Project_part2_atreml

Alexandra Treml

2024-08-06

Part 2 includes: 1. Build a function to create the plots from part 1. 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

2. 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
3. Present one of your boxplots in class.

R Markdown

```
#DATA CLEANING AND PREP

# Set the working directory and read in the files
setwd("C:/Users/AlexandraTreml/Desktop/MS/QBS103/Project/QBS103_proj")
genes <- read.csv("genes_GSE157103.csv")
participant <- read.csv("series_matrix_GSE157103.csv")

# Rename the first column in genes to 'gene'
genes <- genes %>%
  rename(gene = X)

# Pivot the genes dataframe
genes_long <- genes %>%
  tidyr::pivot_longer(cols = -gene, names_to = 'id', values_to = 'expression')

#rename id column in participant, select my categorical covariates and continuous variable (+age for ed
participant <- participant %>%
  filter(icu_status == ' yes') %>%
  rename(id = participant_id) %>%
  select(id, age, sex, mechanical_ventilation, lactate.mmol.l.)

#join 2 dataframes
df <- left_join(participant, genes_long, by = "id")

#fill lactate unknowns with NA
df$lactate.mmol.l.[df$lactate.mmol.l. == " unknown"] <- NA
#fill sex unknowns with NA
df$sex[df$sex == " unknown"] <- NA
#remove : from age, and fill with NA
```

```r
df$age[df$age == " :"] <- NA

head(df)
```

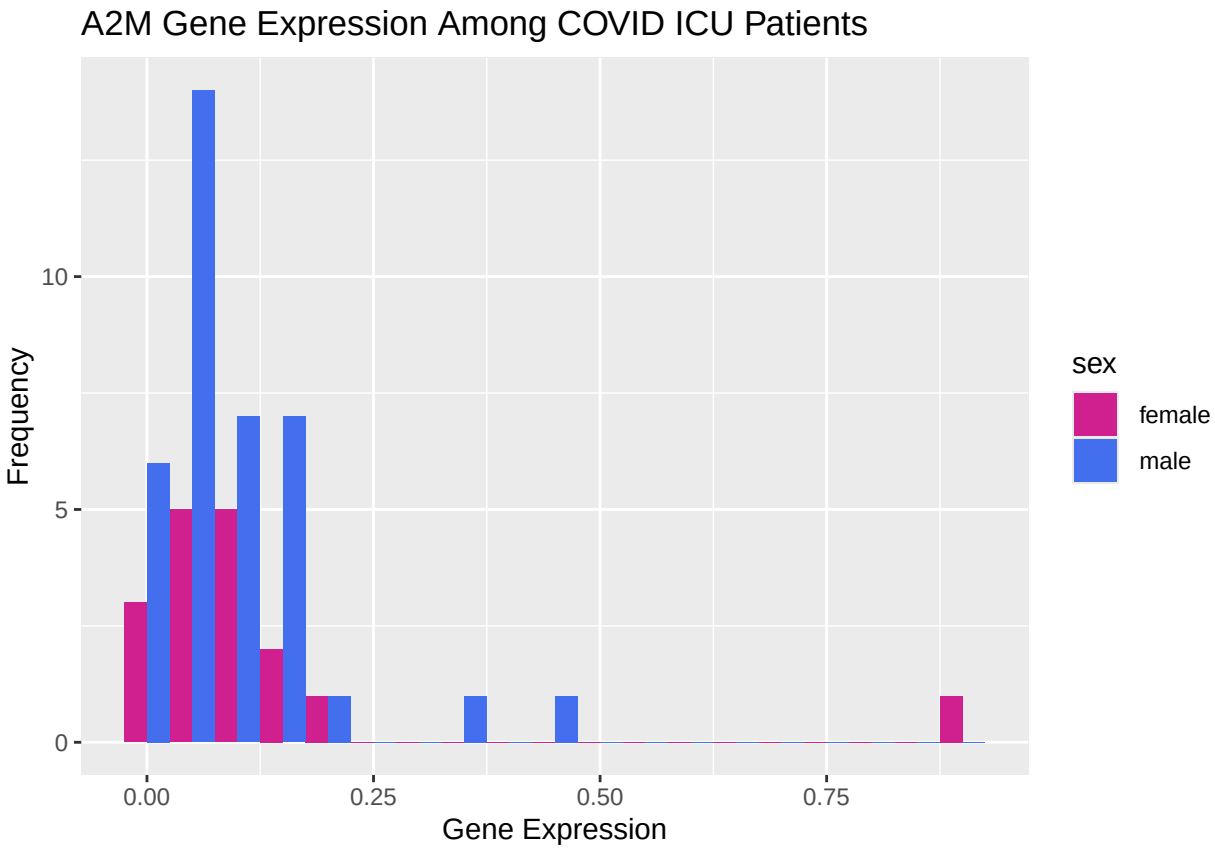
```
##                       id age   sex mechanical_ventilation lactate.mmol.l.
## 1 COVID_08_78y_male_ICU  78  male                     yes            1.65
## 2 COVID_08_78y_male_ICU  78  male                     yes            1.65
## 3 COVID_08_78y_male_ICU  78  male                     yes            1.65
## 4 COVID_08_78y_male_ICU  78  male                     yes            1.65
## 5 COVID_08_78y_male_ICU  78  male                     yes            1.65
## 6 COVID_08_78y_male_ICU  78  male                     yes            1.65
##      gene expression
## 1    A1BG       0.12
## 2    A1CF       0.00
## 3     A2M       0.08
## 4   A2ML1       0.01
## 5 A3GALT2       0.00
## 6  A4GALT       0.00
```

