QBS103 Project - Elodie Richard

2024-07-25

```
project1.data <- list.files(path = "/Users/elodierichard/Documents/QBS103/Project Submission 1 Data", pa
print(project1.data) #I first moved both data files into one folder on my laptop to retrieve it
```

```
## [1] "QBS103_GSE157103_genes.csv"         "QBS103_GSE157103_series_matrix.csv"
```

```
setwd("/Users/elodierichard/Documents/QBS103/Project Submission 1 Data") #this is to set the working di

genes <- read.csv("QBS103_GSE157103_genes.csv") #this is to rename and retrieve the first gene data fil
the_matrix <- read.csv("QBS103_GSE157103_series_matrix.csv") #this is to rename and retrieve the second

#head(genes) #this is used to visualize the data and only for 6 rows
#head(the_matrix)
```

```
#creating a genes data frame for the genes file
test_genes <- as.data.frame(t(genes))
names(test_genes) <- test_genes[1,] #this allows the genes table to be organized according to names by
test_genes <- test_genes[-1,] #this removes the first row containing "x" in the genes file so that it c
```

```
test_genes$participant_id <- row.names(test_genes) #this will move the participant id into it's own col
combined <- merge(test_genes, the_matrix, by = 'participant_id') #this combines the genes file and matr
```

```
#citation(package = 'tidyverse')
```

```
#Histogram using the gene AAAS
library(ggplot2)
setwd("/Users/elodierichard/Documents/QBS103/Project Submission 1 Data")

combined$ABCA13 <- as.numeric(combined$ABCA13) #needed to make them all as.numeric for submission 2
combined$AACS <- as.numeric(combined$AACS)
combined$AAAS <- as.numeric(combined$AAAS) #this is so that the plot can pull just the gene AAAS from t

histogram <- ggplot(combined, aes(x=combined$AAAS)) + #this called on ggplot to use the file "combined"
  geom_histogram(bins = 20, color = 'navy', fill = 'lightblue') +  #this generated the histogram with t
  labs(title = 'Gene Expression of AAAS', #this labeled the title and the axis
       x= 'Gene: AAAS' ,
       y= 'Frequency of AAAS' )
#plot(histogram)
```

```
#Scatterplot of the gene expression of AAAS compared to ferritin levels
library(ggplot2)
combined$ferritin.ng.ml. <- as.numeric(combined$ferritin.ng.ml.) #used to pull out ferritin to plot
```

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

```
#comments mostly the same as for histogram except for a few changes
scatterplot <- ggplot(combined, aes(x= combined$ferritin.ng.ml., y = combined$AAAS)) + #need to specifi
  geom_point(bins = 10, color = 'violet') +  #use geom_point for a scatter plot to be generated
  labs(title = 'Gene Expression of AAAS vs. Ferritin Levels' ,
```

```
       x= 'Ferritin Levels (ng/mL)',
       y= 'Gene Expression of AAAS')
```

```
## Warning in geom_point(bins = 10, color = "violet"): Ignoring unknown
## parameters: `bins`
```

```
#for trendline it's geom_smooth
#plot(scatterplot)
```

```
#Scatterplot for Gene Expression vs Age (this was run to compare different data to see differences) not
library(ggplot2)

scatterplot_practice<- ggplot(combined, aes(x= combined$age, y = combined$AAAS)) +
  geom_point(bins = 10, color = 'green') +
  labs(title = 'Gene Expression of AAAS vs. Age' , x= 'Age of Participant (yrs)', y= 'Gene AAAS')
```

```
## Warning in geom_point(bins = 10, color = "green"): Ignoring unknown parameters:
## `bins`
```

```
#plot(scatterplot_practice)
```

```
#Boxplot comparing gene expression of AAAS related to ICU status depending on Age
library(ggplot2)
#similar process to histogram and scatterplot with a few adjustments
boxplot <- ggplot(combined, aes(x=icu_status, y = AAAS, fill = age)) + #need to add a fill to demonstra
  geom_boxplot(bins = 10, fill = 'maroon') + #to generate a box plot use geom_boxplot
  labs(title = 'Gene Expression of AAAS vs ICU Status and Participant Age',  #to label each attribute o
       x= 'ICU Status of Participant' ,
       y= 'Gene Expression of AAAS',
       fill= 'Age of Participant (yrs)')
```

```
## Warning in geom_boxplot(bins = 10, fill = "maroon"): Ignoring unknown
## parameters: `bins`
```

```
#plot(boxplot)
```

```
### this is the fixed boxplot from the previous submission so that it includes a categorical variable (
#Boxplot comparing gene expression of AAAS related to ICU status depending on Age
library(ggplot2)
#similar process to histogram and scatterplot with a few adjustments
boxplot <- ggplot(combined, aes(x=icu_status, y = AAAS, fill = mechanical_ventilation)) + #need to add
  geom_boxplot() + #to generate a box plot use geom_boxplot
  scale_fill_manual(values = c('pink','salmon')) +
  labs(title = 'Gene Expression of AAAS vs ICU Status and Mechanical Ventilation',  #to label each attr
       x= 'ICU Status of Participant' ,
       y= 'Gene Expression of AAAS',
       fill= 'Mechanical Ventilation')
#plot(boxplot)
```

