Final Project 2

Annie Glenning

2024-08-06

Getting the Data

```r
getwd() # finding my working directory
```

```
## [1] "/Users/annieglenning/Documents/GitHub/QBS103"
```

```r
# set my working directory to where the data is stored
setwd("/Users/annieglenning/Documents/Dartmouth/SU24/QBS_103/Data")

# reading in the data
original_gene_data <- read.table("QBS103_GSE157103_genes.csv", header = TRUE, sep = ",")
series_data <- read.table("QBS103_GSE157103_series_matrix.csv", header = TRUE, sep = ",")
```

```r
# transposing the gene data
original_gene_data <- as.data.frame(original_gene_data)
if (is.character(original_gene_data[1, 1])) {
  gene_data <- t(original_gene_data)
  colnames(gene_data) <- gene_data[1, ]
  gene_data <- gene_data[-1, ]
  gene_data <- as.data.frame(gene_data)
}
```

Building the Functions

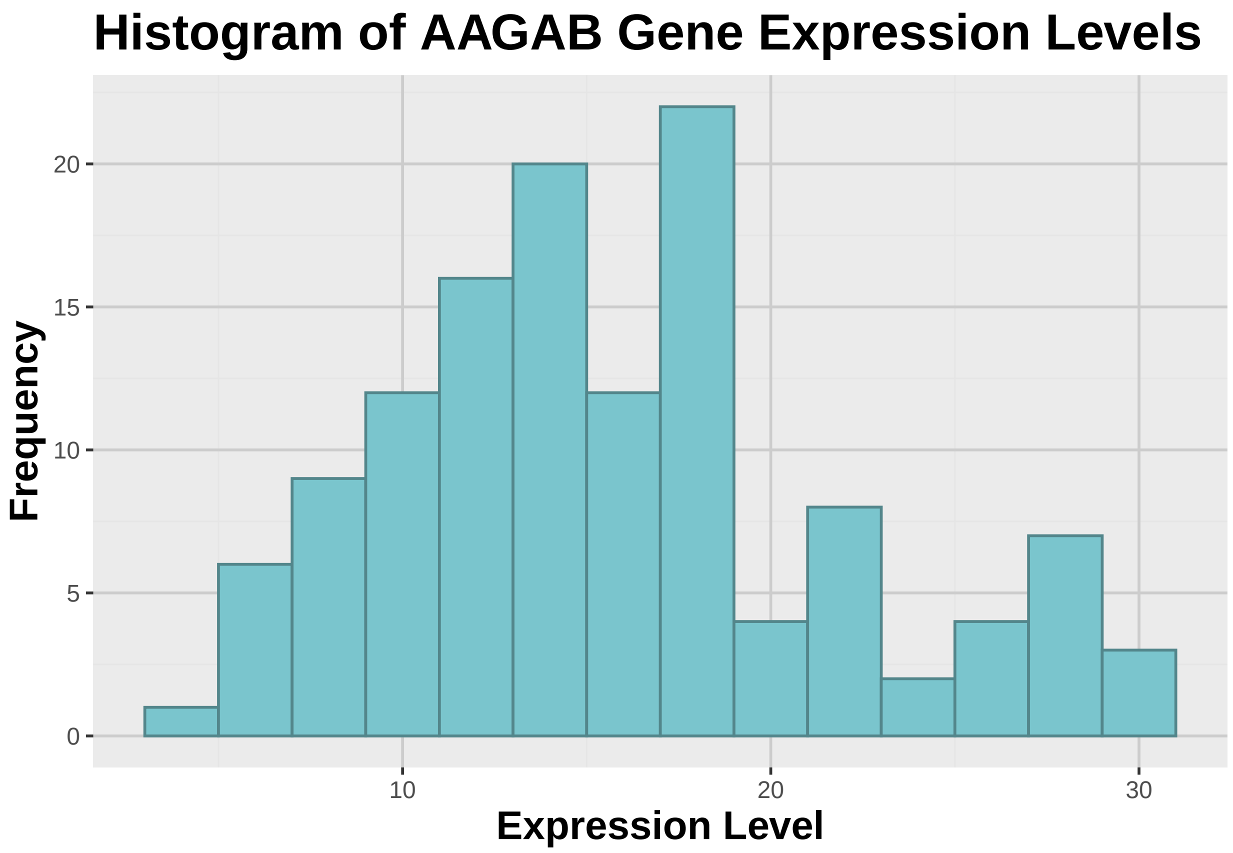
```r
library(tidyverse)
```

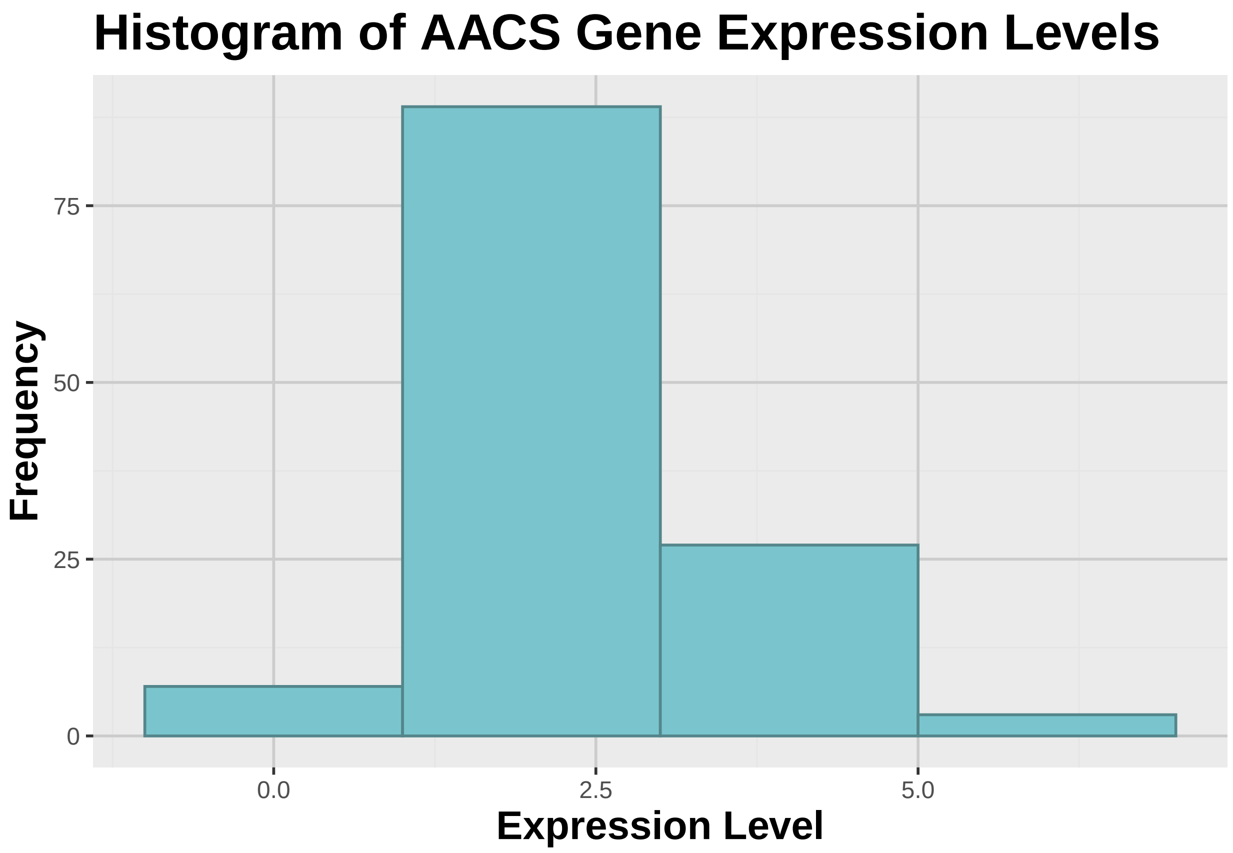
```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.5
## v forcats   1.0.0     v stringr   1.5.1
## v ggplot2   3.5.1     v tibble    3.2.1
## v lubridate 1.9.3     v tidyr     1.3.1
## v purrr     1.0.2
## -- Conflicts ------------------------------------------ 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 error
```