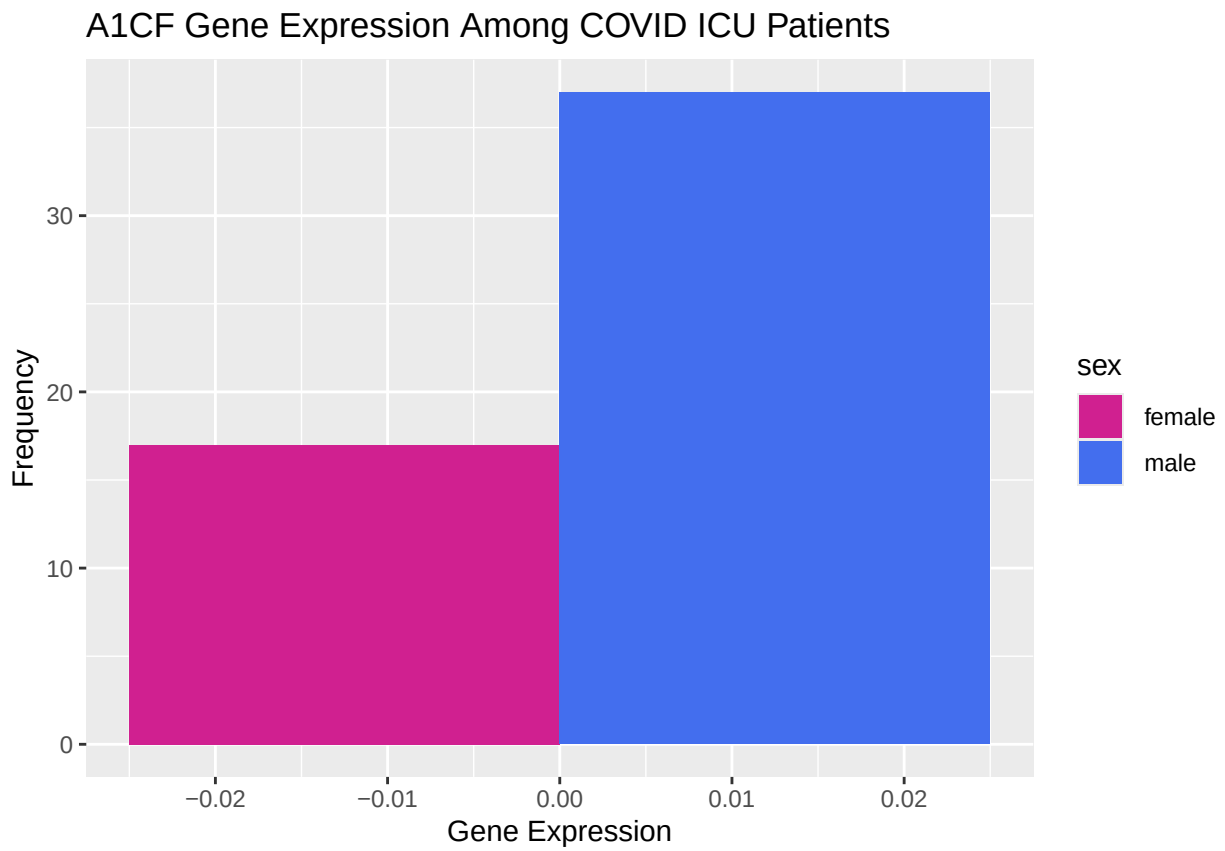
```r
hist_func <- function(df, gene_name, cont_covar, cat_covar1, cat_covar2) {
  df <- na.omit(df) #get rid of na

  df <- df %>%
    filter(gene == gene_name) %>% #filter for my gene in the for loop
    select(id, gene, cont_covar, cat_covar1, cat_covar2) #select just the cols I want from df

  df[[cat_covar1]] <- as.factor(df[[cat_covar1]]) #create as factor for my plot

  p <- ggplot(data = df, aes_string(x = cont_covar, fill = cat_covar2)) +
    geom_histogram(binwidth = 0.05, position = 'dodge') +
    labs(x = 'Gene Expression', y = 'Frequency',
         title = paste(gene_name, 'Gene Expression Among COVID ICU Patients')) +
    scale_fill_manual(values = c("violetred", "royalblue2"))

  print(p)
}

genes <- c('A4GALT', 'A2M', 'A1CF') #make a list of my 3 genes

# Loop through gene names to create 3 plots
for (gene in genes) {
  hist_func(df, gene, 'expression', 'age', 'sex') #loop through each gene and call func
}
```

```
## Warning: Using an external vector in selections was deprecated in tidyselect 1.1.0.
## i Please use ‘all_of()‘ or ‘any_of()‘ instead.
##   # Was:
##   data %>% select(cont_covar)
##
##   # Now:
##   data %>% select(all_of(cont_covar))
##
## See <https://tidyselect.r-lib.org/reference/faq-external-vector.html>.
```

```
## This warning is displayed once every 8 hours.
## Call ‘lifecycle::last_lifecycle_warnings()‘ to see where this warning was
## generated.

## Warning: Using an external vector in selections was deprecated in tidyselect 1.1.0.
## i Please use ‘all_of()‘ or ‘any_of()‘ instead.
##   # Was:
##   data %>% select(cat_covar1)
##
##   # Now:
##   data %>% select(all_of(cat_covar1))
##
## See <https://tidyselect.r-lib.org/reference/faq-external-vector.html>.
## This warning is displayed once every 8 hours.
## Call ‘lifecycle::last_lifecycle_warnings()‘ to see where this warning was
## generated.

## Warning: Using an external vector in selections was deprecated in tidyselect 1.1.0.
## i Please use ‘all_of()‘ or ‘any_of()‘ instead.
##   # Was:
##   data %>% select(cat_covar2)
##
##   # Now:
##   data %>% select(all_of(cat_covar2))
##
## See <https://tidyselect.r-lib.org/reference/faq-external-vector.html>.
## This warning is displayed once every 8 hours.
## Call ‘lifecycle::last_lifecycle_warnings()‘ to see where this warning was
## generated.

## 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.
```


