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)

# 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)
```

```
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 = "darkseagreen", 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
#}

# gene 1
#fun_stats_pretty_plots(matrix = merged_matrix,gene_name='AAMP',continuous_name = 'age',categorical1_na


# gene 2
```

```
#fun_stats_pretty_plots(matrix = merged_matrix,gene_name='AAAS',continuous_name = 'age',categorical1_na


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

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
```

```
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 lubridate 1.9.3     v tibble    3.2.1
## v purrr     1.0.2     v tidyr     1.3.1
## -- 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 errors
```

```
library(knitr) # for base kable function
#install.packages('tableone')
library(tableone)

# set the continous variables to numeric
merged_matrix$age <- as.numeric(merged_matrix$age)
merged_matrix$hospital.free_days_post_45_day_followup <- as.numeric(merged_matrix$hospital.free_days_pos
merged_matrix$ventilator.free_days <- as.numeric(merged_matrix$ventilator.free_days)

# vars to be used in tableone
vars <- c("icu_status","mechanical_ventilation","age","hospital.free_days_post_45_day_followup","ventila

## Create Table 1 stratified by sex
tableOne <- CreateTableOne(vars = vars, strata ="sex", data = merged_matrix)

# final table to be saved into CSV
finished_table <- print(tableOne,showAllLevels = T,
                        nonnormal=c("hospital.free_days_post_45_day_followup","ventilator.free_days"))
```