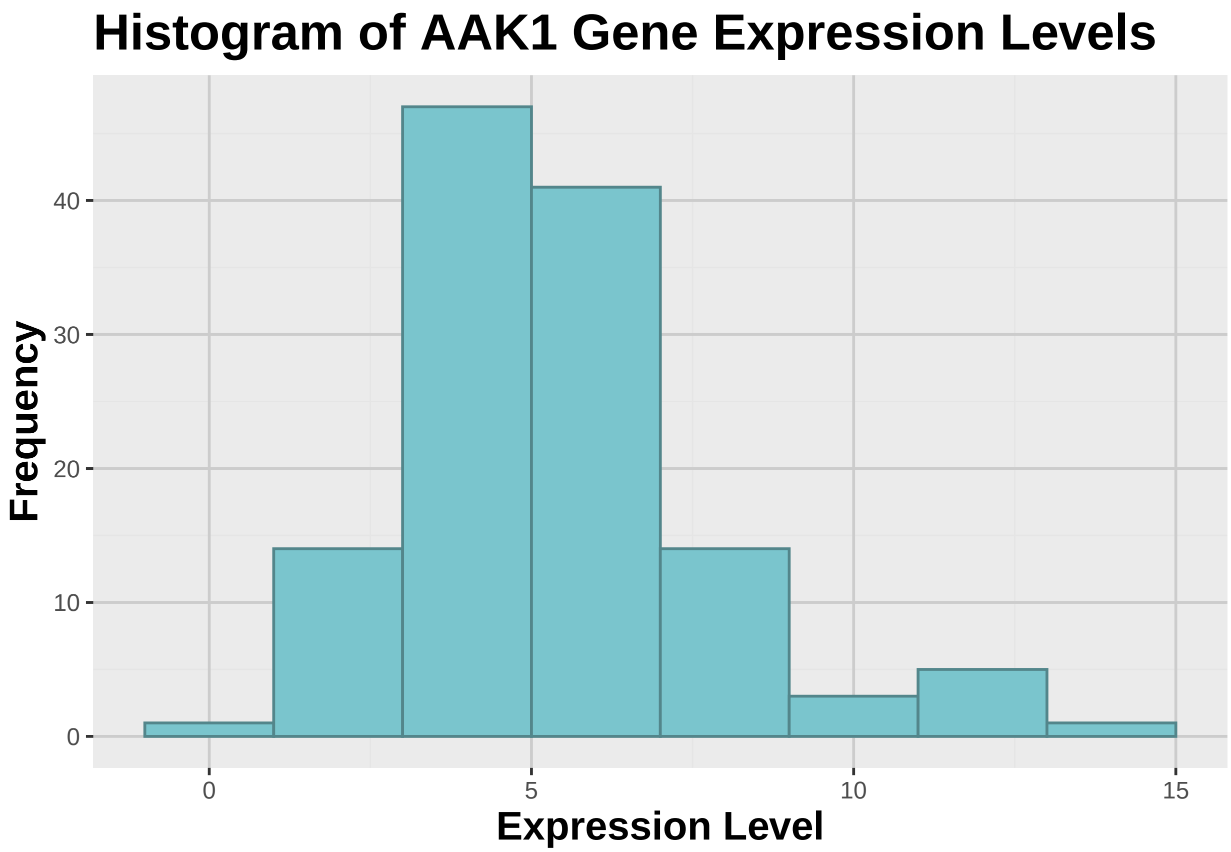
```r
library(ggplot2)
library(stringr)
```

Histogram

```r
histogram <- function(df, genes, cont_cov, cate_cov1, cate_cov2) {
  genes_string <- deparse(substitute(genes)) # turns the genes input into a string
  genes_string <- substr(genes_string, 6, nchar(genes_string)) # gets rid of "list("
  genes_string <- substr(genes_string, 1, nchar(genes_string) - 1) # gets rid of ")"
  genes_string <- str_remove_all(genes_string, ",") # removes all ","
  genes_string <- str_split(genes_string, " ")[[1]] # splits each gene into a lst
  gene_names <- list() # blank list
  for (i in seq_along(genes_string)) {
    word <- substr(genes_string[i], 11, nchar(genes_string[i])) # gets rid of "gene_data$"
    gene_names[i] <- word # adds the gene name to the blank list
  }
  tracker <- 0
  for (gene in genes) {
    tracker <- tracker + 1
    hist <- ggplot(gene_data, aes(x = as.numeric(gene))) +
  geom_histogram(binwidth = 2, fill = "cadetblue3", color = "cadetblue4") +
  labs( # labeling the title and axis
    title = paste0("Histogram of ", gene_names[tracker], " Gene Expression Levels"),
    x = "Expression Level",
    y = "Frequency"
  ) +
  theme(
    plot.title = element_text(size = 19, face = "bold"), # title
    axis.title = element_text(size = 15, face = "bold"), # axis
    panel.grid.major = element_line(color = "grey80"), # background
    panel.grid.minor = element_line(color = "grey90")
    )
    print(hist)
  }
}

#histogram(NULL, gene_data$AAGAB, NULL, NULL, NULL)
histogram(NULL, list(gene_data$AAGAB, gene_data$AACS, gene_data$AAK1), NULL, NULL, NULL)
```


Histogram of AAGAB Gene Expression Levels
Frequency
20
15
10
5
0
10
20
30
Expression Level


Histogram of AACS Gene Expression Levels
Frequency
75
50
25
0
0.0
2.5
5.0
Expression Level