Submission 2 Build a function to create the plots you made for Presentation 1, incorporating any feedback you received on your submission. Your functions should take the following input: (1) the name of the data frame, (2) a list of 1 or more gene names, (3) 1 continuous covariate, and (4) two categorical covariates (10 pts) Select 2 additional genes (for a total of 3 genes) to look at and implement a loop to generate your figures using the function you created (10 pts) Present one of your boxplots in class. Be prepared to explain the gene and covariates you chose and comment on the distribution as if you were presenting your research findings. No slides are required, just bring your plot. In class, be prepared to provide constructive feedback for your classmates (5 pts) Make sure you push your code to your git repository prior to class. As a reminder, we do

not need you to share your GitHub repository until the final submission. Pushing this submission to GitHub will be worth 5 pts on the final submission and you can earn 1 additional point on your final project grade if you push 1 extra time along the way (changes between pushes must be significant to earn the extra point).

```
## comment out things you don't need anymore

#sub_2_plots <- function(data, genes, cat1 , cat2, cont ) { #created a new function with each element d

 #   histogram2 <- ggplot(data, aes_string(x= genes)) + geom_histogram(bins = 20, color = 'navy', fill
 #   scatterplot2 <- ggplot(data, aes_string(x= cont, y = genes)) + geom_point(bins = 10, color = 'viol
 #  boxplot2 <- ggplot(data, aes_string(x= cat1, y = genes, fill = cat2)) + geom_boxplot() + scale_fill_

   # plot(histogram2)
   # plot(scatterplot2)
   # plot(boxplot2)
#}

#specific_genes = subset(combined, select = c("AAAS", "AACS", "ABCA13")) #this is to subset the data ta
### had to be sure to make these chosen genes as.numeric so that it would run through
#for (gene in colnames(specific_genes)) { #created a for loop to run through every gene chosen in the n
 # print(gene)
 # print(combined[,gene]) # this will grab each row and the gene columns specifically
 # sub_2_plots(combined, gene , cat1 = "icu_status", cat2 = "mechanical_ventilation", cont = "ferritin.

 # sub_2_plots(combined, "AAAS", cat1 = "icu_status", cat2 = "mechanical_ventilation", cont = "ferritin
```

FINAL SUBMISSION Generate a table formatted in LaTeX of summary statistics for all the covariates you looked at and 2 additional continuous (3 total) and 1 additional categorical variable (3 total). (5 pts) Stratifying by one of your categorical variables Tables should report n (%) for categorical variables Tables should report mean (sd) or median [IQR] for continuous variables

```
#install.packages("tableone") #install the tableone packages
library(tableone)

#citation(package = 'tableone') #to get the citation
```

```
#making these columns/variables numeric
combined$procalcitonin.ng.ml.. <- as.numeric(combined$procalcitonin.ng.ml..)
```

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

```
combined$lactate.mmol.l. <- as.numeric(combined$lactate.mmol.l.)
```