```
##                                                          Stratified by sex
##                                                           level
##    n
##    icu_status (%)                                          no
##                                                            yes
##    mechanical_ventilation (%)                              no
##                                                            yes
##    age (mean (SD))
##    hospital.free_days_post_45_day_followup (median [IQR])
##    ventilator.free_days (median [IQR])
##                                                          Stratified by sex
##                                                            female
```

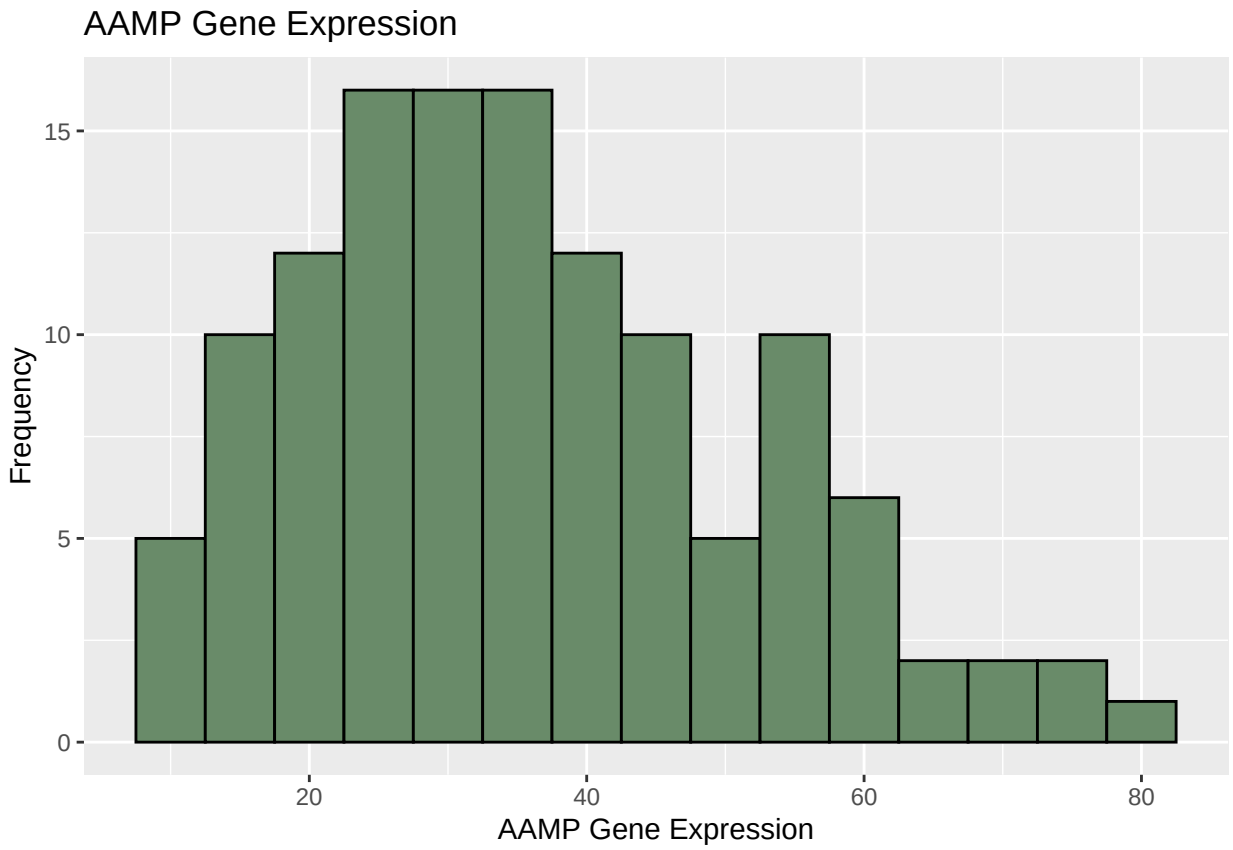
```
##   n                                                          51
##   icu_status (%)                                             27 (52.9)
##                                                              24 (47.1)
##   mechanical_ventilation (%)                                 35 (68.6)
##                                                              16 (31.4)
##   age (mean (SD))                                         59.30 (17.92)
##   hospital.free_days_post_45_day_followup (median [IQR]) 34.00 [14.50, 40.00]
##   ventilator.free_days (median [IQR])                     28.00 [18.00, 28.00]
##                                                          Stratified by sex
##                                                            male
##   n                                                          73
##   icu_status (%)                                             32 (43.8)
##                                                              41 (56.2)
##   mechanical_ventilation (%)                                 38 (52.1)
##                                                              35 (47.9)
##   age (mean (SD))                                         62.28 (14.41)
##   hospital.free_days_post_45_day_followup (median [IQR]) 27.00 [0.00, 39.00]
##   ventilator.free_days (median [IQR])                     28.00 [9.00, 28.00]
##                                                          Stratified by sex
##                                                            unknown
##   n                                                             1
##   icu_status (%)                                                0 (  0.0)
##                                                                 1 (100.0)
##   mechanical_ventilation (%)                                    1 (100.0)
##                                                                 0 (  0.0)
##   age (mean (SD))                                         83.00 (NA)
##   hospital.free_days_post_45_day_followup (median [IQR]) 30.00 [30.00, 30.00]
##   ventilator.free_days (median [IQR])                     28.00 [28.00, 28.00]
##                                                          Stratified by sex
##                                                           p      test
##   n
##   icu_status (%)                                           0.387
##
##   mechanical_ventilation (%)                               0.128
##
##   age (mean (SD))                                             NA
##   hospital.free_days_post_45_day_followup (median [IQR])   0.448 nonnorm
##   ventilator.free_days (median [IQR])                      0.623 nonnorm
```