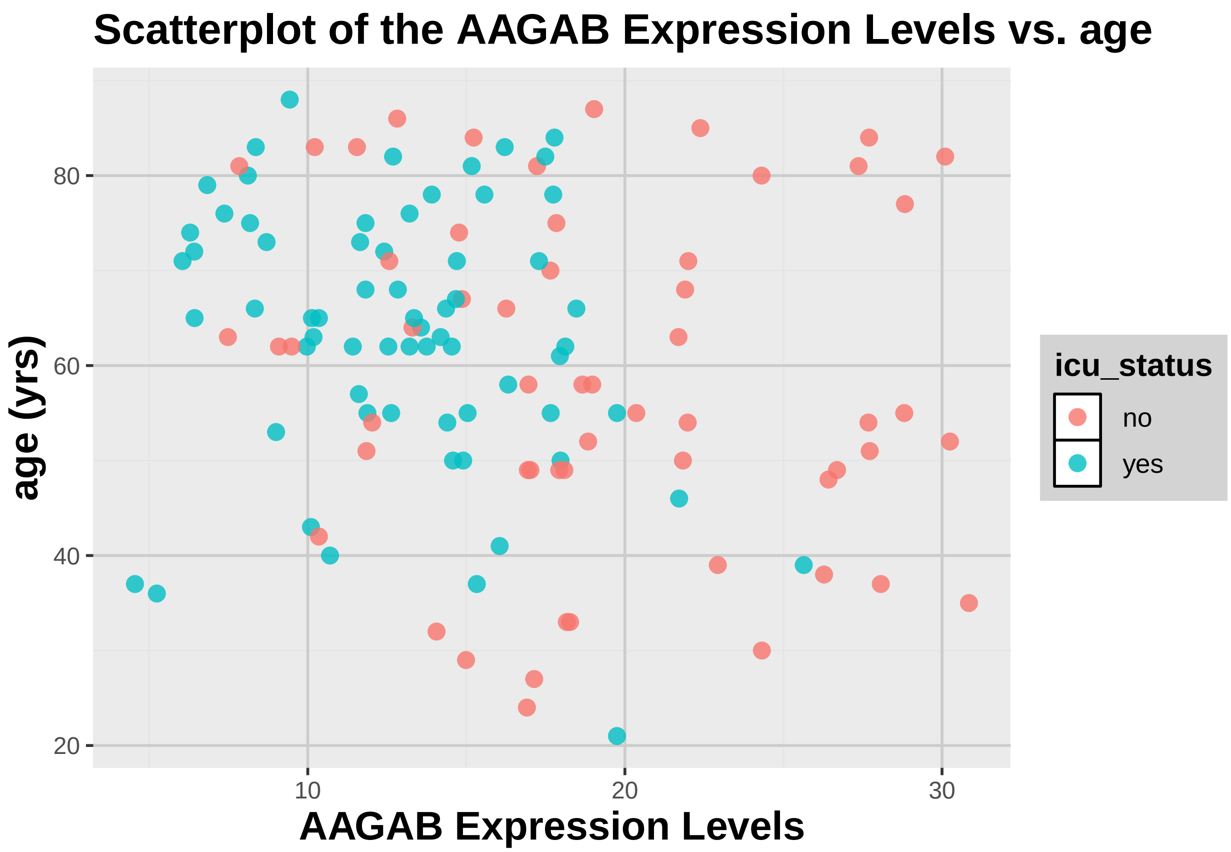


Scatterplot

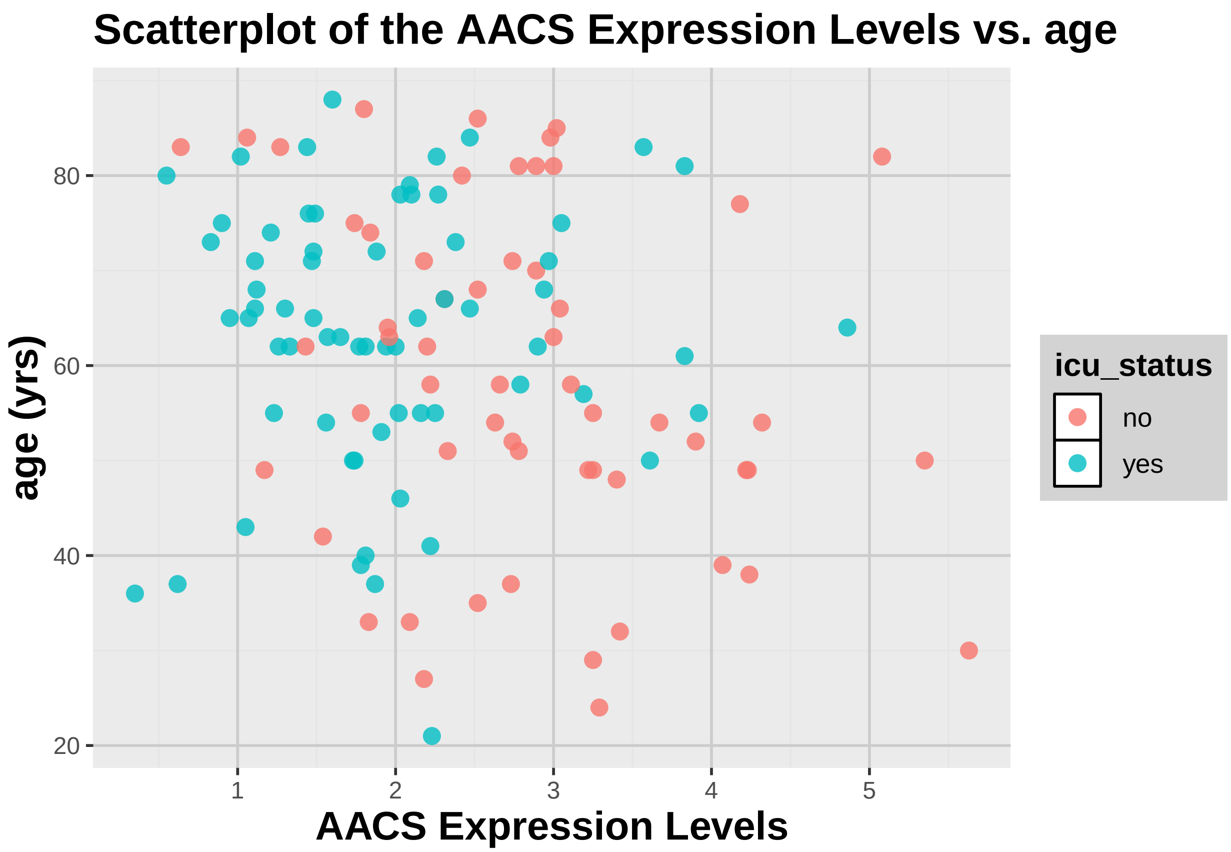
```r
scatterplot <- function(df, genes, cont_cov, cate_cov1, cate_cov2) {
  # for labeling purposes
  cont_cov_string <- deparse(substitute(cont_cov)) # turns the cont_cov input into a string
  cont_cov_string <- substr(cont_cov_string, 13, nchar(cont_cov_string)) # gets rid of series_data$
  cate_cov1_string <- deparse(substitute(cate_cov1)) # turns the cate_cov1 input into a string
  cate_cov1_string <- substr(cate_cov1_string, 13, nchar(cate_cov1_string)) # gets rid of series_data$
  genes_string <- deparse(substitute(genes)) # turns the genes input into a string
  genes_string <- substr(genes_string, 6, nchar(genes_string)) # gets rid of "list("
  genes_string <- substr(genes_string, 1, nchar(genes_string) - 1) # gets rid of ")"
  genes_string <- str_remove_all(genes_string, ",") # removes all ","
  genes_string <- str_split(genes_string, " ")[[1]] # splits each gene into a lst
  gene_names <- list() # blank list
  for (i in seq_along(genes_string)) {
    word <- substr(genes_string[i], 11, nchar(genes_string[i])) # gets rid of "gene_data$"
    gene_names[i] <- word # adds the gene name to the blank list
  }
  tracker <- 0

  for (gene in genes) {
    tracker <- tracker + 1
    scatter <- ggplot(df, aes(x = as.numeric(gene), y = as.numeric(cont_cov),
                        color = factor(cate_cov1))) + # separating the points by icu status
  geom_point(size = 2.5, alpha = 0.8) + # size and transparency of the point
  labs( # labeling the title and axis
    title = paste0("Scatterplot of the ", gene_names[tracker],
                   " Expression Levels vs. ", cont_cov_string),
    x = paste0(gene_names[tracker]," Expression Levels"),
    y = paste0(cont_cov_string," (yrs)"),
    color = cate_cov1_string # legend title
    ) +
  theme(
    plot.title = element_text(size = 16, face = "bold"),
    axis.title = element_text(size = 15, face = "bold"),
    legend.title = element_text(size = 12, face = "bold"),  # editing the legend
    legend.text = element_text(size = 10),
    legend.background = element_rect(fill = "lightgray", color = NA),
    legend.key = element_rect(fill = "white", color = "black"),
    panel.grid.major = element_line(color = "grey80"), # background
    panel.grid.minor = element_line(color = "grey90")
  )
    print(scatter)
  }
}

scatterplot(series_data, list(gene_data$AAGAB, gene_data$AACS, gene_data$AAK1),
            series_data$age, series_data$icu_status, series_data$sex)
```

```
## Warning in FUN(X[[i]], ...): NAs introduced by coercion
```

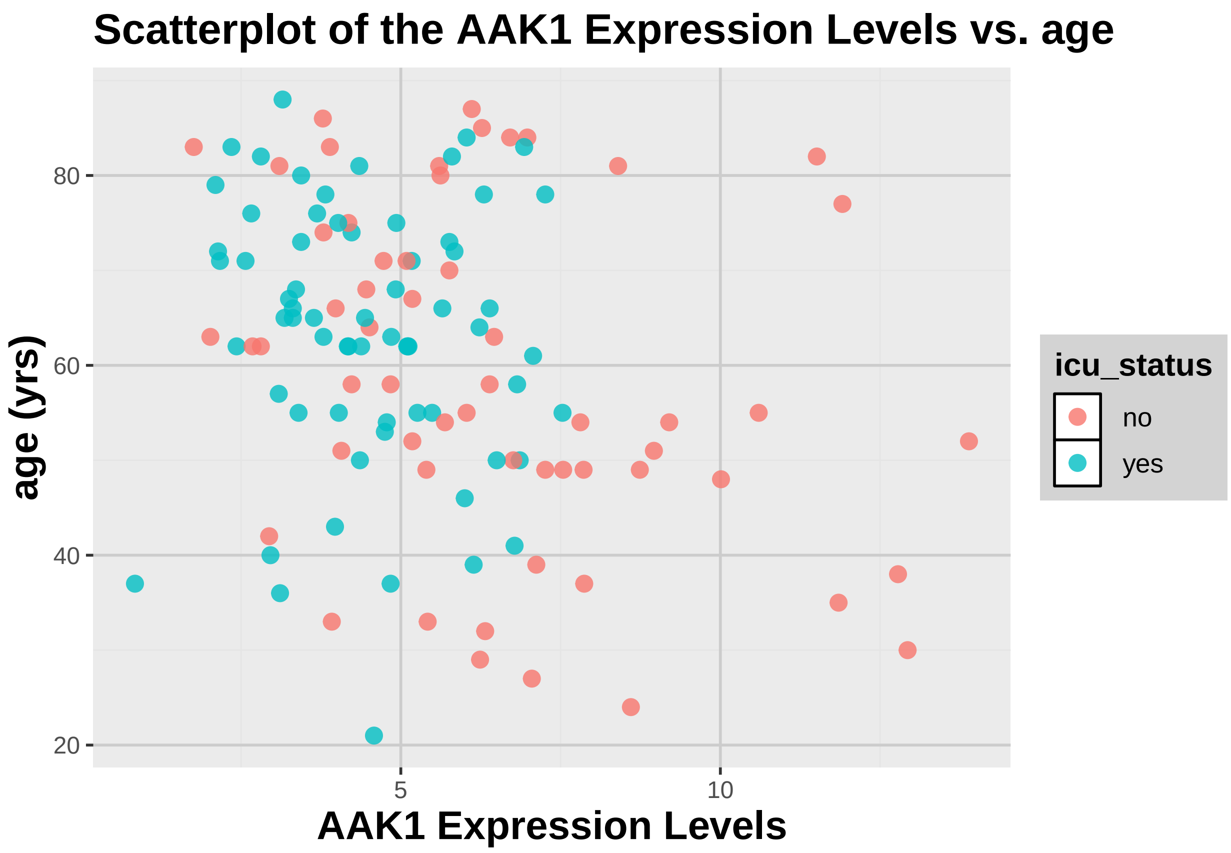
```
## Warning: Removed 3 rows containing missing values or values outside the scale range
## (‘geom_point()‘).
```



```
## Warning in FUN(X[[i]], ...): NAs introduced by coercion
## Warning in FUN(X[[i]], ...): Removed 3 rows containing missing values or values outside the scale ran
## (‘geom_point()‘).
```