A4GALT Gene Expression Among COVID ICU Patients
Frequency
20
10
0
0.0
0.2
0.4
0.6
Gene Expression
sex
female
male


A2M Gene Expression Among COVID ICU Patients
Frequency
10
5
0
0.00
0.25
0.50
0.75
Gene Expression
sex
female
male



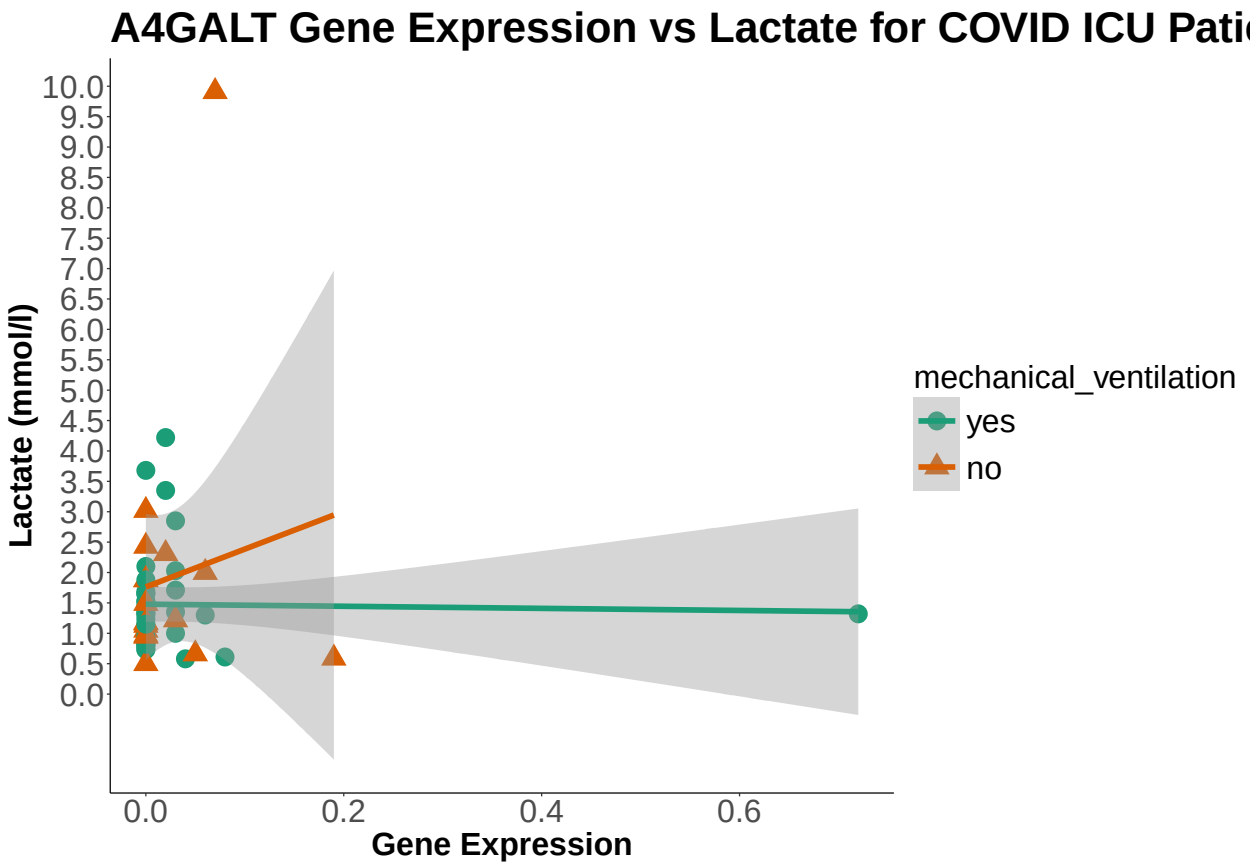

```
#create a scatterplot using gene expression and lactate

# Load necessary libraries
# Load necessary libraries
library(dplyr)
library(ggplot2)

scatterplot_func <- function(df, gene_name, expression, cont_covar, cat_covar1, cat_covar2) {

  #filter for my selected gene in the for loop
  df <- df %>% filter(gene == gene_name)

  ##onvert to numeric and clean up spaces
  df[[cont_covar]] <- as.numeric(as.character(df[[cont_covar]]))
  df[[cat_covar1]] <- trimws(as.character(df[[cat_covar1]]))
  df[[cat_covar2]] <- trimws(as.character(df[[cat_covar2]]))

  # Convert categorical columns to factors
  df[[cat_covar2]] <- factor(df[[cat_covar2]], levels = c("male", "female"))
  df[[cat_covar1]] <- factor(df[[cat_covar1]], levels = c("yes", "no"))

  # Scatterplot
  p <- ggplot(df, aes_string(x = expression, y = cont_covar, color = cat_covar1, shape = cat_covar1)) +
    geom_point(size = 3) +
    scale_y_continuous(breaks = seq(0, ceiling(max(df[[cont_covar]], na.rm = TRUE)), by = 0.5)) +
    geom_smooth(method = "lm", se = TRUE) +
```

```
    labs(x = 'Gene Expression', y = 'Lactate (mmol/l)', title = paste(gene_name, 'Gene Expression vs La
    theme_classic(base_size = 5) +
    scale_color_brewer(palette = 'Dark2') +
    scale_shape_manual(values = c(16, 17)) +
    theme(
      plot.title = element_text(size = 16, face = "bold"), # Center and bold title
      axis.title = element_text(size = 12, face = "bold"), # Bold axis titles
      axis.text = element_text(size = 12), # Increase axis text size
      legend.title = element_text(size = 12),
      legend.text = element_text(size = 12)
    )
  print(p) #print each plot now that it's in a function
}

genes <- c('A4GALT', 'A2M', 'A1CF')

#loop through my gene names to create 3 plots
for (gene in genes) {
  scatterplot_func(df, gene, 'expression', 'lactate.mmol.l.', 'mechanical_ventilation', 'sex')
}
```

