Final_Project

2024-08-20

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(tidyr)

setwd("/Users/jainishah/Desktop/final_project_data") # set working directory
gene_expression <- read.csv("genes.csv") # reading csv files
metadata <- read.csv("metadata.csv")


gene_long <- gene_expression %>% # converts data from wide to long format
  tidyr::gather(key = "ParticipantID", value = "Expression", -X) # 'key' column is 'ParticipantID' and

names(gene_long)[names(gene_long) == "X"] <- "Gene" # renames the 'X' column to 'Gene'

merged_data <- merge(gene_long, metadata, by.x = "ParticipantID", by.y = "participant_id") # combines d


clean_data <- merged_data %>% # extracting specific columns from 'merged data'
  filter(Gene == Gene) %>%
  select(Gene, ParticipantID, Expression, age, ferritin.ng.ml., lactate.mmol.l., sex, icu_status, diseas

# cleaning 'clean_data' df
clean_data$age <- as.numeric(clean_data$age)
```

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

```
clean_data$ferritin.ng.ml. <- as.numeric(clean_data$ferritin.ng.ml.)
```

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

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

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

```
clean_data$sex <- as.factor(clean_data$sex)
clean_data$icu_status <- as.factor(clean_data$icu_status)
clean_data$disease_status <- as.factor(clean_data$disease_status)


clean_data[clean_data == ' unknown'] <- NA # replaces 'unknown' with 'NA'


plot_data <- clean_data %>% # creating new df for 3 plots by selectin relevant columns from 'clean_data
  select(Gene, ParticipantID, Expression, age, ferritin.ng.ml., lactate.mmol.l., sex, icu_status, diseas
  filter(Gene == 'ABCA1') # specifically using 'ABCA1' column
```

```
# 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') Prof Meghan suggested to use this package to create table
#install.packages('kableExtra')

#https://www.rdocumentation.org/packages/tableone/versions/0.13.2/topics/CreateTableOne used this site

#install.packages('xtable')

library(tableone)
library(dplyr)
library(tidyr)
library(xtable)
library(stringr)

table_data <- clean_data %>% # converts data from long to wide format
  tidyr::pivot_wider(names_from = Gene, values_from = Expression) %>% #
  dplyr::select(c('age', 'ferritin.ng.ml.', 'lactate.mmol.l.', 'sex', 'icu_status', 'disease_status')) 
  dplyr::rename( # renaming the columns to be labeled properly
    Age = age,
    Ferritin = ferritin.ng.ml.,
    Lactate = lactate.mmol.l.,
    Sex = sex,
    ICU_Status = icu_status,
    Disease_Status = disease_status
  ) %>%
  mutate(
    Sex = str_to_title(Sex), # converts the values of these two columns to title case
    ICU_Status = str_to_title(ICU_Status)
  )


categorical_vars <- c("Sex", "ICU_Status") # defining categorical variables
continuous_vars <- c("Age", "Ferritin", "Lactate") # defining continuous variables
```

```r
table1 <- CreateTableOne(vars = c(categorical_vars, continuous_vars), # function generates table with c
                         strata = "Disease_Status", data = table_data, factorVars = categorical_vars) #

table1
```

```
##                          Stratified by Disease_Status
##                           disease state: COVID-19 disease state: non-COVID-19
##   n                           99                        26
##   Sex =  Male (%)             61 (61.6)                 12 (48.0)
##   ICU_Status =  Yes (%)       50 (50.5)                 16 (61.5)
##   Age (mean (SD))          60.84 (16.15)             62.80 (15.61)
##   Ferritin (mean (SD))    936.73 (1099.29)          250.50 (238.21)
##   Lactate (mean (SD))       1.25 (0.51)               2.10 (2.03)
##                          Stratified by Disease_Status
##                           p      test
##   n
##   Sex =  Male (%)          0.313
##   ICU_Status =  Yes (%)    0.434
##   Age (mean (SD))          0.586
##   Ferritin (mean (SD))     0.015
##   Lactate (mean (SD))      0.003
```

