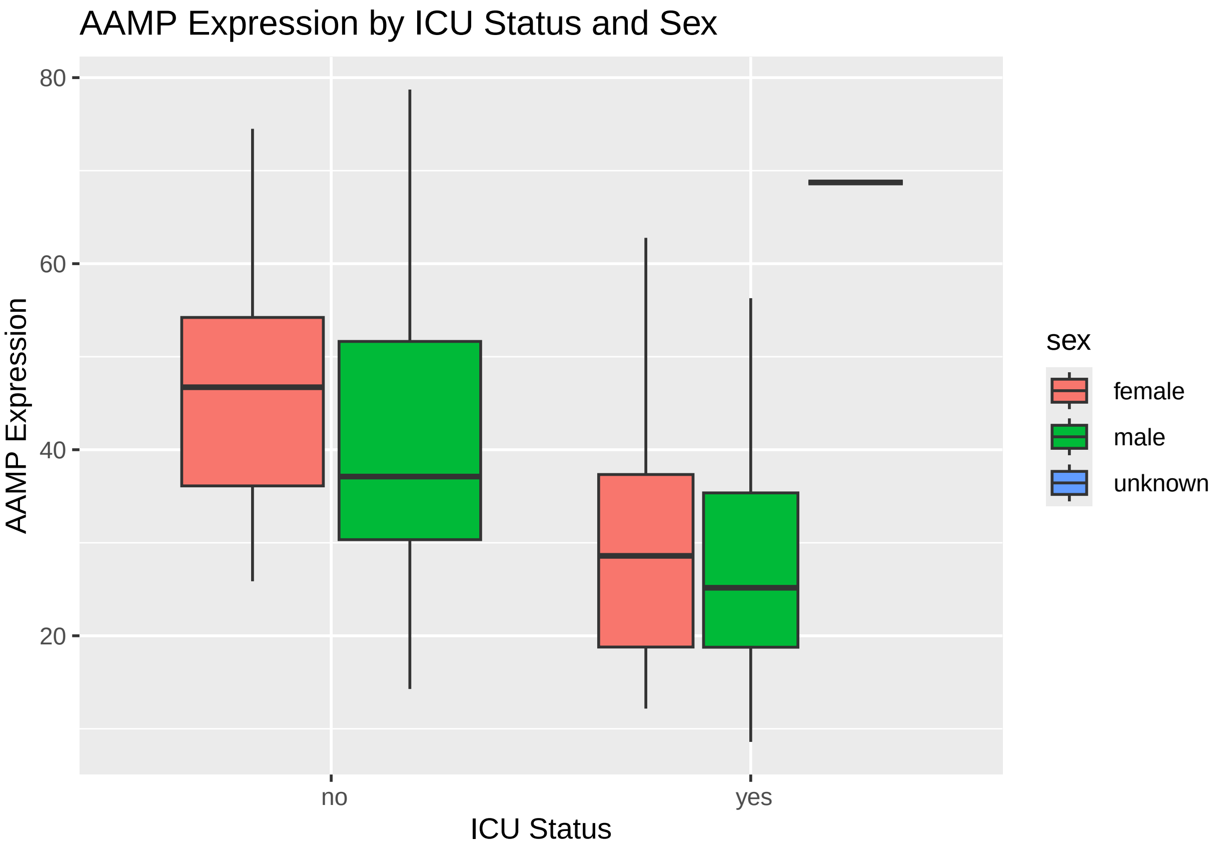
```
## Warning: Removed 2 rows containing missing values or values outside the scale range
## (`geom_point()`).
```



```
# geom_smooth for best fit line

# Save the plot to a file
ggsave("AAMP_gene_expression_vs_age.pdf", plot = scatter, width = 8, height = 5)
```

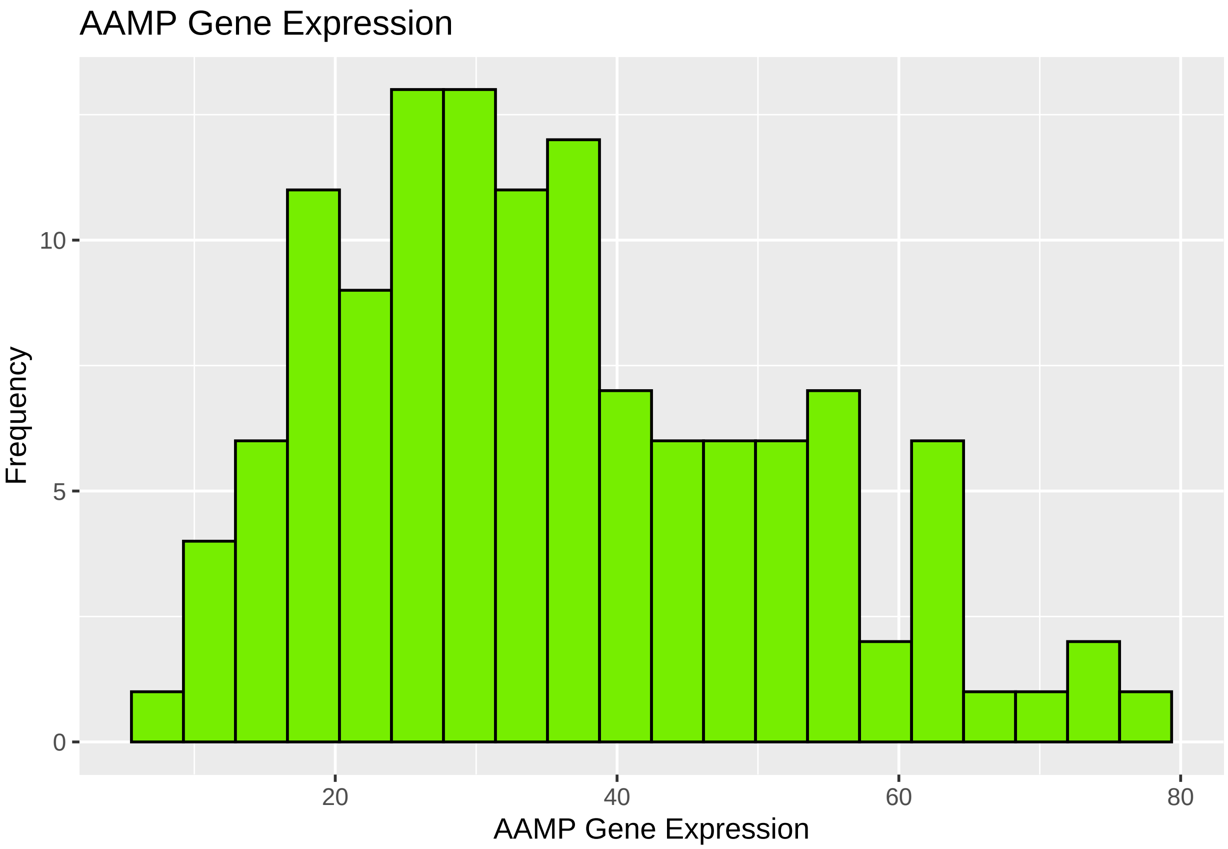
```
## Warning: Removed 2 rows containing missing values or values outside the scale range
## (`geom_point()`).
```

```
library(ggplot2)