```r
write.csv(finished_table,"tableOne.csv") # save to csv
```

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

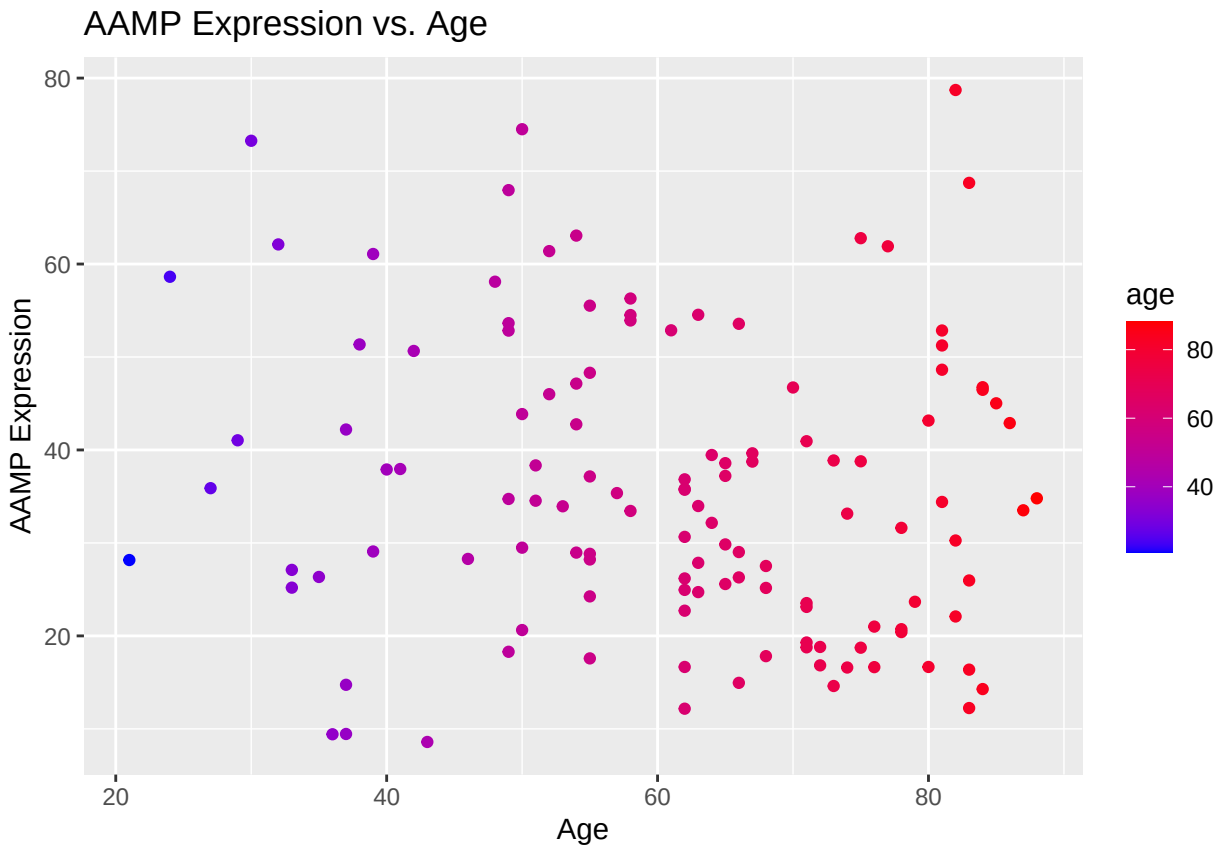
```r
# genes to be used during plotting - stored in a list

# 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
paper_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(paper_histo)
```




```
ggsave("AAMP_gene_expression_histogram.pdf", plot = paper_histo, width = 8, height = 5)