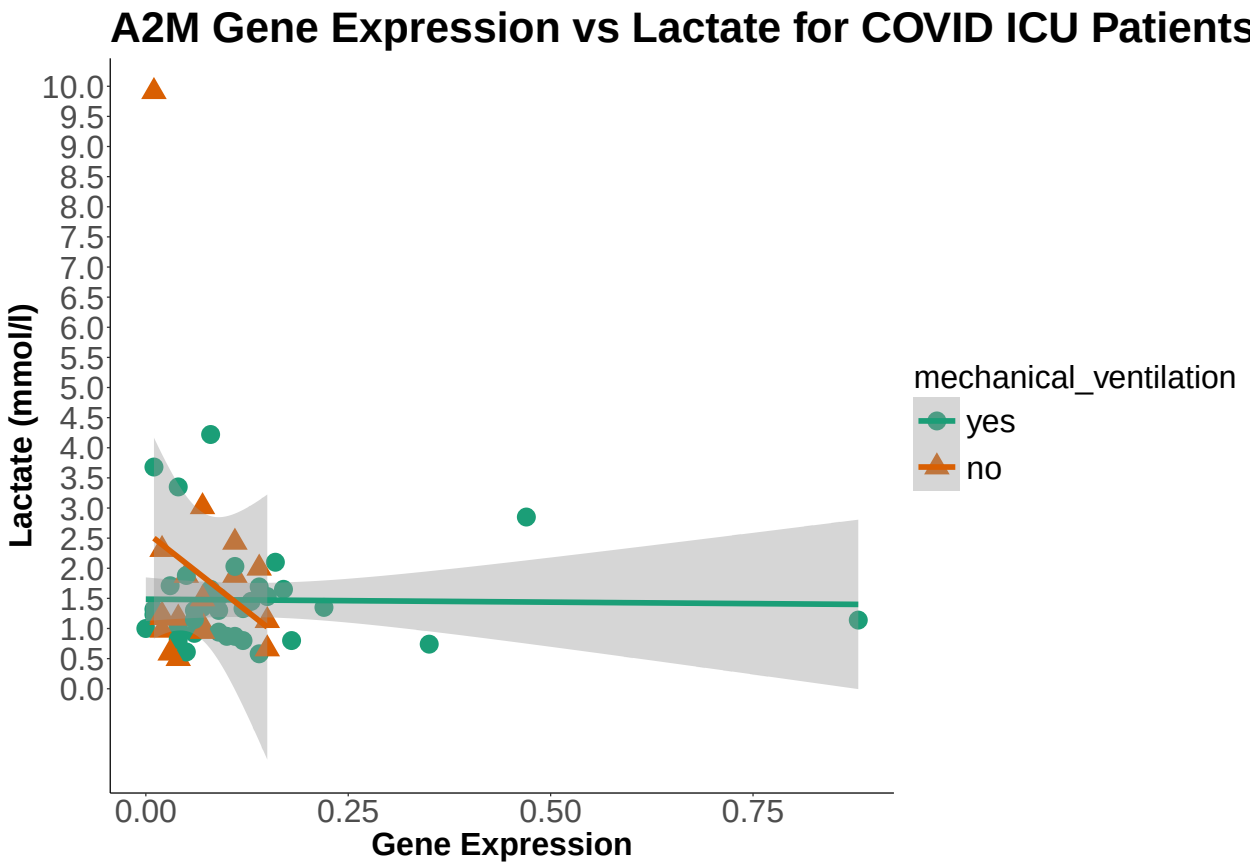
```
## `geom_smooth()` using formula = 'y ~ x'
```

```
## Warning: Removed 11 rows containing non-finite outside the scale range
## (`stat_smooth()`).
```