```r
# converts table1 table into a matrix of strings that can be exported
# ensures that all levels (male and female, yes and no) are displayed
tabAsStringMatrix <- print(table1, showAllLevels = TRUE, printToggle = FALSE, noSpaces = TRUE)

xtable(tabAsStringMatrix)
```

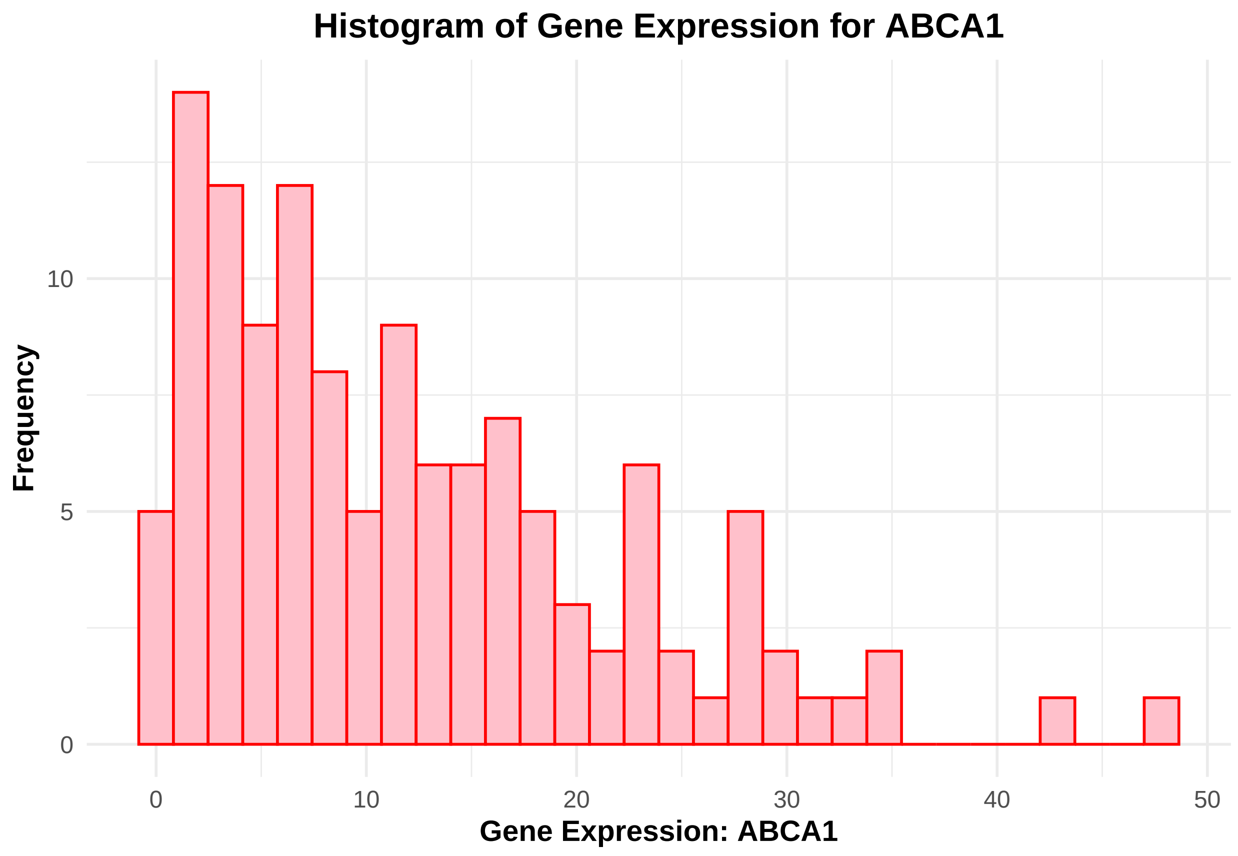
```
## % latex table generated in R 4.4.1 by xtable 1.8-4 package
## % Wed Aug 28 19:53:02 2024
## \begin{table}[ht]
## \centering
## \begin{tabular}{rllllll}
##   \hline
##  & level & disease state: COVID-19 & disease state: non-COVID-19 & p & test \\
##   \hline
## n &  & 99 & 26 &  &  \\
##   Sex.... & Female & 38 (38.4) & 13 (52.0) & 0.313 &  \\
##   X & Male & 61 (61.6) & 12 (48.0) &  &  \\
##   ICU\_Status.... & No & 49 (49.5) & 10 (38.5) & 0.434 &  \\
##   X.1 & Yes & 50 (50.5) & 16 (61.5) &  &  \\
##   Age..mean..SD.. &  & 60.84 (16.15) & 62.80 (15.61) & 0.586 &  \\
##   Ferritin..mean..SD.. &  & 936.73 (1099.29) & 250.50 (238.21) & 0.015 &  \\
##   Lactate..mean..SD.. &  & 1.25 (0.51) & 2.10 (2.03) & 0.003 &  \\
##    \hline
## \end{tabular}
## \end{table}
```