setwd("/Users/sarahmirza/Documents/GitHub/QBS103_Repository/")


#a boxplot of AAMP expression separated by sex and ICU Status
ggplot(merged_matrix, aes(x = icu_status, y = AAMP, fill = sex)) +
  geom_boxplot() +
  labs(title = "AAMP Expression by ICU Status and Sex",
       x = "ICU Status",
       y = "AAMP Expression",
       fill = "sex")
```

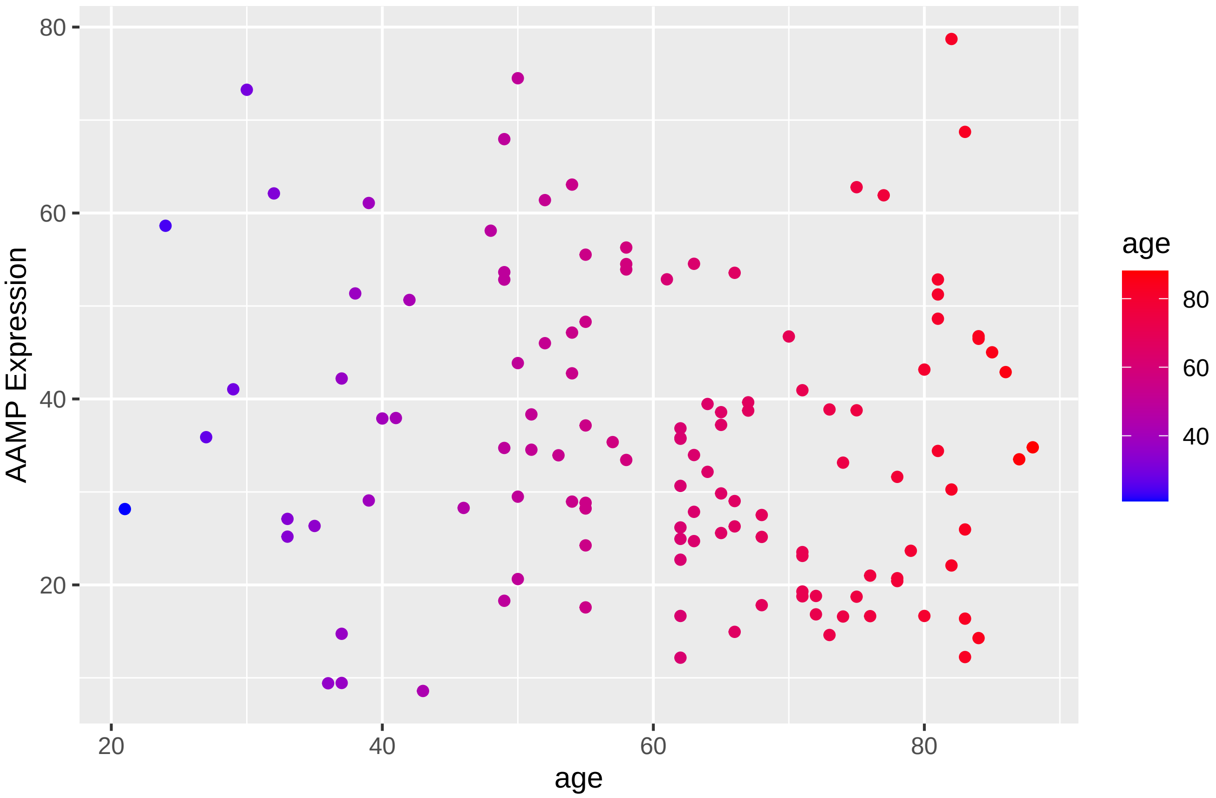


```
#merged_matrix$ferritin.ng.ml. <- as.factor(merged_matrix$ferritin.ng.ml.)