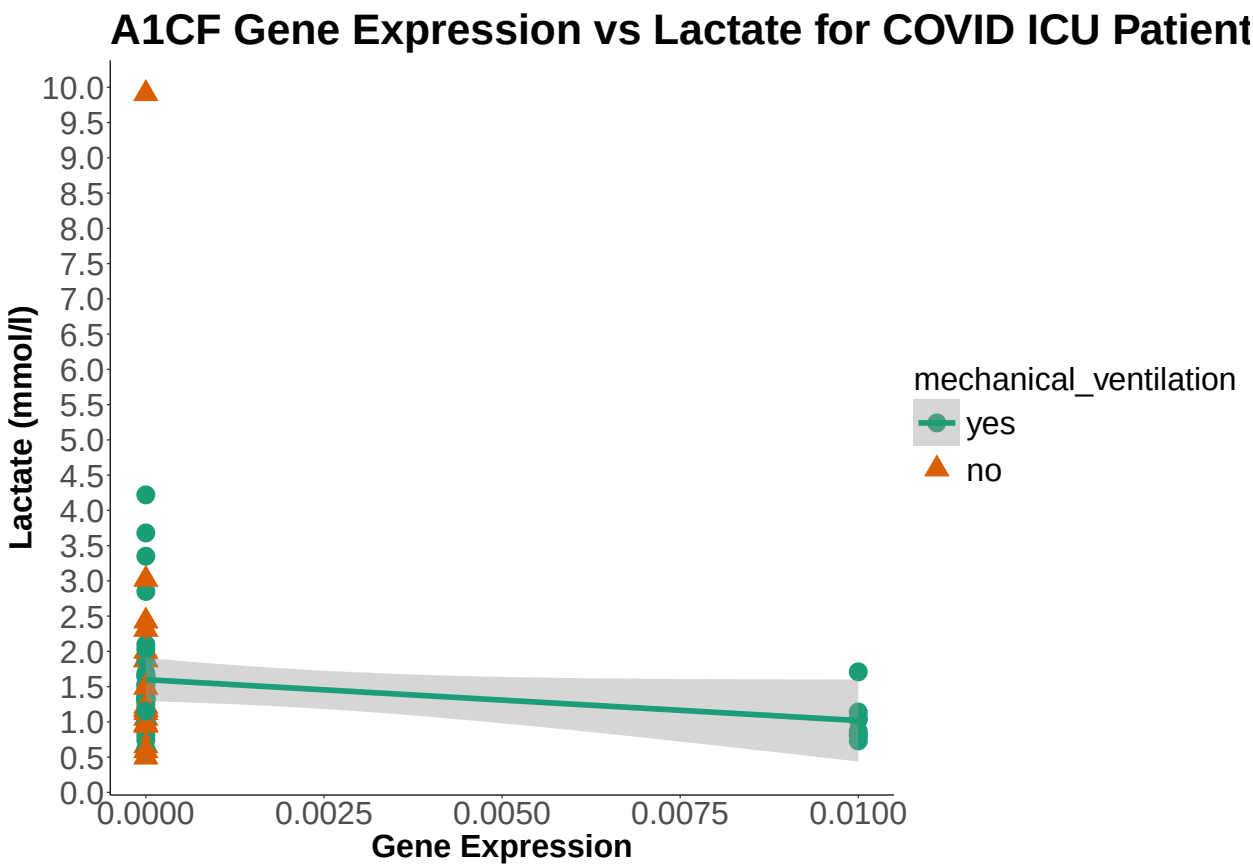
```
## Warning: Removed 11 rows containing missing values or values outside the scale range
## (`geom_point()`).
```