```r
# converts matrix into LaTeX table, adds caption, label, and specifies alignment
latex_table <- xtable(tabAsStringMatrix, caption = "Descriptive Statistics Stratified by Disease Status
                      label = "tab:descriptive_stats", align = "lrrrrr")
```

```r
# outpits LaTeX table to file name and ensures that the output is in latex format
print(latex_table, type = "latex", file = "table1.tex", include.rownames = FALSE)
```

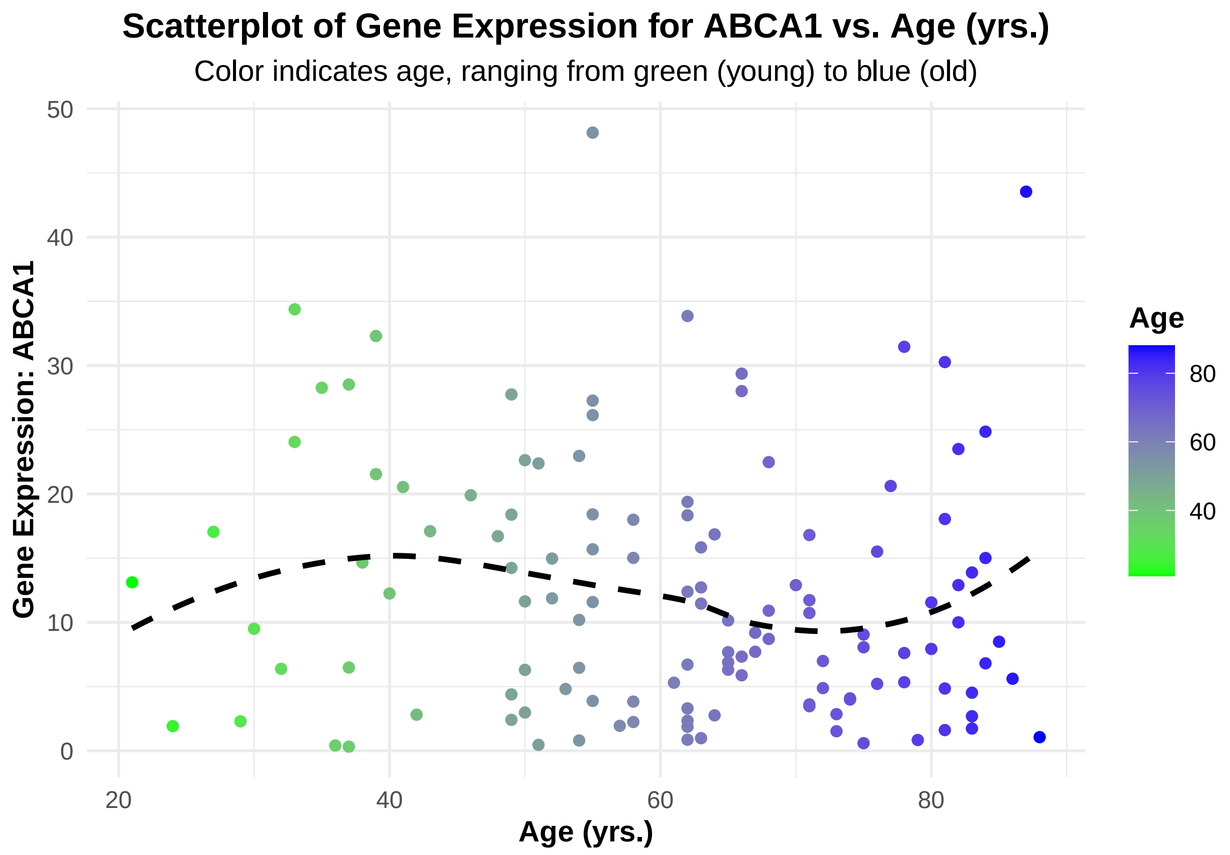
```r
# Generate final histogram, scatter plot, and boxplot from submission 1 (i.e. only for your first gene