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

```
#creating a table with the data from 'combined' dataset and then pulling out the variables wanted for t
project_table <- CreateTableOne(data = combined, vars = c('ferritin.ng.ml.', 'procalcitonin.ng.ml..', 'l
#print(project_table)
#the nonnormal is meant for the variables that don't have a normal distribution
project_table1 <- print(project_table, showAllLevels = T, nonnormal = c('ferritin.ng.ml.','procalcitonin
```

```
##                                      Stratified by mechanical_ventilation
##                                       level     no
##   n                                                74
##   ferritin.ng.ml. (median [IQR])                411.00 [131.00, 968.00]
##   procalcitonin.ng.ml.. (median [IQR])            0.37 [0.14, 0.73]
##   lactate.mmol.l. (median [IQR])                  1.17 [0.87, 1.49]
```

```
##   icu_status (%)                          no          54 (73.0)
##                                           yes         20 (27.0)
##   sex (%)                                 female      35 (47.3)
##                                           male        38 (51.4)
##                                           unknown      1 ( 1.4)
##                                        Stratified by mechanical_ventilation
##                                           yes                    p      test
##   n                                          51
##   ferritin.ng.ml. (median [IQR])        697.00 [337.75, 1111.25]  0.057 nonnorm
##   procalcitonin.ng.ml.. (median [IQR])   1.27 [0.34, 2.78]      <0.001 nonnorm
##   lactate.mmol.l. (median [IQR])         1.30 [0.92, 1.65]       0.412 nonnorm
##   icu_status (%)                            5 ( 9.8)            <0.001
##                                            46 (90.2)
##   sex (%)                                  16 (31.4)             0.128
##                                            35 (68.6)
##                                             0 ( 0.0)
```

```
#making the file csv so it can be uploaded into overleaf
#write.csv(project_table1, '/Users/elodierichard/Documents/QBS103/Project_Table1.csv')
#table1(~ferritin.ng.ml. + procalcitonin.ng.ml.. + lactate.mmol.l. + icu_status + mechanical_ventilatio
```

#Generate final a publication quality histogram, scatter plot, and boxplot from submission 1 (i.e. only for your first gene of interest) (5 pts)

```
#citation(package = 'ggplot2') #inorder to cite ggplot
```

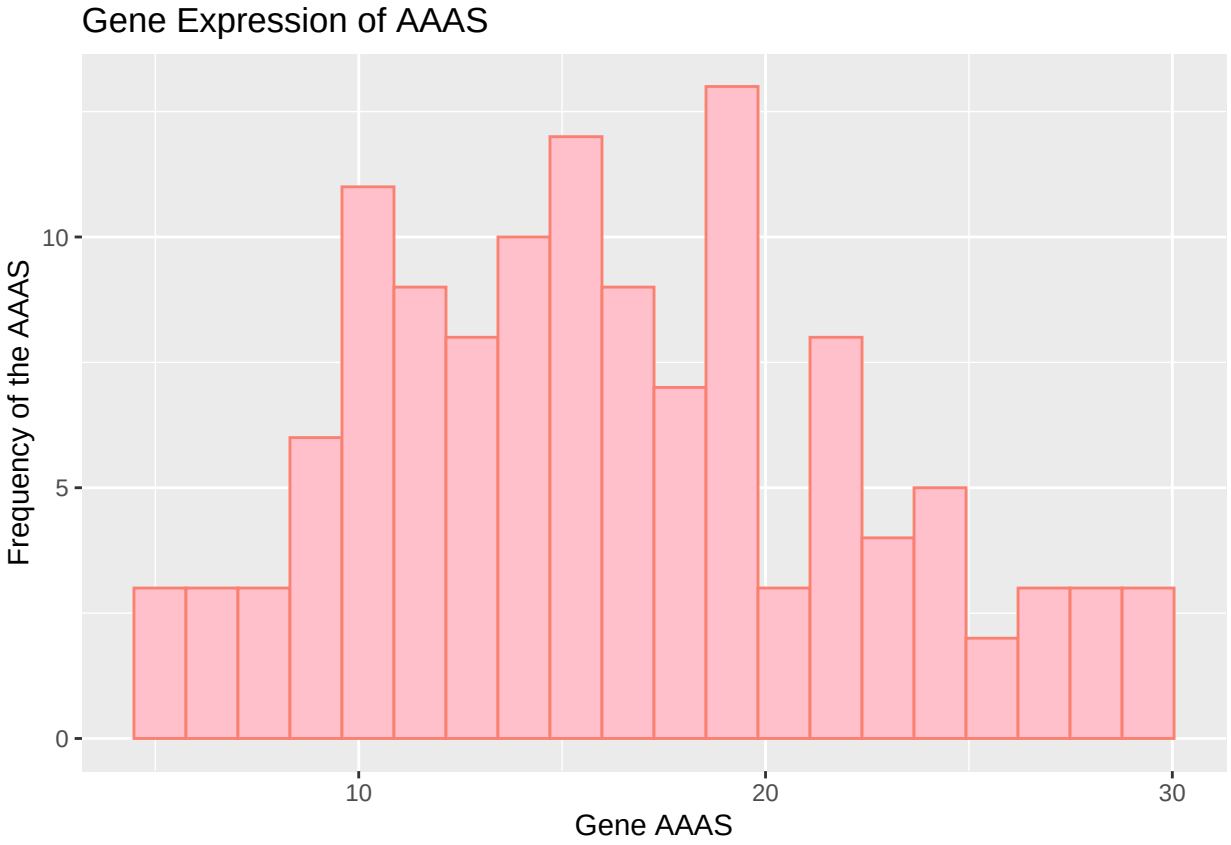
```
#library(ggplot2)
#setwd("/Users/elodierichard/Documents/QBS103/Project Submission 1 Data")

#used the function for loop from submission 2 but only used it for gene AAAS instead of 3 total genes
sub_2_plots <- function(data, genes, cat1 , cat2, cont ) { #created a new function with each element de

    histogram2 <- ggplot(data, aes_string(x= genes)) + geom_histogram(bins = 20, color = 'salmon', fill
    scatterplot2 <- ggplot(data, aes_string(x= cont, y = genes)) + geom_point(bins = 10, color = 'salmo
   boxplot2 <- ggplot(data, aes_string(x= cat1, y = genes, fill = cat2)) + geom_boxplot() + scale_fill_

    plot(histogram2)
    plot(scatterplot2)
    plot(boxplot2)
}

#this is just choosing one gene
specific_genes = subset(combined, select = c("AAAS")) #this is to subset the data table so that it only
### had to be sure to make these chosen genes as.numeric so that it would run through
for (gene in colnames(specific_genes)) { #created a for loop to run through every gene chosen in the ne
  print(gene)
  print(combined[,gene]) # this will grab each row and the gene columns specifically
  sub_2_plots(combined, gene , cat1 = "icu_status", cat2 = "mechanical_ventilation", cont = "ferritin.ng
```

```
## [1] "AAAS"
##   [1] 18.92 18.68 13.85 22.11  8.45 28.59 10.50 22.78 15.47 18.40 26.98  9.10
##  [13]  8.42 29.27 16.00 22.10 10.30  9.37 23.99 19.46 18.82 18.73 12.61  7.10
##  [25]  5.17  8.87 11.16 24.38 15.47 14.32 11.91  9.74 15.31 10.40  8.96 21.24
##  [37] 10.45 14.82 14.16 14.76 12.17 10.22 14.60  6.63 15.10  5.78 10.80  5.36
##  [49] 19.77 12.44 10.85 23.14  6.16 20.18 11.07 16.28 13.81 15.18 25.29 19.47
##  [61] 18.66 21.99 19.80 16.31 15.76  9.99 19.42 28.19 25.11 16.03 23.40 22.49
```