#merged_matrix$icu_status <- as.factor(merged_matrix$icu_status)

# Create a boxplot of AAMP expression separated by icu status and ferritin.ng.ml.
#ggplot(merged_matrix, aes(x = icu_status, y = AAMP, fill = ferritin.ng.ml.)) +
#  geom_boxplot() +
#  labs(title = "AAMP Expression by ICU Status and ferritin.ng.ml.",
#       x = "ICU Status",
#       y = "AAMP Expression",
#       fill = "ferritin.ng.ml.")

# Create a boxplot of AAMP expression separated by source_name_ch1 and icu status
#ggplot(merged_matrix, aes(icu_status, y = AAMP, fill = source_name_ch1)) +
#  geom_boxplot() +
#  labs(title = "AAMP Expression by ICU Status and source_name_ch1",
#       x = "icu_status",
#       y = "AAMP Expression",
#       fill = "source_name_ch1")
```

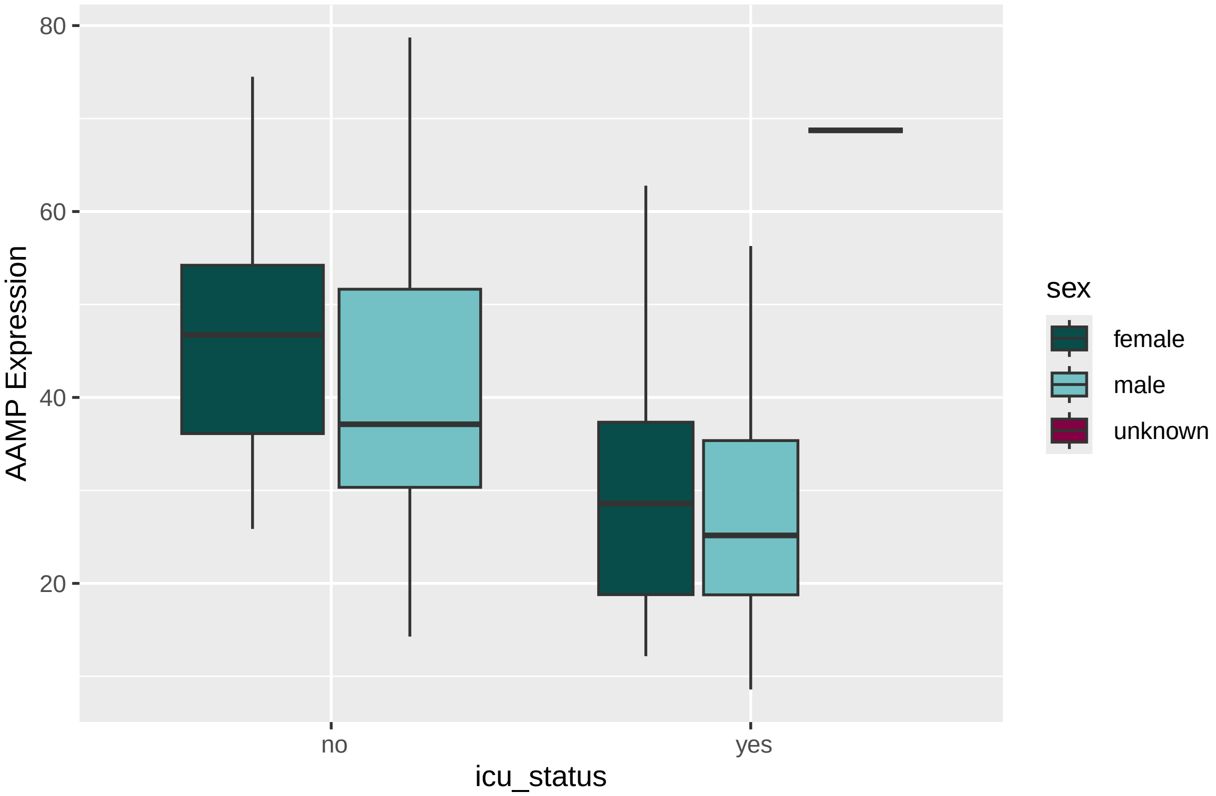
```
library(harrypotter)

# function to plot all three plots
fun_stats_pretty_plots <- function(matrix,gene_name,continuous_name,categorical1_name,categorical2_name
 # print(matrix)
    matrix$gene <- matrix[,gene_name] # dummy variable that creates a new column with the name of the g
    # this is used for plotting, and the function is fed a string that I use for labeling which is the
    #matrix$gene <- as.numeric(matrix$gene)
    matrix$continuous <- matrix[,continuous_name] # do the same for all variables so they are read in a
    matrix$categorical1 <- matrix[,categorical1_name]
    matrix$categorical2 <- matrix[,categorical2_name]

    histogram <- ggplot(matrix, aes(x=gene)) + geom_histogram(bins = 20, fill = "chartreuse2", color =
    labs(title = paste0(gene_name," Gene Expression"),
       x = paste0(gene_name," Gene Expression"),
       y = "Frequency")

    scatterplot <- ggplot(matrix, aes(x=continuous, y=gene, color = continuous)) +
    geom_point() + scale_color_gradient(low = "blue", high = "red",name = (paste0(continuous_name))) +
    labs(title = paste0(gene_name," Expression vs. ",continuous_name),
       x = paste0(continuous_name),
       y = paste0(gene_name," Expression"))


    boxplot <- ggplot(matrix, aes(x=categorical1, y=gene, fill = categorical2)) +
    geom_boxplot() + scale_fill_hp_d(option = "lunalovegood") +
    labs (title = paste0(gene_name," Expression by ",categorical1_name, " and ",categorical2_name),
       x = paste0(categorical1_name),
       y = paste0(gene_name," Expression"),
       fill = paste0(categorical2_name))

    plot(histogram)
    plot(scatterplot)
    plot(boxplot)
}

# genes to be used during plotting - stored in a list
plot_genes = c("AAMP","AAK1","ABCA7")

# for loop - replace gene name with gene in list
for (g in plot_genes) {
  fun_stats_pretty_plots(matrix = merged_matrix,gene_name=g,continuous_name = 'age',categorical1_name =
}
```