# Histogram for Gene Expression
library(ggplot2)

# initializes the plot with 'final_data' and maps all 'Expression' values to the x-axis
ggplot(plot_data, aes(x = Expression)) +
  geom_histogram(fill
                 = "pink", color = "red") +
  # plots the histogram with bars outlined and filled
  labs(title = "Histogram of Gene Expression for ABCA1", # adds title and axis labels
       x = "Gene Expression: ABCA1",
       y = "Frequency") +
  theme_minimal() + # applied minimal theme for appearance
  theme( # making the titles of plot and axis bold
    plot.title = element_text(hjust = 0.5, face = "bold"),
    axis.title = element_text(face = "bold"),
    legend.title = element_text(face = "bold")
  )
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

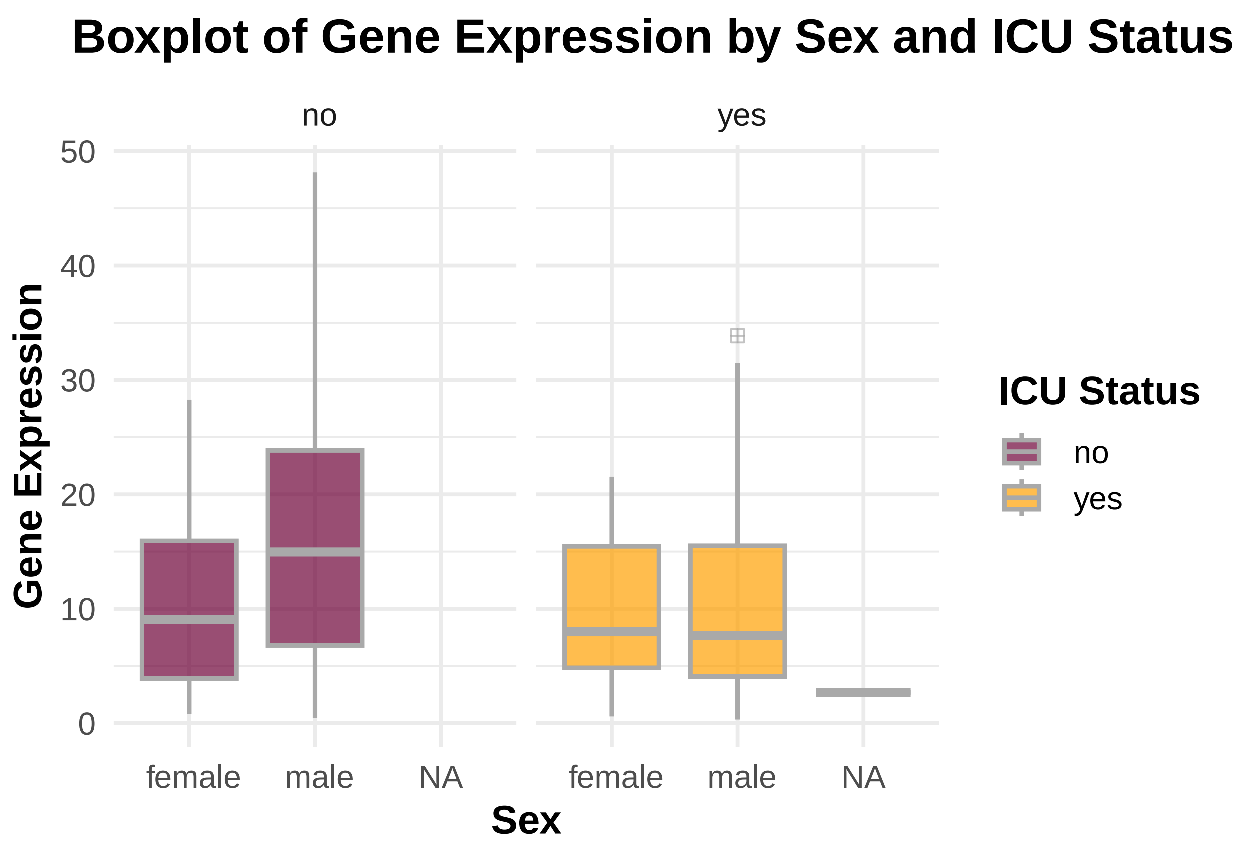


```r
# Scatterplot for gene expression and continuous covariate (Age)
library(ggplot2)

# initializes the plot with 'final_data' and maps all 'age' values to x-axis 'Expression' values to the
ggplot(plot_data,aes(x = age,y = Expression ,color = age)) +
  geom_point() +