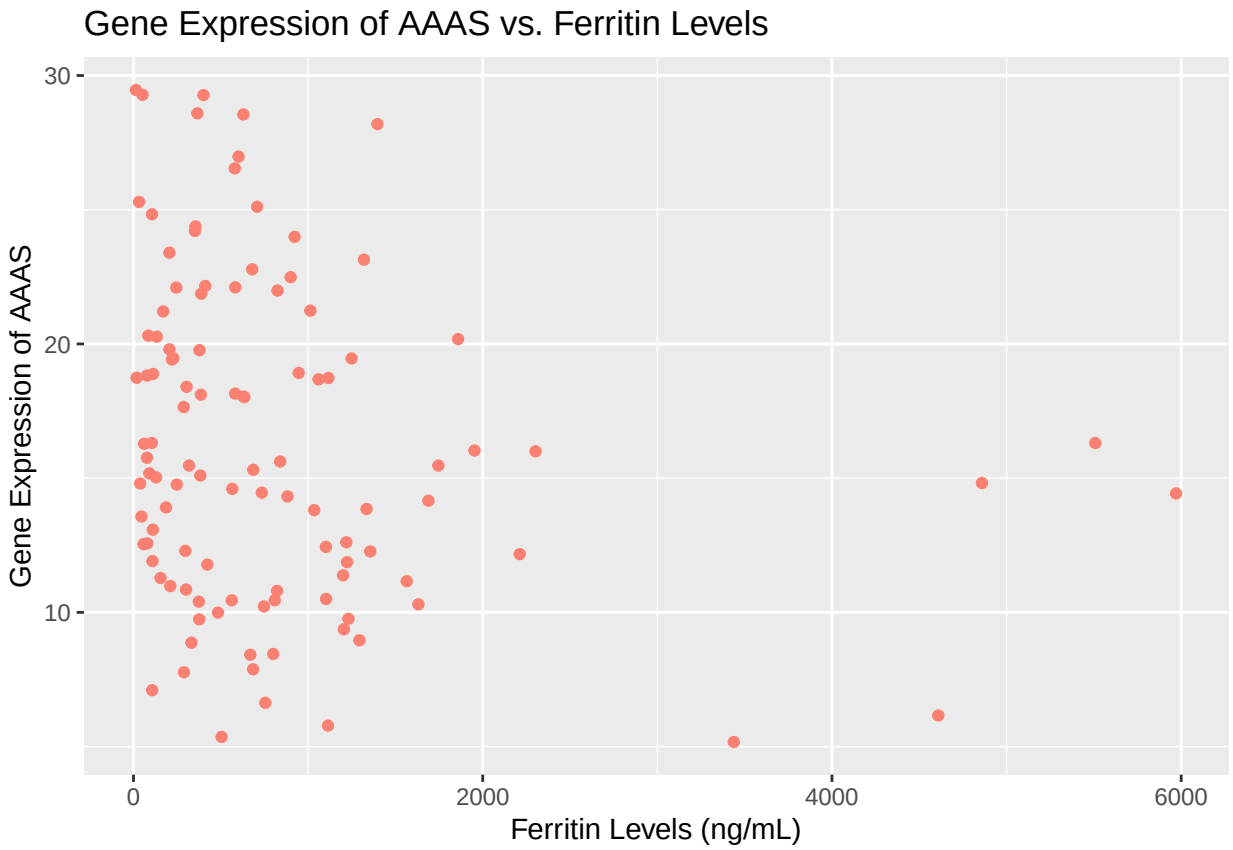
```
##  [73] 12.27 29.46 28.55 13.91 14.43  7.88 11.87 18.02 18.88 11.38 17.10 20.27
##  [85] 15.62 11.78 24.21 21.21 14.80 17.65 19.02 13.08 21.87 29.28 18.11 16.89
##  [97] 14.46 18.15  9.76 18.74 12.29 10.45 12.54 15.03 26.54 17.95 13.92 22.16
## [109] 12.57 18.04 11.35 15.53  7.77 24.64 16.26 16.31 10.98 11.28 13.57 24.83
## [121] 17.06 20.31 27.25 21.64  5.54
```