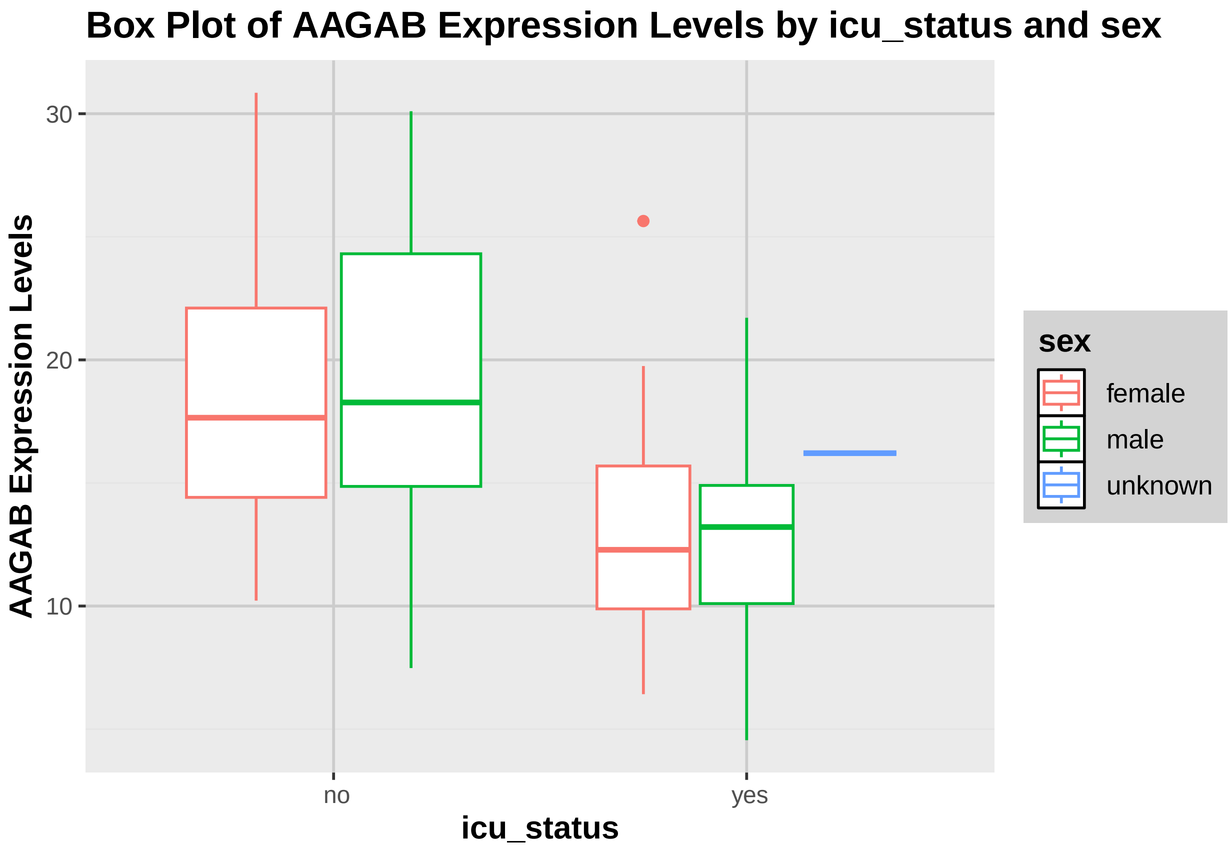


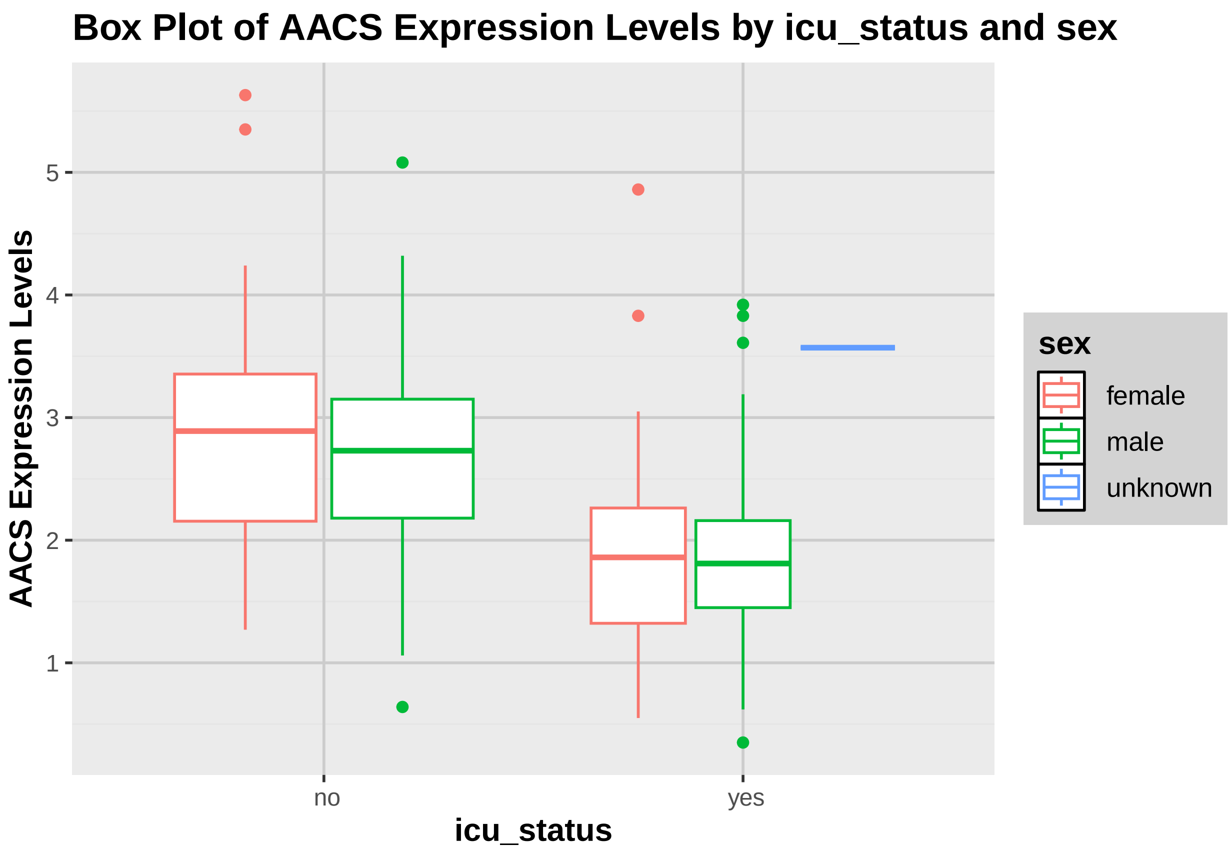
```
## Warning in FUN(X[[i]], ...): NAs introduced by coercion
## Warning in FUN(X[[i]], ...): Removed 3 rows containing missing values or values outside the scale ran
## (‘geom_point()‘).
```