  scale_color_gradient(low = "green", high = "blue") + # color gradient for age
  labs(title = "Scatterplot of Gene Expression for ABCA1 vs. Age (yrs.)", # title for plot and axis
       subtitle = "Color indicates age, ranging from green (young) to blue (old)",
       x = "Age (yrs.)",
       y = "Gene Expression: ABCA1",
       color = "Age") +
  theme_minimal() +
  theme( # making the titles of axis and plot bold
    plot.title = element_text(hjust = 0.5, face = "bold"),
    plot.subtitle = element_text(hjust = 0.5),
    axis.title = element_text(face = "bold"),
    legend.title = element_text(face = "bold")
  ) +

# adding a smooth line to show trend in data
geom_smooth(method = "loess", se = FALSE, color = "black", linetype = "dashed")
```

```
## `geom_smooth()` using formula = 'y ~ x'

## Warning: Removed 2 rows containing non-finite outside the scale range
## (`stat_smooth()`).

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



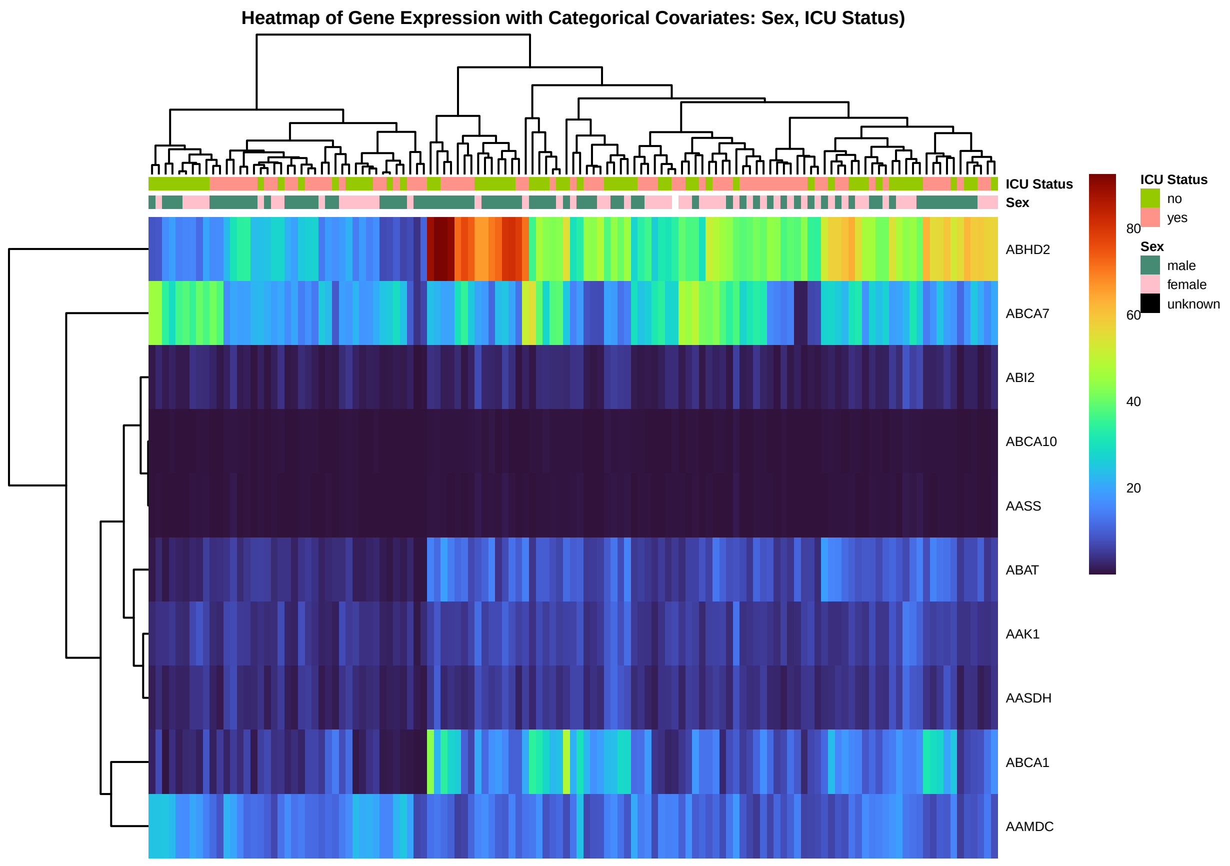
```
# Boxplot of gene expression separated by both categorical covariates (Sex and ICU Status)
library(ggplot2)
library(harrypotter)