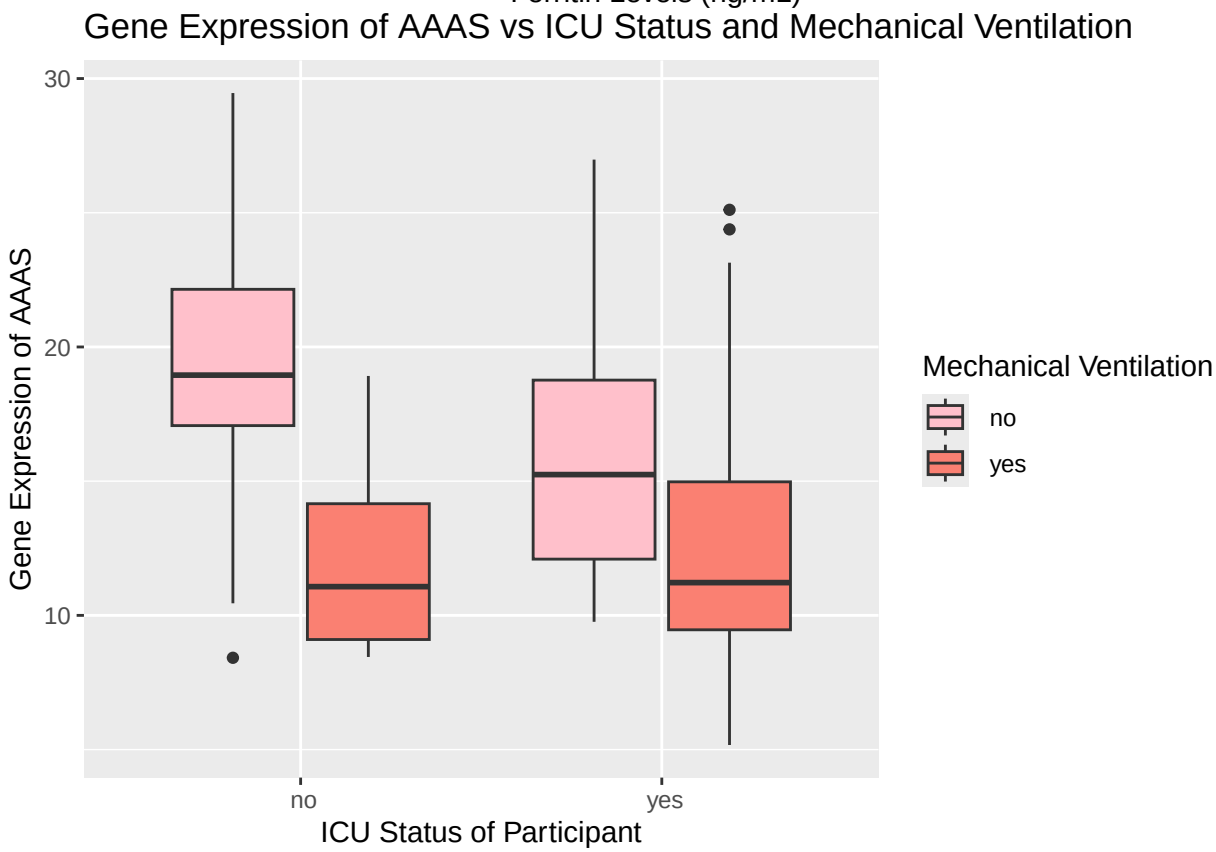
```
## Warning: `aes_string()` was deprecated in ggplot2 3.0.0.
## i Please use tidy evaluation idioms with `aes()`.
## i See also `vignette("ggplot2-in-packages")` for more information.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```