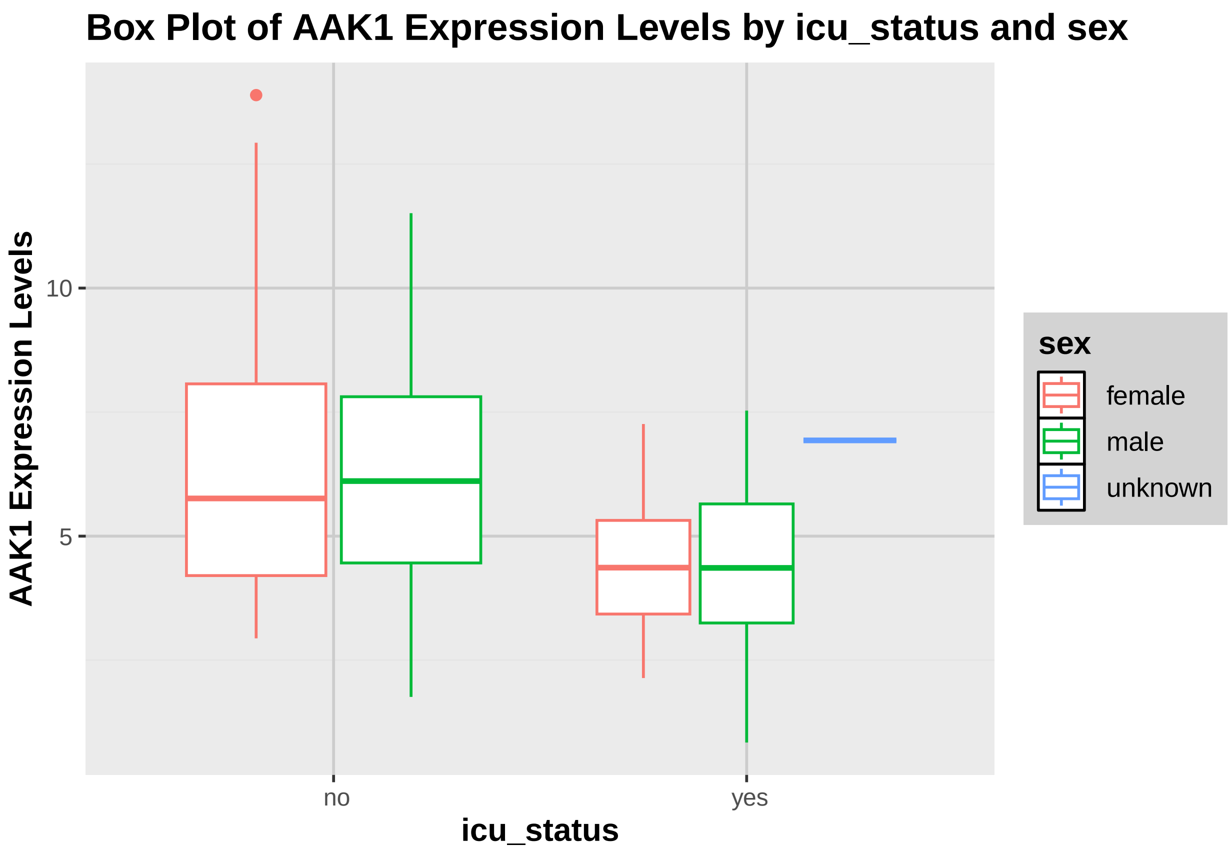


Box Plot

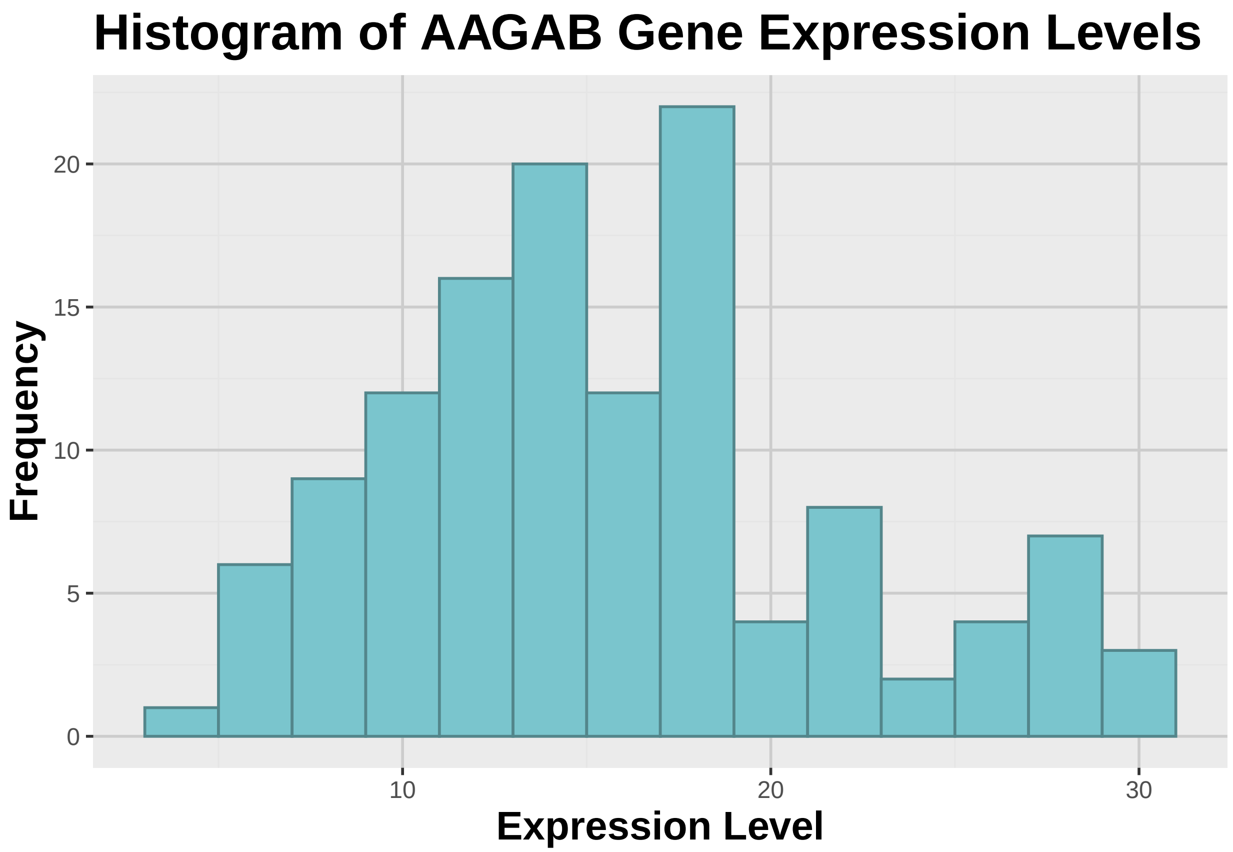
```
box_plot <- function(df, genes, cont_cov, cate_cov1, cate_cov2) {
  # for labeling purposes
  cate_cov1_string <- deparse(substitute(cate_cov1)) # turns the cate_cov1 input into a string
  cate_cov1_string <- substr(cate_cov1_string, 13, nchar(cate_cov1_string)) # gets rid of series_data$
  cate_cov2_string <- deparse(substitute(cate_cov2)) # turns the cate_cov2 input into a string
  cate_cov2_string <- substr(cate_cov2_string, 13, nchar(cate_cov2_string)) # gets rid of series_data$
  genes_string <- deparse(substitute(genes)) # turns the genes input into a string
  genes_string <- substr(genes_string, 6, nchar(genes_string)) # gets rid of "list("
  genes_string <- substr(genes_string, 1, nchar(genes_string) - 1) # gets rid of ")"
  genes_string <- str_remove_all(genes_string, ",") # removes all ","
  genes_string <- str_split(genes_string, " ")[[1]] # splits each gene into a lst
  gene_names <- list() # blank list
  for (i in seq_along(genes_string)) {
    word <- substr(genes_string[i], 11, nchar(genes_string[i])) # gets rid of "gene_data$"
    gene_names[i] <- word # adds the gene name to the blank list
  }
  tracker <- 0

  for (gene in genes) {
    tracker <- tracker + 1
    bp <- ggplot(df,aes(x = cate_cov1, y = as.numeric(gene), color = cate_cov2)) +
  geom_boxplot() +
  labs( # labeling the title and axis
    title = paste0("Box Plot of ", gene_names[tracker],
          " Expression Levels by ", cate_cov1_string, " and ", cate_cov2_string),
    x = cate_cov1_string,
    y = paste0(gene_names[tracker], " Expression Levels"),
    color = cate_cov2_string # label title
  ) +
  theme(
    plot.title = element_text(size = 14, face = "bold"),
    axis.title = element_text(size = 12, face = "bold"),
    legend.title = element_text(size = 12, face = "bold"), # editing the legend
    legend.text = element_text(size = 10),
    legend.background = element_rect(fill = "lightgray", color = NA),
    legend.key = element_rect(fill = "white", color = "black"),
    panel.grid.major = element_line(color = "grey80"), # background
    panel.grid.minor = element_line(color = "grey90")
  )
  print(bp)
  }
}

box_plot(series_data, list(gene_data$AAGAB, gene_data$AACS, gene_data$AAK1), NULL,
         series_data$icu_status, series_data$sex)
```



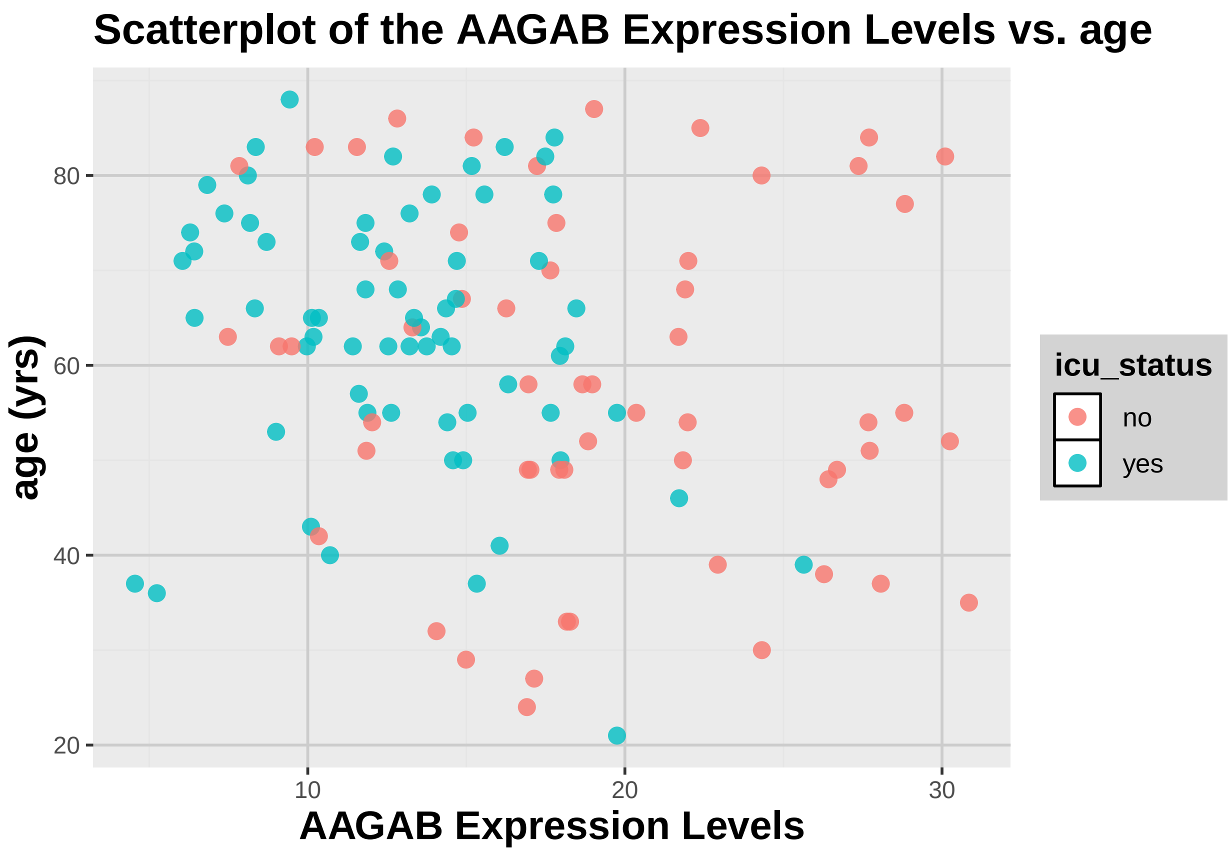

Box Plot of AACS Expression Levels by icu_status and sex
AACS Expression Levels
5
4
3
2
1
no
yes
icu_status
sex
female
male
unknown