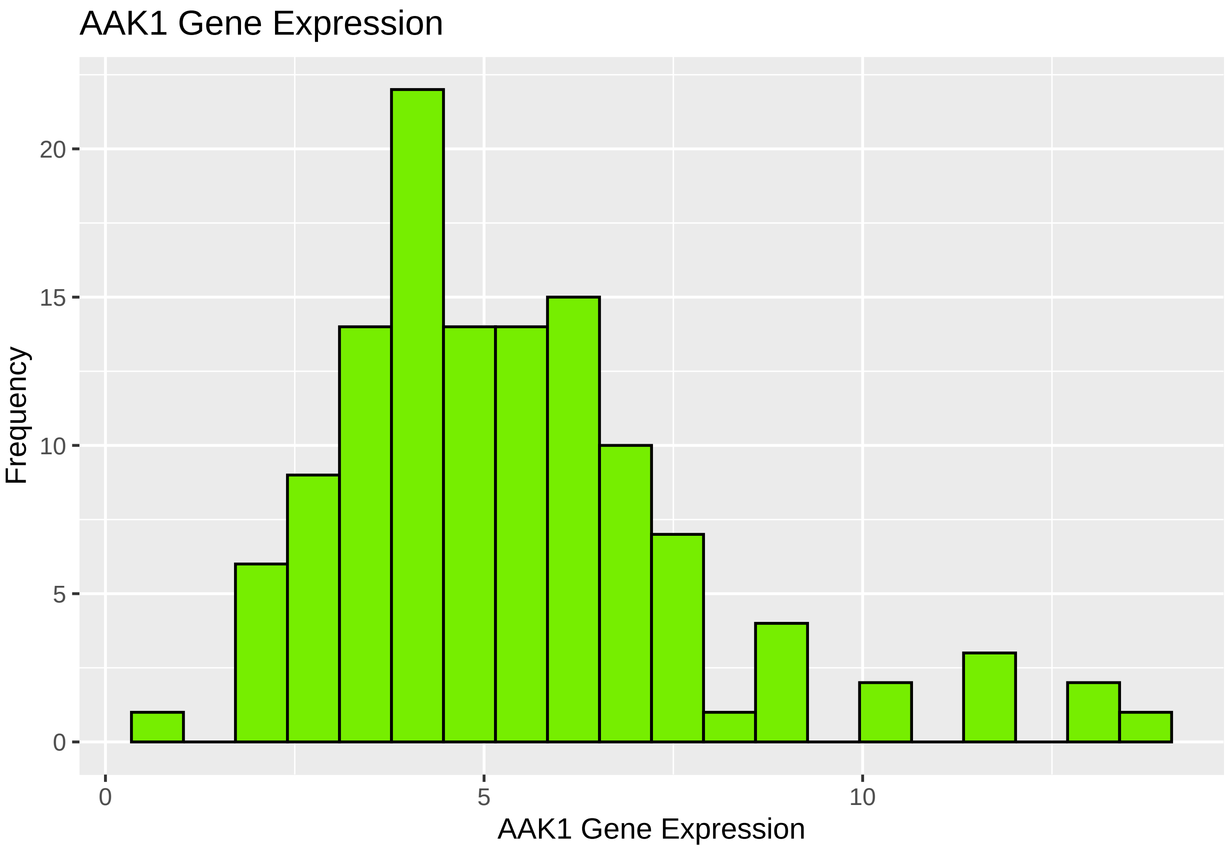
```
## Warning: Removed 2 rows containing missing values or values outside the scale range
## (`geom_point()`).
```



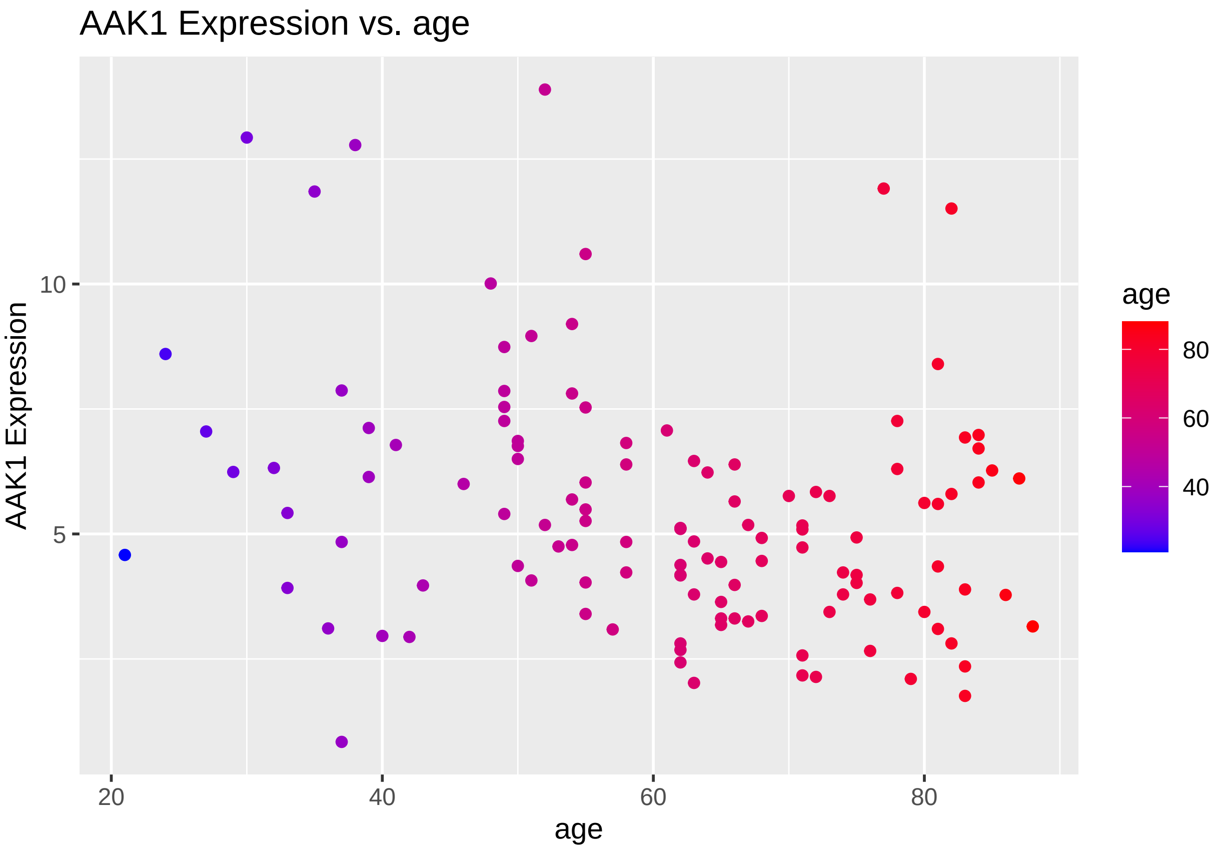
AAMP Expression vs. age



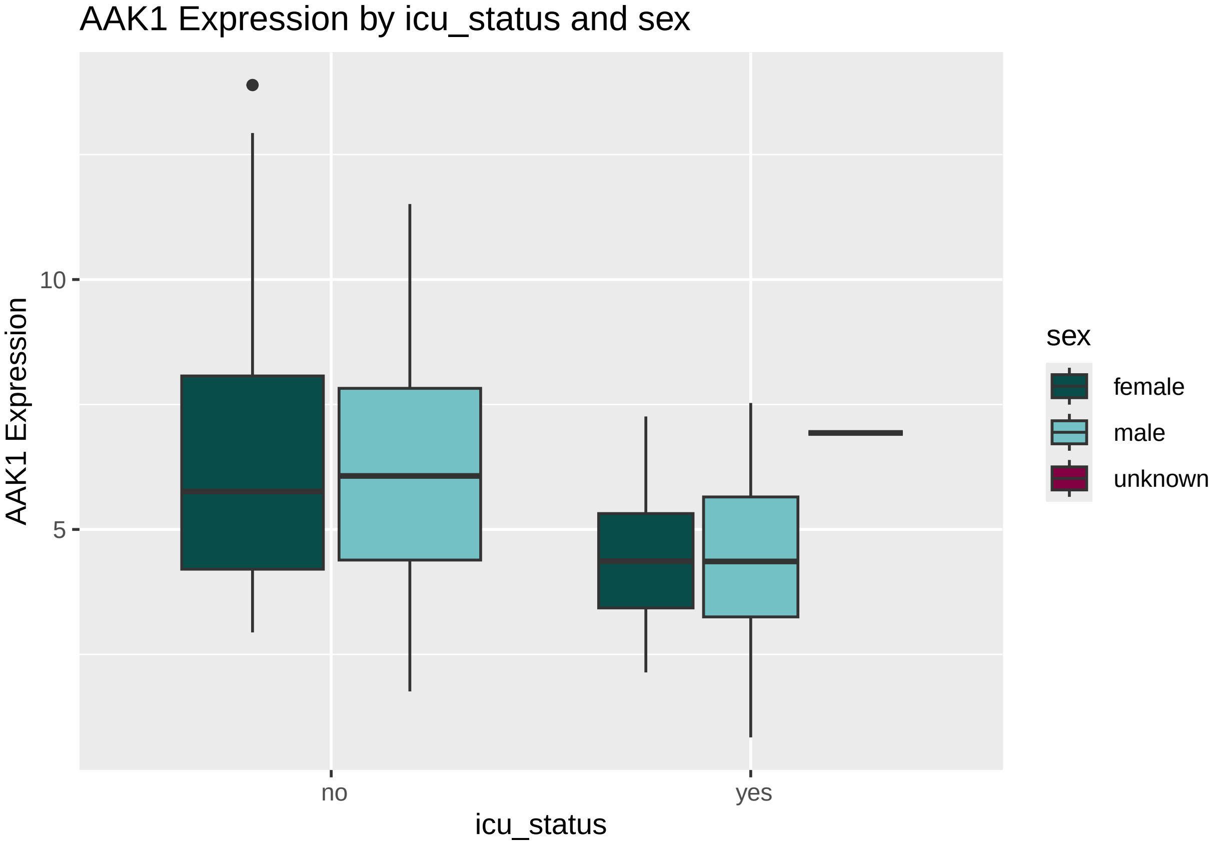