```
## Warning in geom_point(bins = 10, color = "salmon"): Ignoring unknown
## parameters: `bins`
```




```
## Warning: Removed 16 rows containing missing values or values outside the scale range
## (`geom_point()`).
```


Gene Expression of AAAS vs. Ferritin Levels
Gene Expression of AAAS
30
20
10
0
2000
4000
6000
Ferritin Levels (ng/mL)


Gene Expression of AAAS vs ICU Status and Mechanical Ventilation
Gene Expression of AAAS
30
20
10
no
yes
ICU Status of Participant
Mechanical Ventilation
no
yes

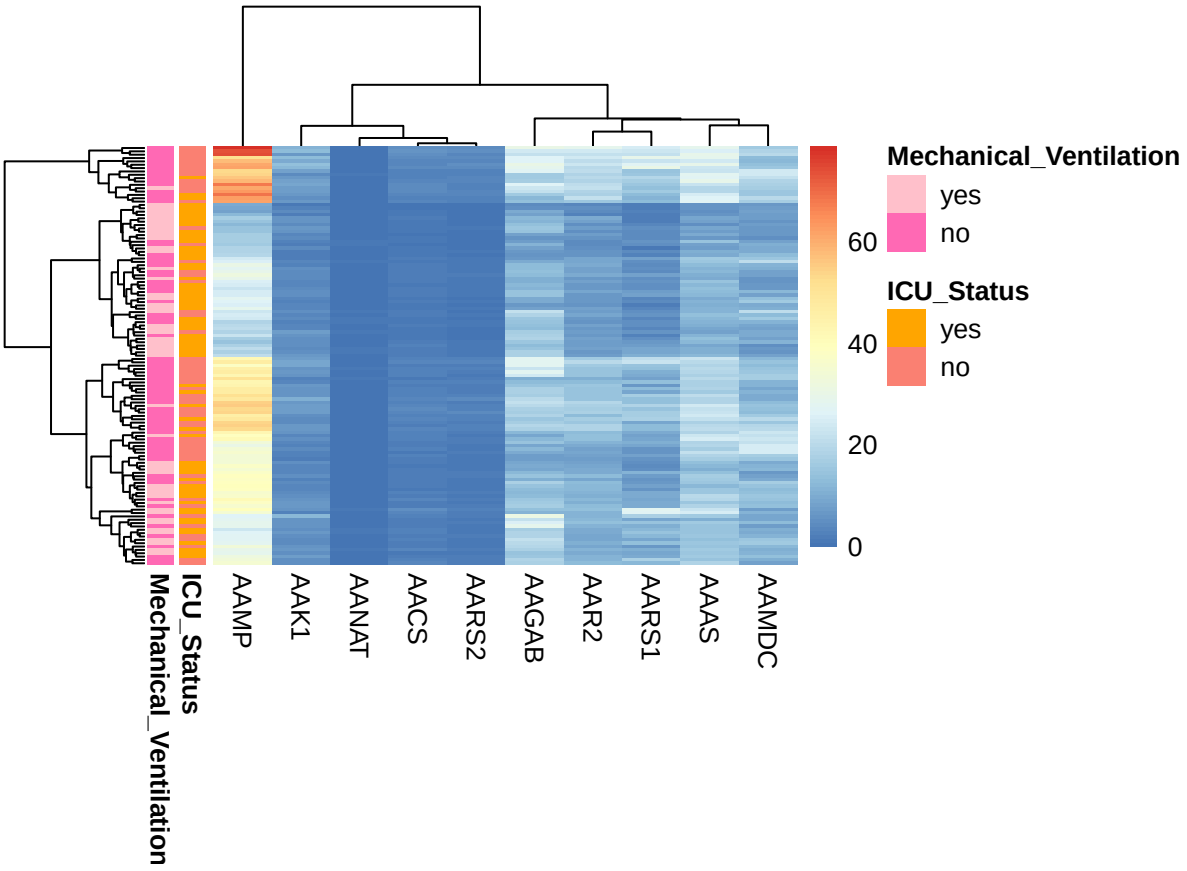
```
#sub_2_plots(combined, "AAAS", cat1 = "icu_status", cat2 = "mechanical_ventilation", cont = "ferritin.
```

Generate a heatmap (5 pts) Heatmap should include at least 10 genes Include tracking bars for the 2 categorical covariates in your boxplot Heatmaps should include clustered rows and columns