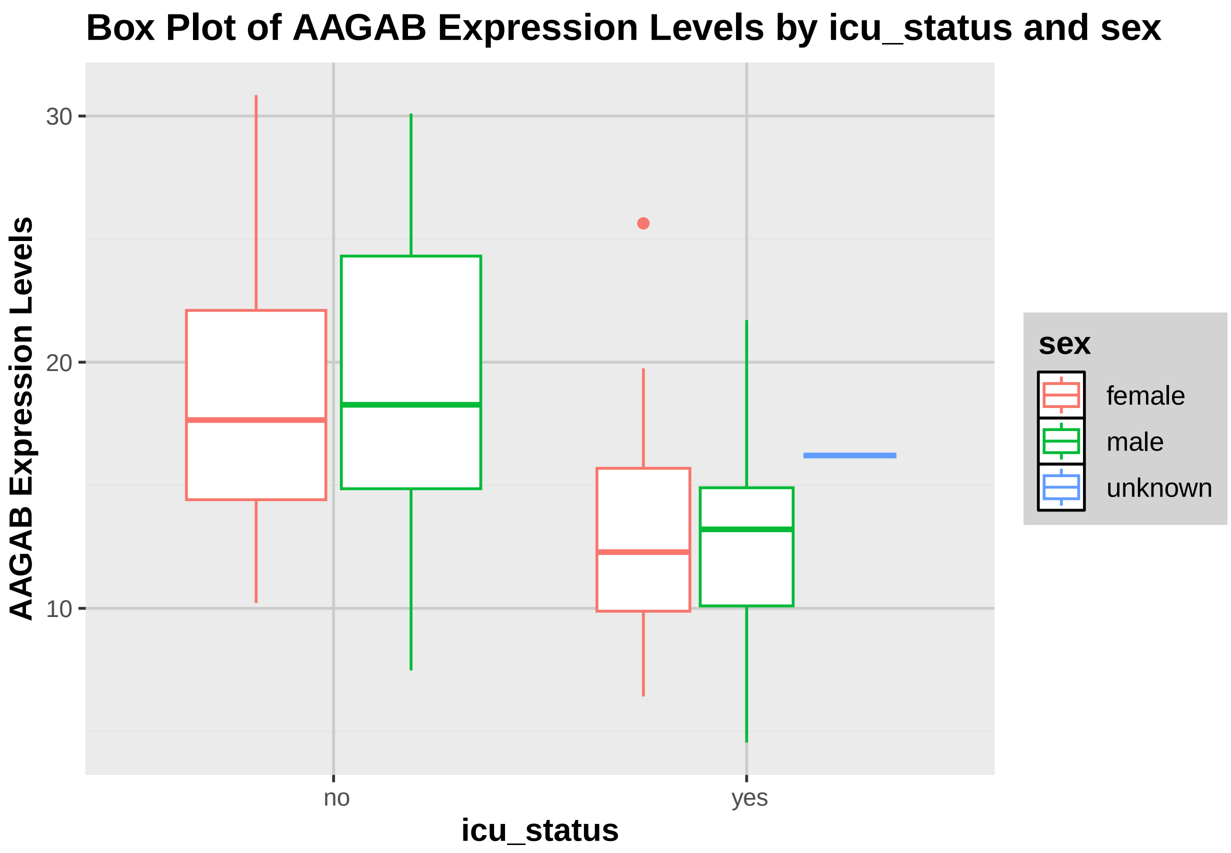


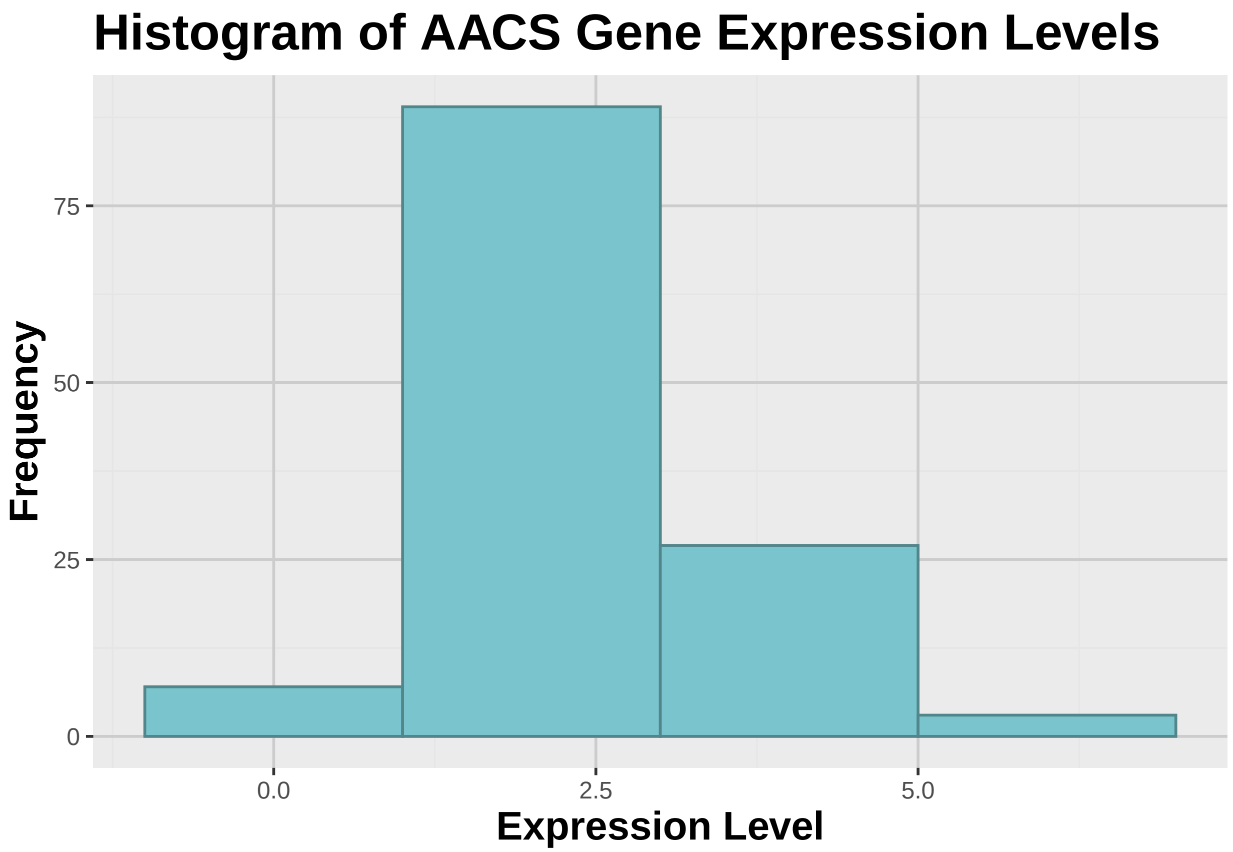
```r
# combined functions
graphing_functions <- function(df, genes, cont_cov, cate_cov1, cate_cov2) {
  # for labeling purposes
  cont_cov_string <- deparse(substitute(cont_cov)) # turns the cont_cov input into a string
  cont_cov_string <- substr(cont_cov_string, 13, nchar(cont_cov_string)) # gets rid of series_data$
  cate_cov1_string <- deparse(substitute(cate_cov1)) # turns the cate_cov1 input into a string
  cate_cov1_string <- substr(cate_cov1_string, 13, nchar(cate_cov1_string)) # gets rid of series_data$
  cate_cov2_string <- deparse(substitute(cate_cov2)) # turns the cate_cov2 input into a string
  cate_cov2_string <- substr(cate_cov2_string, 13, nchar(cate_cov2_string)) # gets rid of series_data$
  genes_string <- deparse(substitute(genes)) # turns the genes input into a string
  genes_string <- substr(genes_string, 6, nchar(genes_string)) # gets rid of "list("
  genes_string <- substr(genes_string, 1, nchar(genes_string) - 1) # gets rid of ")"
  genes_string <- str_remove_all(genes_string, ",") # removes all ","
  genes_string <- str_split(genes_string, " ")[[1]] # splits each gene into a lst
  gene_names <- list() # blank list
  for (i in seq_along(genes_string)) {
    word <- substr(genes_string[i], 11, nchar(genes_string[i])) # gets rid of "gene_data$"
    gene_names[i] <- word # adds the gene name to the blank list
  }
  tracker <- 0

  for (gene in genes) {
    tracker <- tracker + 1
    # histogram
    hist <- ggplot(gene_data, aes(x = as.numeric(gene))) +
  geom_histogram(binwidth = 2, fill = "cadetblue3", color = "cadetblue4") +
labs( # labeling the title and axis
  title = paste0("Histogram of ", gene_names[tracker], " Gene Expression Levels"),
  x = "Expression Level",
  y = "Frequency"
) +
theme(
  plot.title = element_text(size = 19, face = "bold"), # title
  axis.title = element_text(size = 15, face = "bold"), # axis
  panel.grid.major = element_line(color = "grey80"), # background
  panel.grid.minor = element_line(color = "grey90")
  )
  print(hist)

  # scatterplot
  scatter <- ggplot(df, aes(x = as.numeric(gene), y = as.numeric(cont_cov),
                      color = factor(cate_cov1))) + # separating the points by icu status
geom_point(size = 2.5, alpha = 0.8) + # size and transparency of the point
labs( # labeling the title and axis
  title = paste0("Scatterplot of the ", gene_names[tracker]," Expression Levels vs. ",
                 cont_cov_string),
  x = paste0(gene_names[tracker]," Expression Levels"),
  y = paste0(cont_cov_string," (yrs)"),
  color = cate_cov1_string # legend title
  ) +
theme(
  plot.title = element_text(size = 16, face = "bold"),
  axis.title = element_text(size = 15, face = "bold"),
  legend.title = element_text(size = 12, face = "bold"),  # editing the legend
  legend.text = element_text(size = 10),
  legend.background = element_rect(fill = "lightgray", color = NA),
  legend.key = element_rect(fill = "white", color = "black"),
  panel.grid.major = element_line(color = "grey80"), # background
  panel.grid.minor = element_line(color = "grey90")
)
  print(scatter)

  # box plot
  bp <- ggplot(df,aes(x = cate_cov1, y = as.numeric(gene), color = cate_cov2)) +
geom_boxplot() +
labs( # labeling the title and axis
  title = paste0("Box Plot of ", gene_names[tracker]," Expression Levels by ",
                 cate_cov1_string, " and ", cate_cov2_string),
  x = cate_cov1_string,
  y = paste0(gene_names[tracker], " Expression Levels"),
  color = cate_cov2_string # label title
) +
theme(
  plot.title = element_text(size = 14, face = "bold"),
  axis.title = element_text(size = 12, face = "bold"),
  legend.title = element_text(size = 12, face = "bold"), # editing the legend
  legend.text = element_text(size = 10),
  legend.background = element_rect(fill = "lightgray", color = NA),
  legend.key = element_rect(fill = "white", color = "black"),
    panel.grid.major = element_line(color = "grey80"), # background
    panel.grid.minor = element_line(color = "grey90")
  )
  print(bp)
  }
}
graphing_functions(series_data, list(gene_data$AAGAB, gene_data$AACS, gene_data$AAK1),
                   series_data$age, series_data$icu_status, series_data$sex)
```