AAMP Expression by icu_status and sex



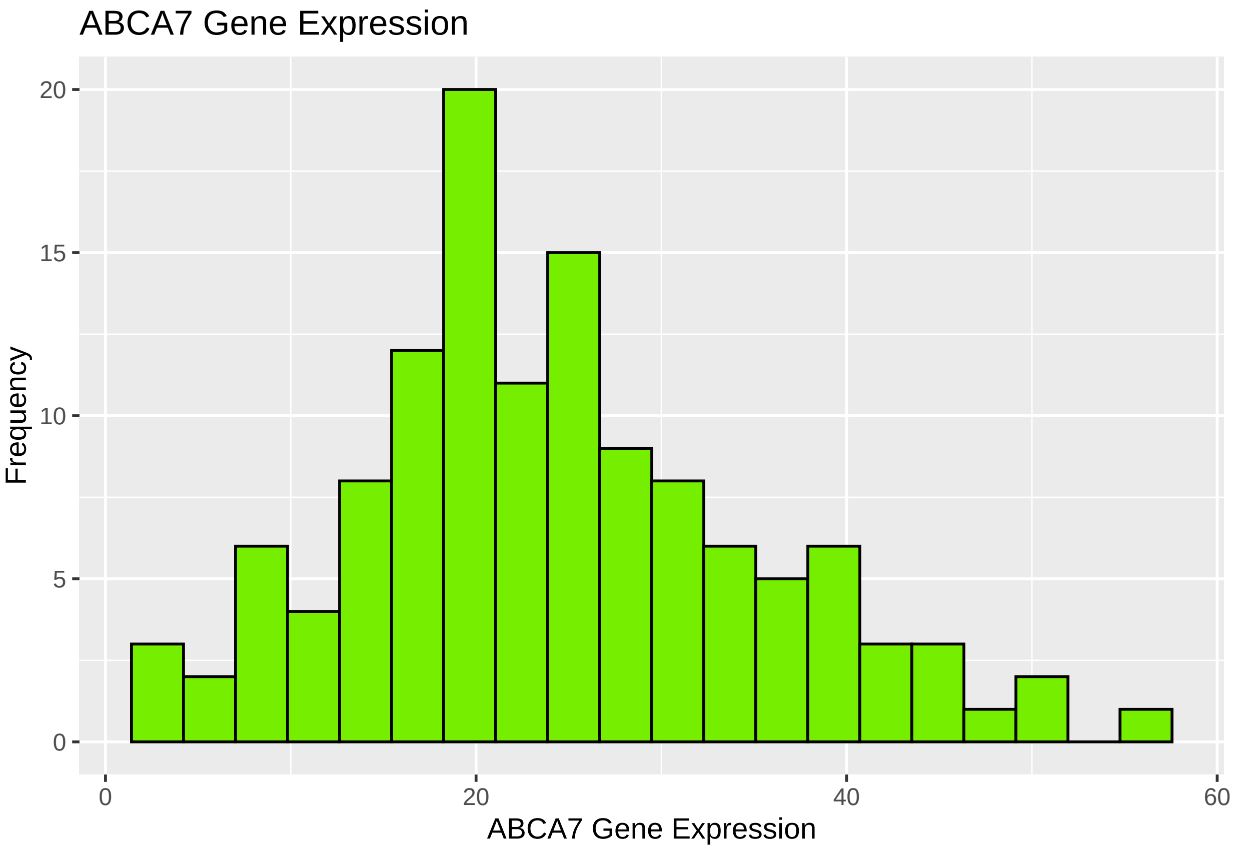
```
## Warning: Removed 2 rows containing missing values or values outside the scale range
## (`geom_point()`).
```



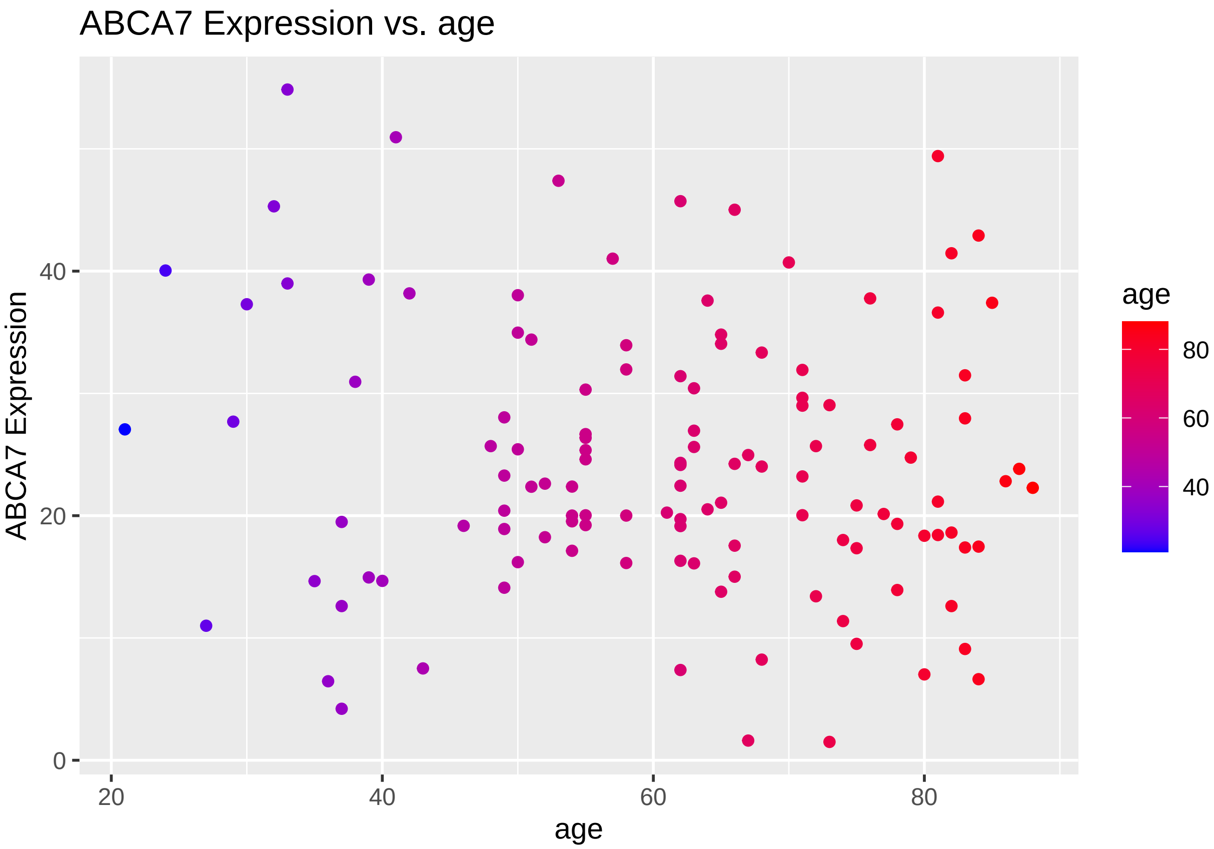