# create boxplot with categorical covariates
ggplot(plot_data,aes(x = sex ,y = Expression, fill = icu_status)) +
   geom_boxplot(outlier.shape = 12, outlier.size = 2, color = "darkgray", lwd = 0.8, alpha = 0.7) +
    scale_fill_hp_d(option = "ronweasley") + # change border color and box details
  facet_wrap(~ icu_status) + # separate plots by ICU status
  labs(title = "Boxplot of Gene Expression by Sex and ICU Status", # assigned titles for axis and plot
       x = "Sex",
       y = "Gene Expression",
       fill = "ICU Status") +
  theme_minimal(base_size = 15) + # minimal theme with font size
  theme( # making the titles of axis and plot bold, changing placement of titles
    plot.title = element_text(hjust = 0.125, face = "bold"),
    axis.title = element_text(face = "bold"),
    legend.title = element_text(face = "bold")
  )
```



```r
# 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

#ask about cololring variables and changing fontsize

#install.packages("viridis")
library(pheatmap)
library(dplyr)
library(tidyr)
library(tibble)
library(viridis)
```

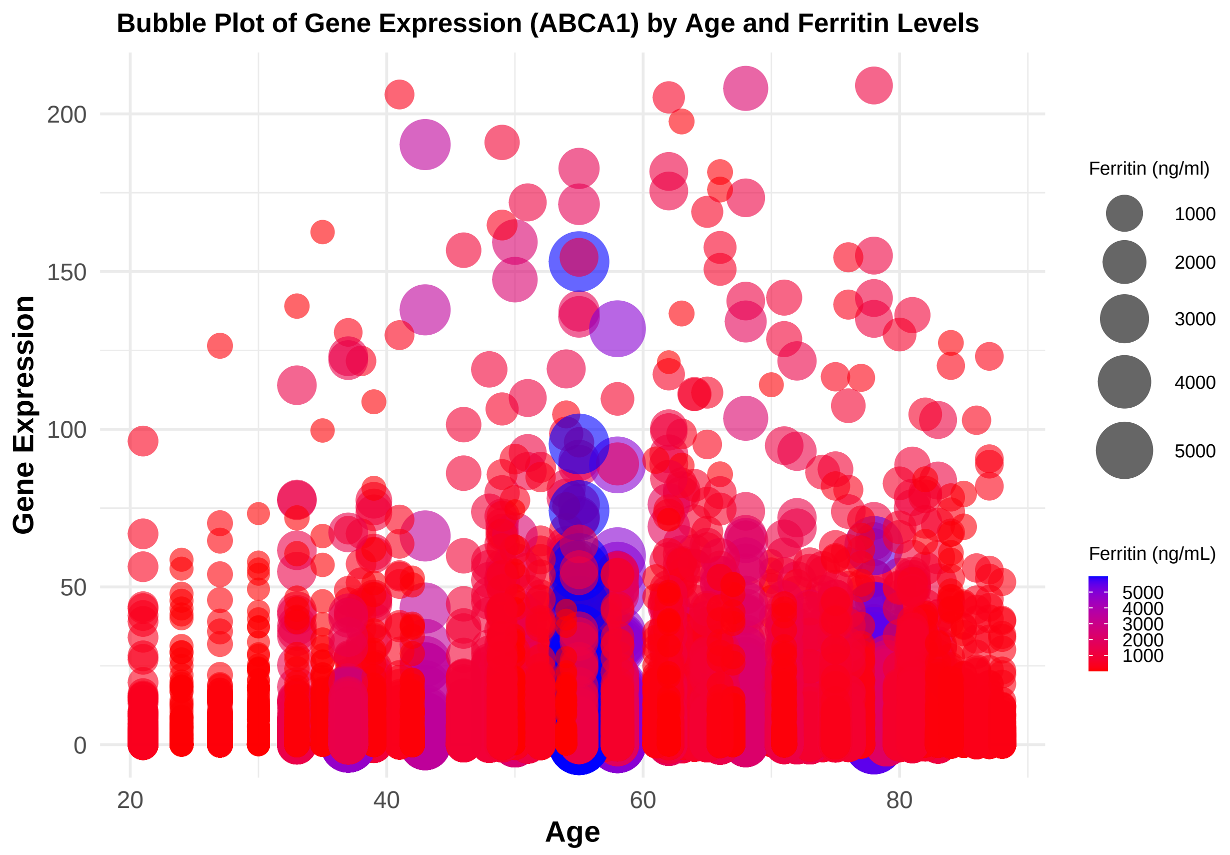
```
## Loading required package: viridisLite
```