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

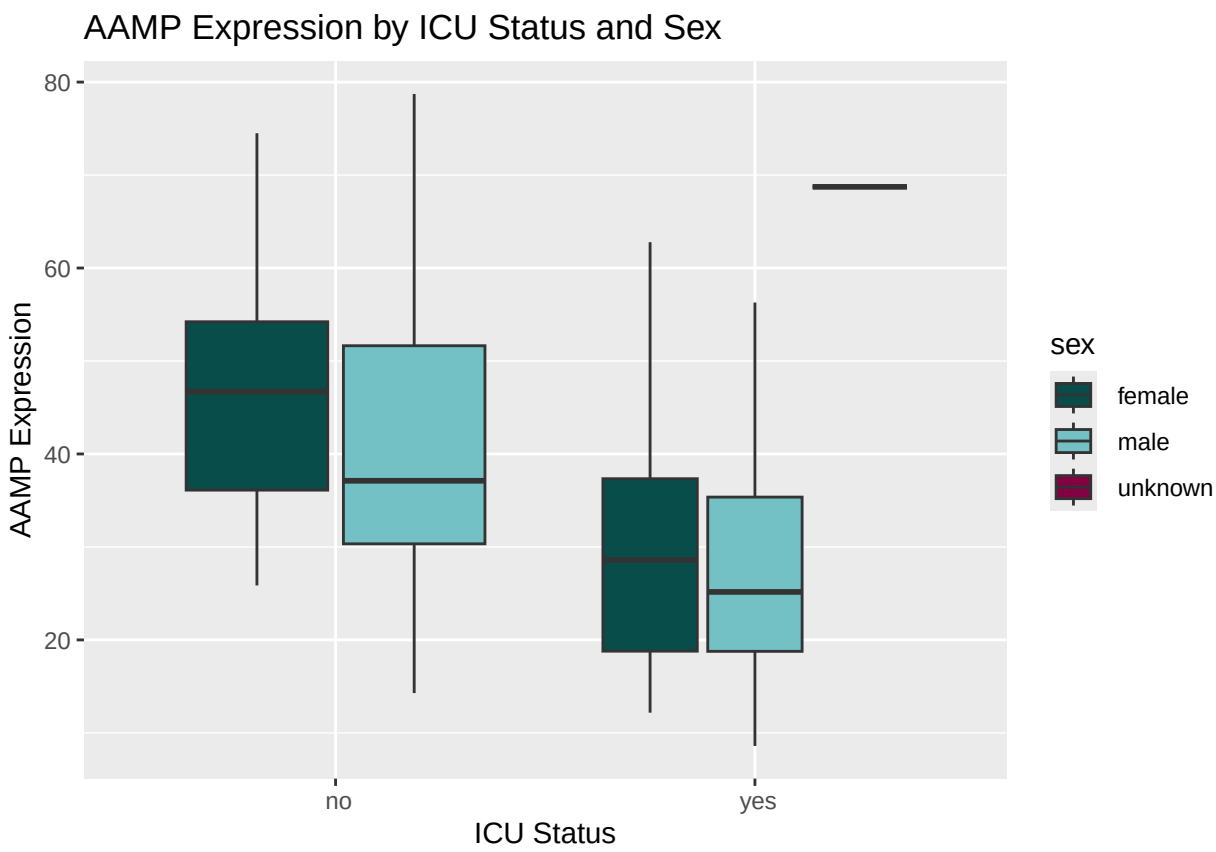
```
## Warning: Removed 2 rows containing missing values or values outside the scale range
## (`geom_point()`).
```




```
ggsave("AAMP_gene_expression_vs_age_scatterplot.pdf", plot = paper_scatter, width = 8, height = 5)
```

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

```
#a boxplot of AAMP expression separated by sex and ICU Status
paper_box <- 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") + scale_fill_hp_d(option = "lunalovegood")
plot(paper_box)
```