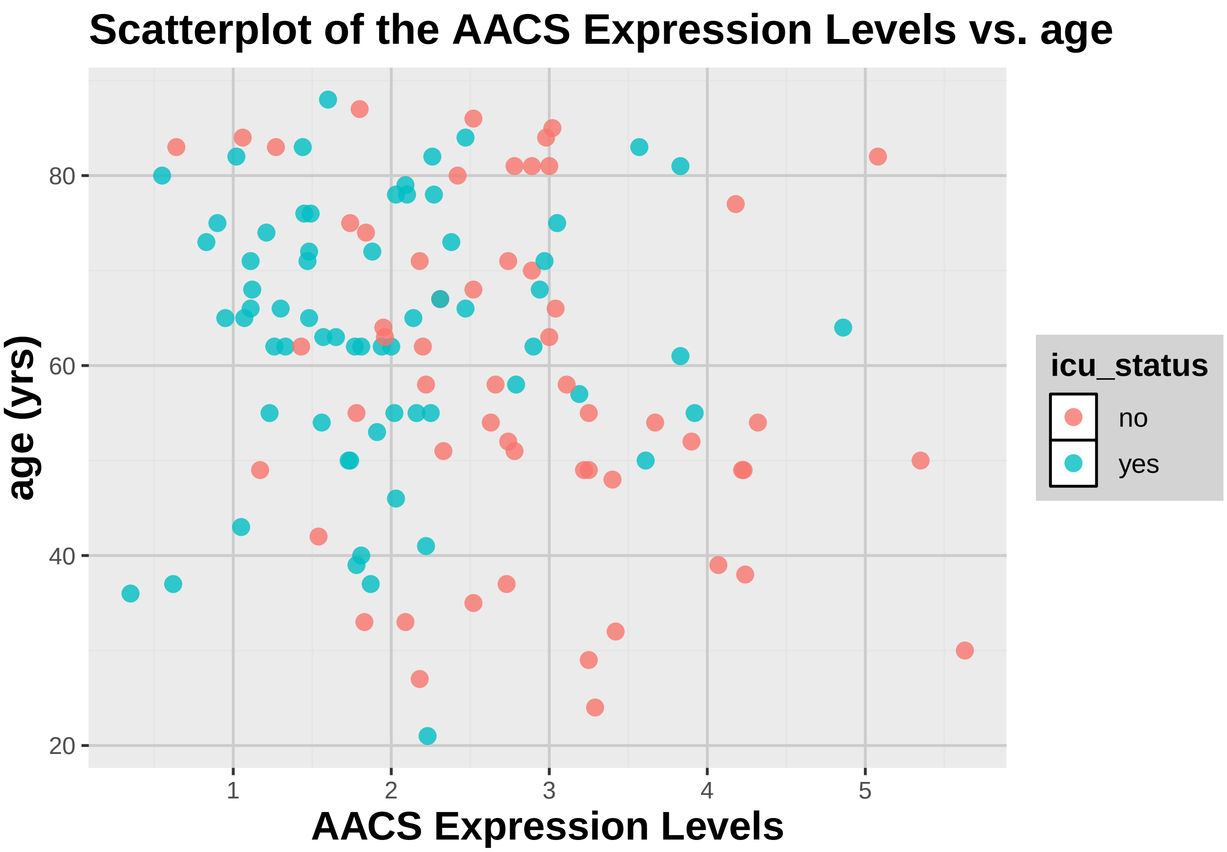
```
## Warning in FUN(X[[i]], ...): NAs introduced by coercion

## Warning: Removed 3 rows containing missing values or values outside the scale range
## (‘geom_point()‘).
```

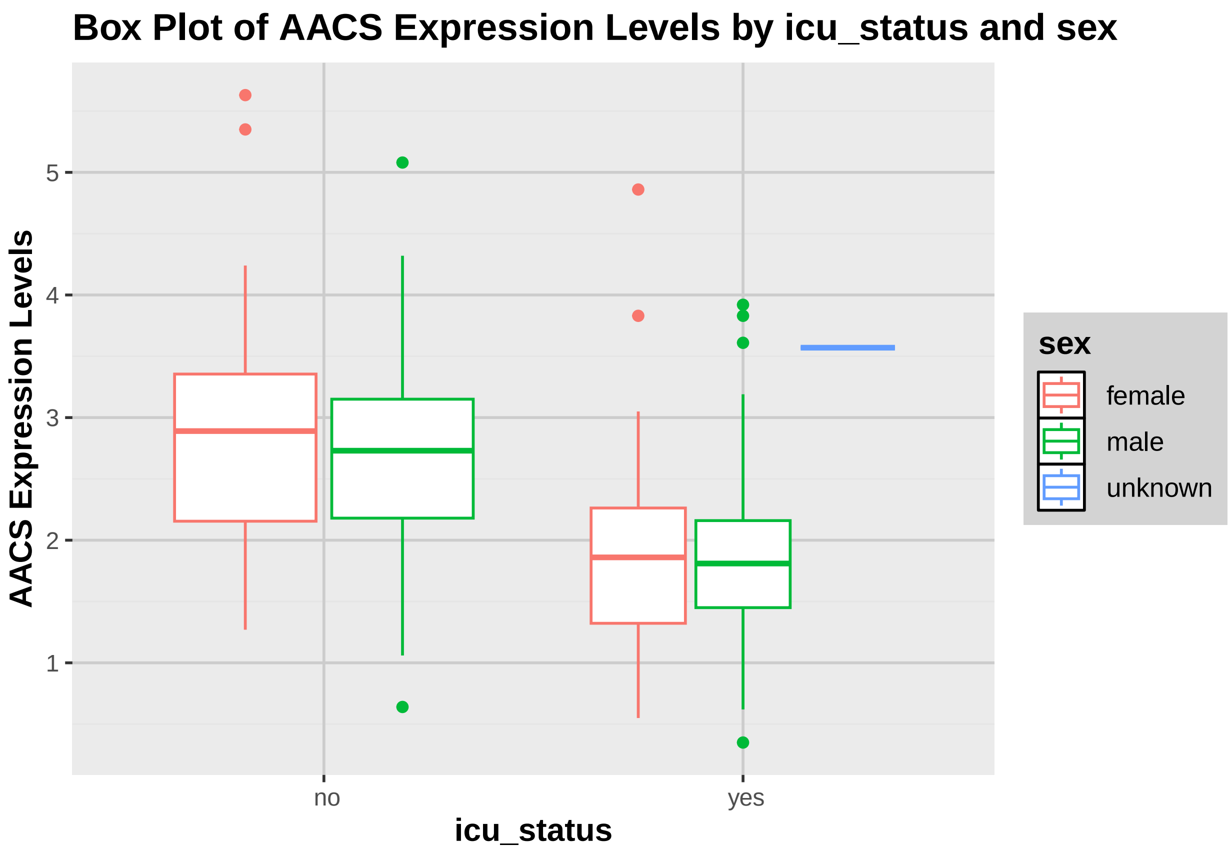

Scatterplot of the AAGAB Expression Levels vs. age
age (yrs)
80
60
40
20
10
20
30
AAGAB Expression Levels
icu_status
no
yes