```r
# filtering the data for the 10 genes
heatmap_data <- clean_data %>%
  dplyr::filter(Gene %in% c('ABCA1','AAK1', 'AASDH', 'ABI2', 'ABHD2', 'AAMDC', 'ABI2','ABCA10','ABCA7',
  tidyr::pivot_wider(names_from = Gene, values_from = Expression) # reshaping the data from long to wid


# selecting the columns corresponding to the genes of interest to create matrix
heatmap_matrix <- heatmap_data %>%
  select(all_of(c('ABCA1', 'AAK1', 'AASDH','ABI2', 'ABHD2','AAMDC','ABI2','ABCA10','ABCA7','AASS','ABAT
  as.matrix()

row.names(heatmap_matrix) <- heatmap_data$ParticipantID # 'participantID' column us used as row names f

heatmap_matrix <- as.data.frame(t(heatmap_matrix)) # transpose the matrix s othe genes are rows and par


annotation1 <- data.frame(row.names = colnames(heatmap_matrix),
                         'Sex' = heatmap_data$sex,
                         'Icu Status' = heatmap_data$icu_status)


# list is created to define the colors for each annotation
annotation_colors <- list(
  Sex = c(' male' = 'aquamarine4', ' female' = 'pink', ' unknown' = 'black'),
  Icu.Status = c('yes' = 'royalblue', 'no' = 'deepskyblue'))


colnames(annotation1) <- c('Sex','ICU Status')

pheatmap(heatmap_matrix,
         color = viridis::turbo(500), # color palette applied to represent expression levels
         annotation_col = annotation1, # annotations 'sex' and 'icu_status' are applied to annotations
         annotation_colors = annotation_colors, # predefined colors are specified
         clustering_distance_rows = 'euclidean', # euclidean distance is used to cluster genes
         clustering_distance_cols = 'euclidean', # euclidean distance is used to cluster participants
         show_rownames = TRUE, # 'genes' displayed
         show_colnames = FALSE, # 'particiapntID' not displayed
         fontsize = 5, # font size
         annotation_legend = TRUE, # annotation legend displayed
         main = 'Heatmap of Gene Expression with Categorical Covariates: Sex, ICU Status)') # title
```



```r
#   Going through the documentation for ggplot2, generate a plot type that we did not previously discus
#install.packages('gganimate')

#https://r-statistics.co/Top50-Ggplot2-Visualizations-MasterList-R-Code.html#Density%20Plot used this s

library(ggplot2)
library(dplyr)


# mapping 'age' to x axis and 'expression' to y axis, 'ferritin' to size of bubbles and color of bubble
ggplot(clean_data, aes(x = age, y = Expression, size = ferritin.ng.ml., color = ferritin.ng.ml.)) +
  geom_point(alpha = 0.6) + # makes points semi-transparent
  scale_size(range = c(3, 10)) +  # adjusts range of sizes for the bubbles
  scale_color_gradient(low = 'red', high = 'blue') + # gradient color scale for bubbles
  labs(title = "Bubble Plot of Gene Expression (ABCA1) by Age and Ferritin Levels", # title
       x = "Age", # x axis label
       y = "Gene Expression", # y axis label
       size = "Ferritin (ng/ml)", # label for size legend
       color = 'Ferritin (ng/mL)') + # label fo color legend
  theme_minimal() + # minimalistic theme
  theme(plot.title = element_text(hjust = 0.2, face = "bold",size = 10), # centers plot title and bolds
        axis.title = element_text(face = "bold"),  #bolds the axis titles
        legend.key.size = unit(0.25, 'cm'), #change legend key size
        legend.key.height = unit(0.25, 'cm'), #change legend key height
        legend.key.width = unit(0.25, 'cm'),
        legend.text = element_text(size = 7),
        legend.title = element_text(size = 7))
```

```
## Warning: Removed 1800 rows containing missing values or values outside the scale range
## (`geom_point()`).
```


Bubble Plot of Gene Expression (ABCA1) by Age and Ferritin Levels
Gene Expression
Age
Ferritin (ng/ml)
Ferritin (ng/mL)