AAK1 Expression vs. age

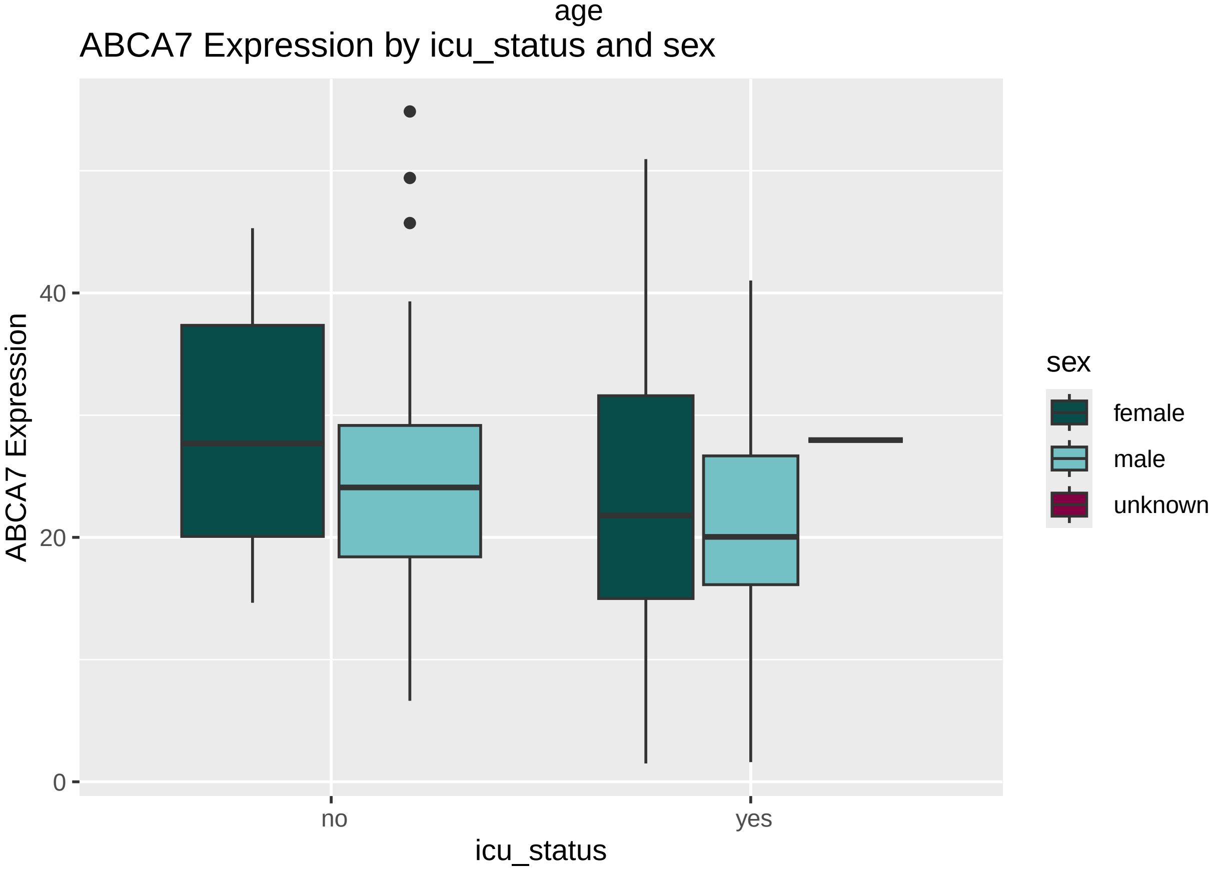


AAK1 Expression by icu_status and sex



```
## Warning: Removed 2 rows containing missing values or values outside the scale range
## (`geom_point()`).
```


ABCA7 Expression vs. age
ABCA7 Expression
0
20
40
20
40
60
80
age
age
80
60
40


ABCA7 Expression by icu_status and sex
ABCA7 Expression
0
20
40
no
yes
icu_status
sex
female
male
unknown

```
# gene 1
#fun_stats_pretty_plots(matrix = merged_matrix,gene_name='AAMP',continuous_name = 'age',categorical1_nam


# gene 2
#fun_stats_pretty_plots(matrix = merged_matrix,gene_name='AAAS',continuous_name = 'age',categorical1_nam


# gene 3
#fun_stats_pretty_plots(matrix = merged_matrix,gene_name='ABHD14A-ACY1',continuous_name = 'age',categor
```