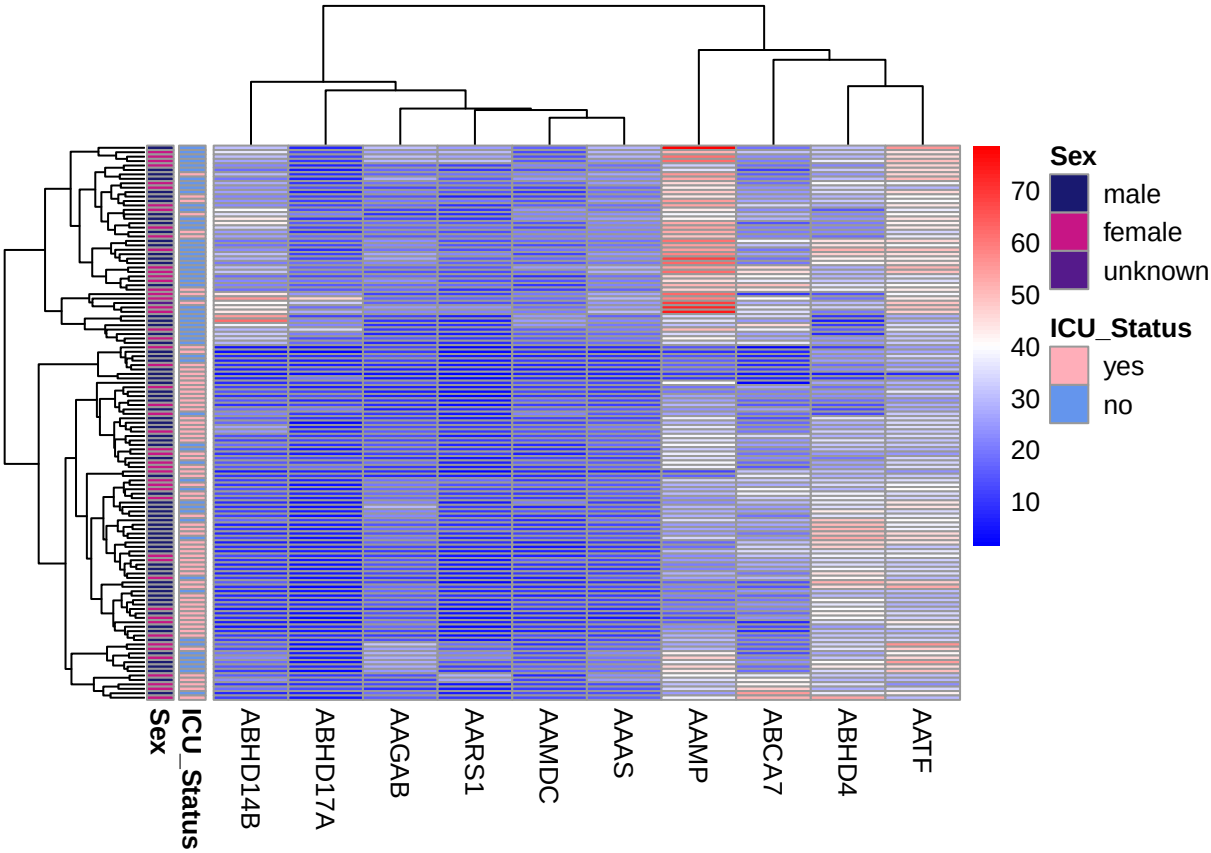

```
ggsave("AAMP_gene_expression_vs_icu_sex_boxplot.pdf", plot = paper_box, width = 8, height = 5)
```

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
```

```
merged_matrix$AAMP <- as.numeric(merged_matrix$AAMP) # 1 - my gene
merged_matrix$AAMDC <- as.numeric(merged_matrix$AAMDC) # 2
merged_matrix$AAAS <- as.numeric(merged_matrix$AAAS) # 3
merged_matrix$AARS1 <- as.numeric(merged_matrix$AARS1) # 4
merged_matrix$AAGAB <- as.numeric(merged_matrix$AAGAB) # 5
merged_matrix$ABCA7 <- as.numeric(merged_matrix$ABCA7) # 6
merged_matrix$ABHD14B <- as.numeric(merged_matrix$ABHD14B) # 7
merged_matrix$ABHD4 <- as.numeric(merged_matrix$ABHD4) # 8
merged_matrix$AATF <- as.numeric(merged_matrix$AATF) # 9
merged_matrix$ABHD17A <- as.numeric(merged_matrix$ABHD17A) # 10

# genes for heatmap (& balloon plot)
heatmap_genes <- c('AAMP','AAMDC','AAAS','AARS1','AAGAB','ABCA7','ABHD14B','ABHD4','AATF','ABHD17A')

# new data frame for annotating
annotation <- data.frame(ICU_Status = merged_matrix$icu_status,
                         Sex = merged_matrix$sex,
                         row.names = row.names(merged_matrix))
```

```r
# row names of annotation match the merged matrix row names
row.names(annotation) <- row.names(merged_matrix)
row.names(merged_matrix) <- row.names(merged_matrix)

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


# colors for annotating
annotationColors <- list(ICU_Status = c(' yes' = 'lightpink1',' no' = 'cornflowerblue'),
                         Sex =c(' male' = 'midnightblue',' female' = 'mediumvioletred',' unknown'='purpl