```
#install.packages('pheatmap')
library(pheatmap)

#citation(package = 'pheatmap')
```

```
#head(select(participant_id, AAAS, AACS, AAGAB, AAK1, AAMDC, AAMP, AANAT, AAR2, AARS1, AARS2))
#making all the selected genes numeric
combined$AAGAB <- as.numeric(combined$AAGAB)
combined$AAK1 <- as.numeric(combined$AAK1)
combined$AAMDC <- as.numeric(combined$AAMDC)
combined$AAMP <- as.numeric(combined$AAMP)
combined$AANAT <- as.numeric(combined$AANAT)
combined$AAR2 <- as.numeric(combined$AAR2)
combined$AARS1 <- as.numeric(combined$AARS1)
combined$AAAS <- as.numeric(combined$AAAS)
combined$AACS <- as.numeric(combined$AACS)
combined$AARS2 <- as.numeric(combined$AARS2)

#head(combined)

#new_combined(combined) <- c('AAAS', 'AACS', 'AAGAB', 'AAK1', 'AAMDC', 'AAMP', 'AANAT', 'AAR2', 'AARS1'
```

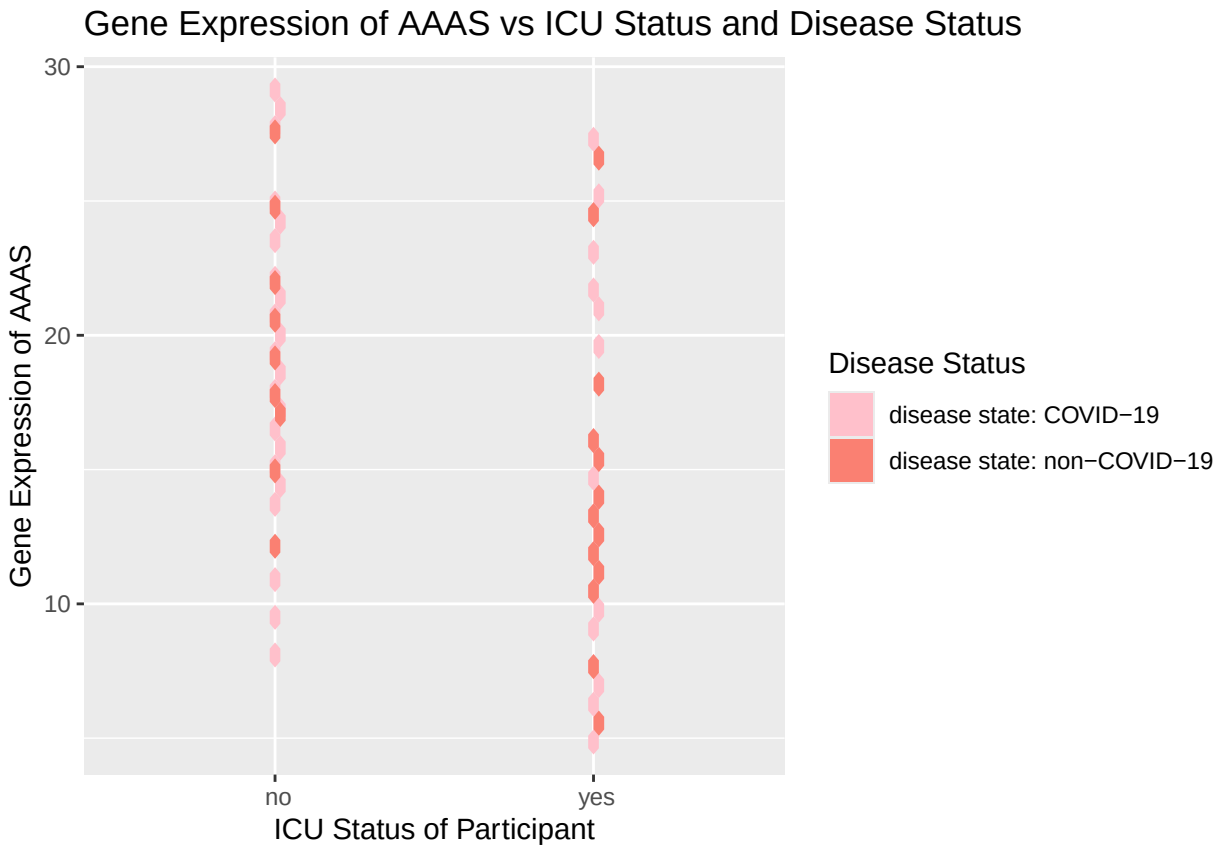
```
combined_df <- data.frame(combined[,c('AAAS', 'AACS', 'AAGAB', 'AAK1', 'AAMDC', 'AAMP', 'AANAT', 'AAR2'