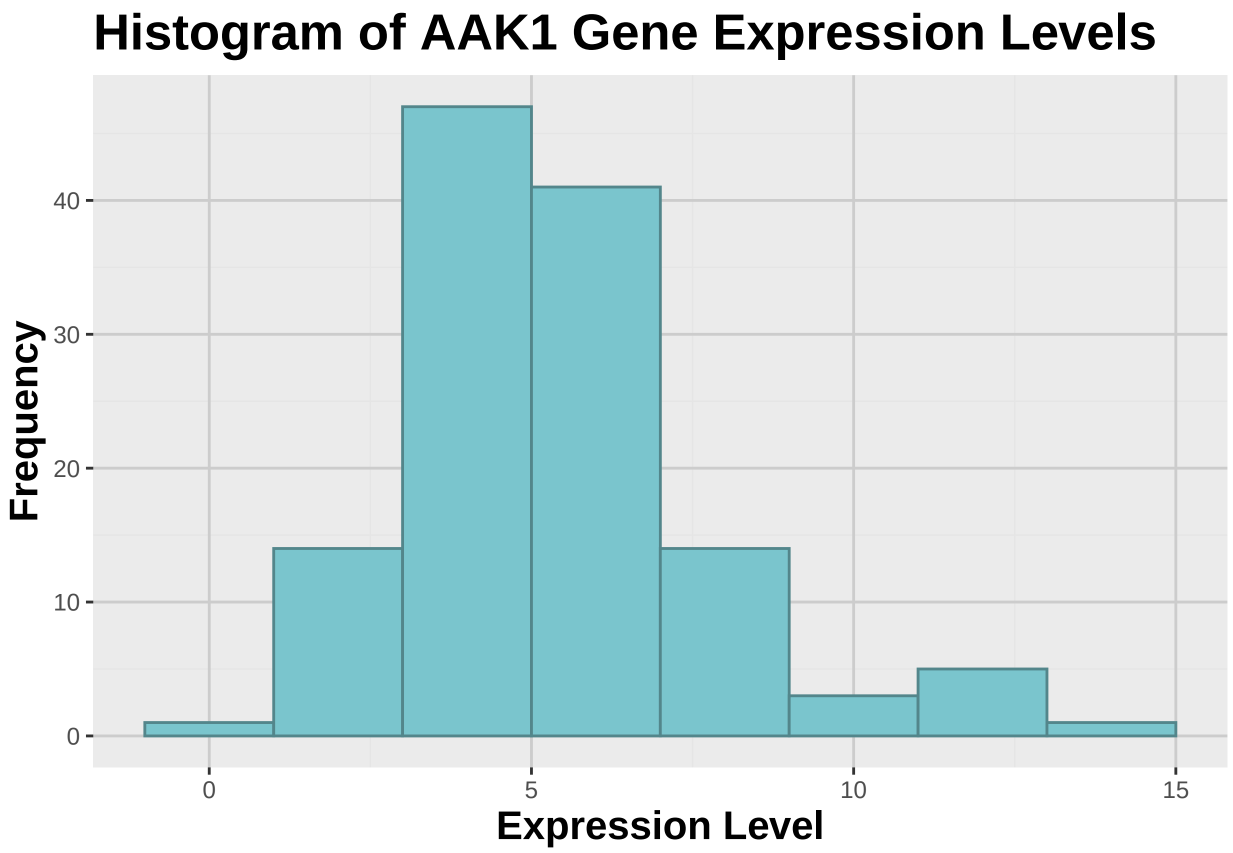

Box Plot of AAGAB Expression Levels by icu_status and sex
AAGAB Expression Levels
30
20
10
no
yes
icu_status
sex
female
male
unknown



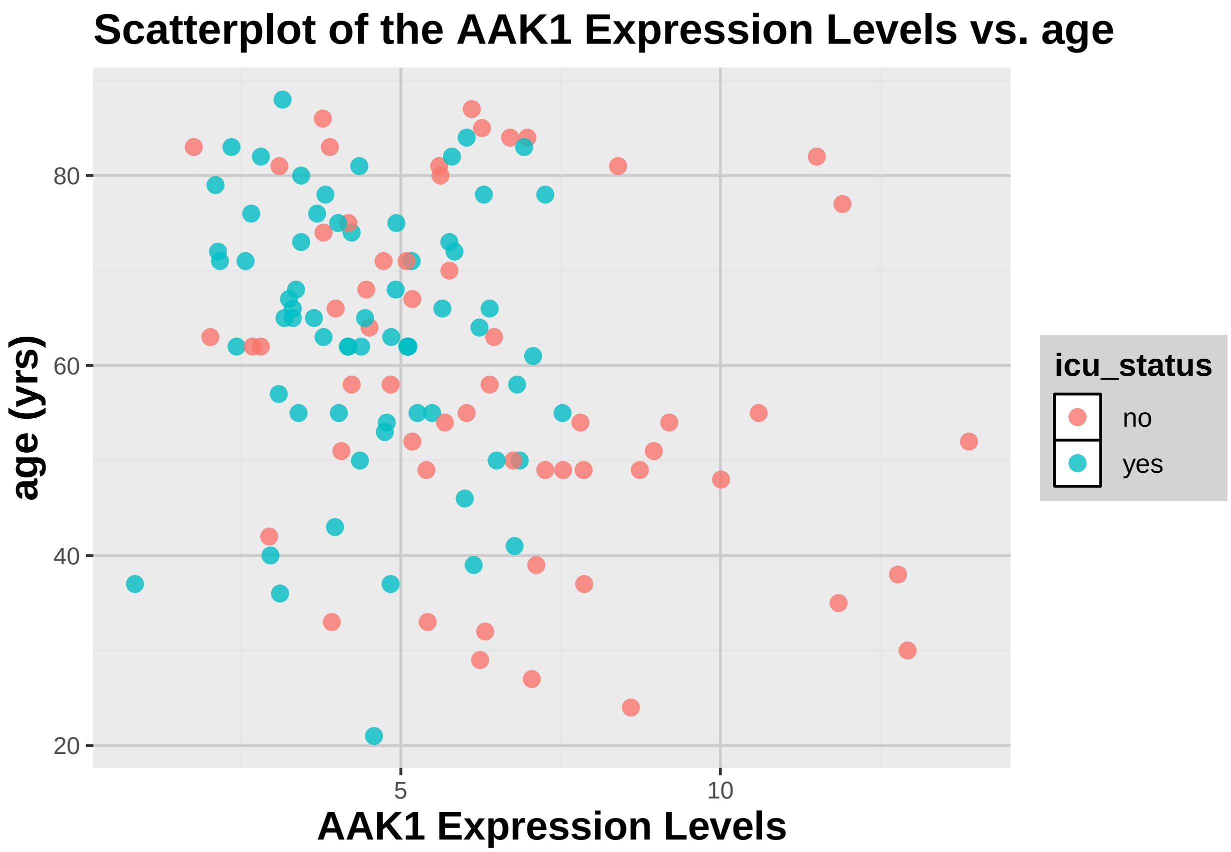
```
## Warning in FUN(X[[i]], ...): NAs introduced by coercion
## Warning in FUN(X[[i]], ...): Removed 3 rows containing missing values or values outside the scale ran
## (‘geom_point()‘).
```

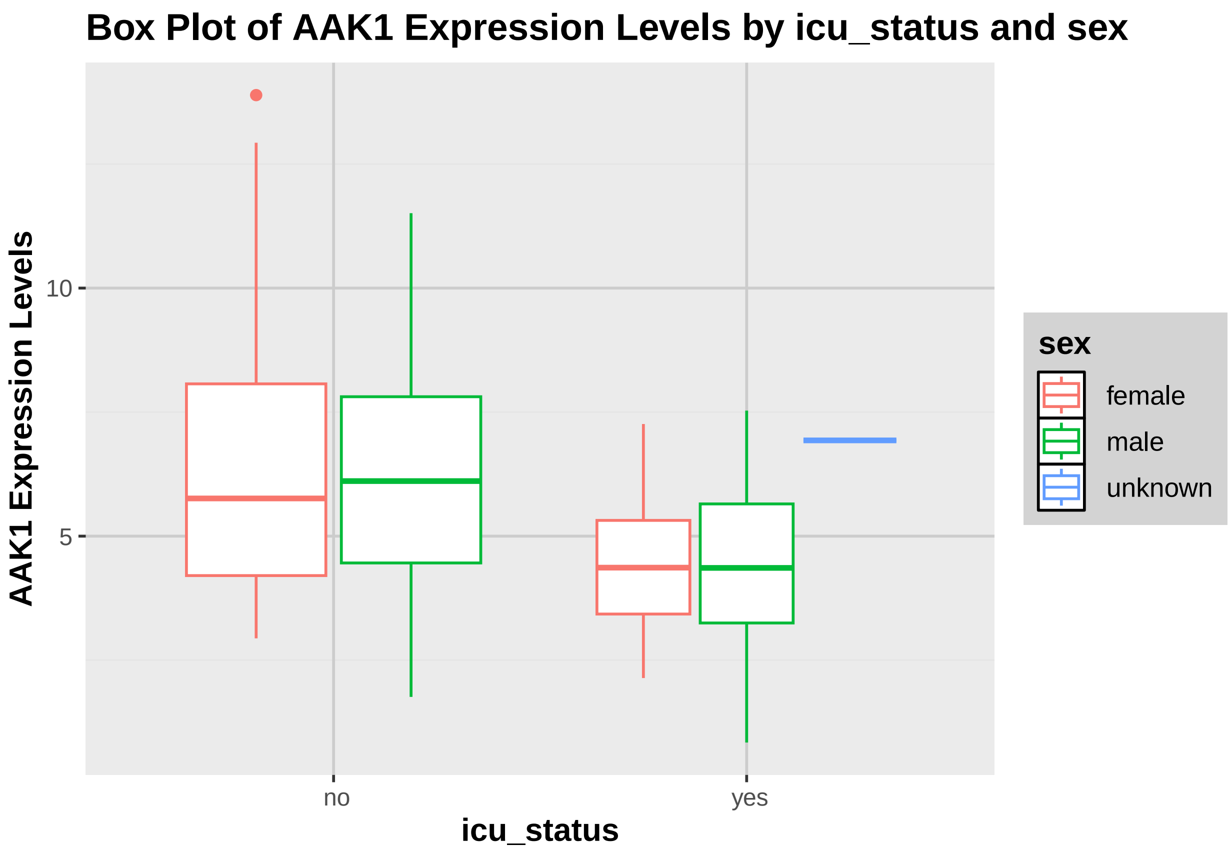

Scatterplot of the AACS Expression Levels vs. age
age (yrs)
80
60
40
20
1
2
3
4
5
AACS Expression Levels
icu_status
no
yes


Box Plot of AACS Expression Levels by icu_status and sex
AACS Expression Levels
5
4
3
2
1
no
yes
icu_status
sex
female
male
unknown



```
## Warning in FUN(X[[i]], ...): NAs introduced by coercion
## Warning in FUN(X[[i]], ...): Removed 3 rows containing missing values or values outside the scale ran
## (‘geom_point()‘).
```


Scatterplot of the AAK1 Expression Levels vs. age
age (yrs)
80
60
40
20
5
10
AAK1 Expression Levels
icu_status
● no
● yes


Box Plot of AAK1 Expression Levels by icu_status and sex
AAK1 Expression Levels
10
5
no
yes
icu_status
sex
female
male
unknown