# heatmap
pheatmap(merged_matrix[,heatmap_genes],
  show_rownames = F,
  cluster_rows = T,
  cluster_cols = T,
  color = colorRampPalette(c("blue", "white", "red"))(100), # color bar
  annotation_row = annotation,
  annotation_colors = annotationColors) #,
```



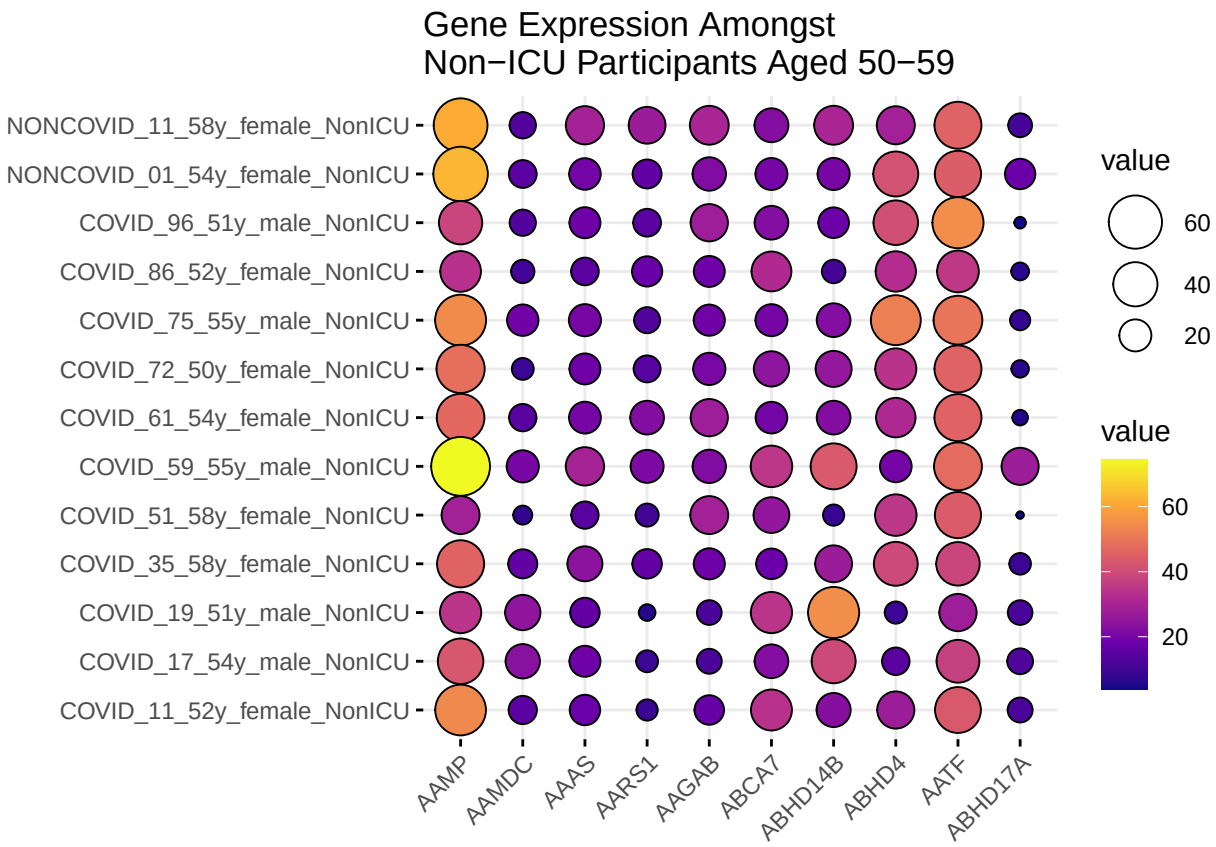

```r
  #filename = 'heatmap.pdf' )
```

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)

```
library(ggpubr)
#install.packages("viridis")
library(viridis)
```

```
## Loading required package: viridisLite
```

```
#balloon_genes <- c("AAMP","AAAS")

fifties_df <- subset(merged_matrix, age >= 50 & age <= 59 & icu_status == " no") # subset by participan
#that have not been to the icu

#head(fifties_df)

# balloon plot using the genes used in the heatmap
balloon <- ggballoonplot(fifties_df[,heatmap_genes], fill = "value")+
  scale_fill_viridis_c(option = "C")  +  # Apply the Viridis color scale
  scale_y_discrete(labels = fifties_df$participant_id) + labs(title = "Gene Expression Amongst\nNon-ICU

plot(balloon)
```




```
ggsave("balloon_plot.pdf", plot = balloon, width = 8, height = 5) # save to a pdf
```