#new data frame with labeled ICU Status from the column same for mechanical ventilation
annotationData <- data.frame(ICU_Status =combined$icu_status ,
                             Mechanical_Ventilation= combined$mechanical_ventilation,
                             row.names = row.names(combined_df)
                             )

row.names(combined_df) <- row.names(combined) #making the row names from this data
row.names(annotationData) <- row.names(combined)

#coloring the icu and mechanical ventilation with the colors, be sure to put space between yes and no b
annotationColors <- list(ICU_Status= c(' yes' = 'orange',
                                       ' no' = 'salmon'),
                         Mechanical_Ventilation= c(' yes' = 'pink',
                                                   ' no' = 'hotpink')
                         )

#combined$icu_status

pheatmap(combined_df,
         show_rownames = F,
         cluster_rows = T,
         cluster_cols = T,
         annotation_row = annotationData, #be sure this is not annotation_col
         annotation_colors = annotationColors
         )
```



Going through the documentation for ggplot2, generate a plot type that we did not previously discuss in class that describes your data in a new and unique way (5 pts)

```
#citation(package = 'hexbin') #citation for the new plot
```

```
#look through ggplot to find cool new ways
#install.packages('hexbin')
library(ggplot2)

#similar process to histogram and scatterplot with a few adjustments
new_plot <- ggplot(combined, aes(x=icu_status, y = AAAS, fill = disease_status)) + #need to add a fill
  geom_hex() + #to generate a hexbin use geom_hex
  scale_fill_manual(values = c('pink','salmon')) +
  labs(title = 'Gene Expression of AAAS vs ICU Status and Disease Status',  #to label each attribute of
       x= 'ICU Status of Participant' ,
       y= 'Gene Expression of AAAS',
       fill= 'Disease Status')
plot(new_plot)
```



Submit a LaTeX file and knitted PDF file summarizing your results (20 pts total). This file should include the following sections: Table of summary statistics Histogram of gene Scatter plot of gene + continuous covariate Boxplot of gene stratified by 2 categorical covariates Heatmap Your selected new plot type Introduction: Brief description of the data set and your gene of choice for your main plots. Additional gene descriptions aren't required for genes included in your heatmap. Methods: Brief summary of methods including data source, R version and packages, and clustering algorithm used, as discussed in class. Results: Description of the findings of each table/figure (outlined below). While you do not need to provide an extensive analysis of each item, you must provide a brief statement referencing them and then cite the relevant table or figure, as discussed in class. Example: "Gene x did not appear to be associated with covariate y (Figure 2)." Additionally, you must typeset and provide captions/figure legends for each item as discussed in class. Required elements include: References: At a minimum, your references must include the paper that the dataset came from, an original source for your gene description, and R packages used. All references should be cited within the text as shown in class. Submit a link to your github repository for review (15 pts total; 5 per presentation) Push all the clearly commented code for your final submission. You must have a commit from before each presentation including all of the code used for each presentation. Repository must be public facing after final submission. Provide a brief presentation of your new plot type providing a description of what it shows and why you think it's useful. (5 pts) Reminder: All figures and tables should be "publication ready" (i.e. clean variable names, etc. as discussed in class). While we have not deducted points for this so far in prior iterations of this project, you will lose points on your final submission for sloppy figures and tables.