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

## Warning: Removed 11 rows containing non-finite outside the scale range
## (`stat_smooth()`).
## Removed 11 rows containing missing values or values outside the scale range
## (`geom_point()`).
```

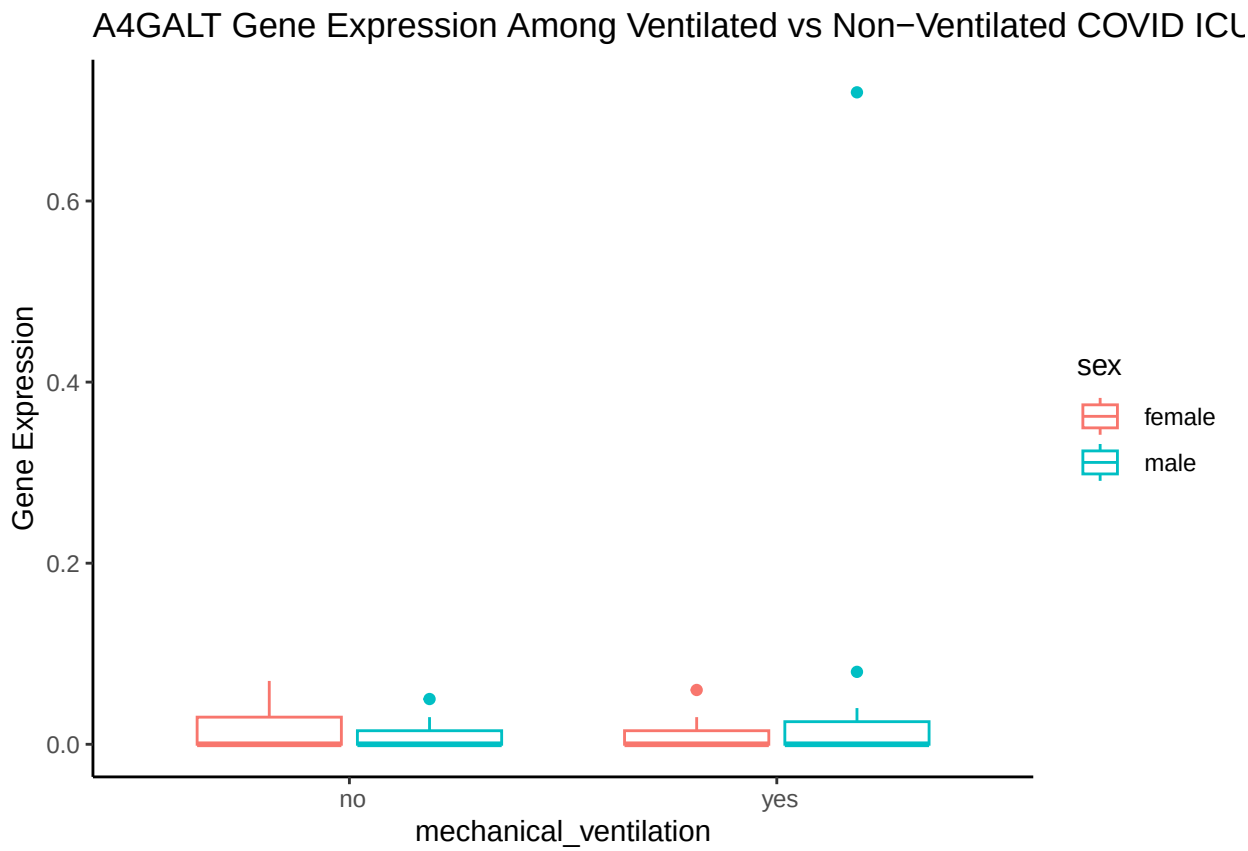



```
## `geom_smooth()` using formula = 'y ~ x'

## Warning: Removed 11 rows containing non-finite outside the scale range
## (`stat_smooth()`).
## Removed 11 rows containing missing values or values outside the scale range
## (`geom_point()`).
```

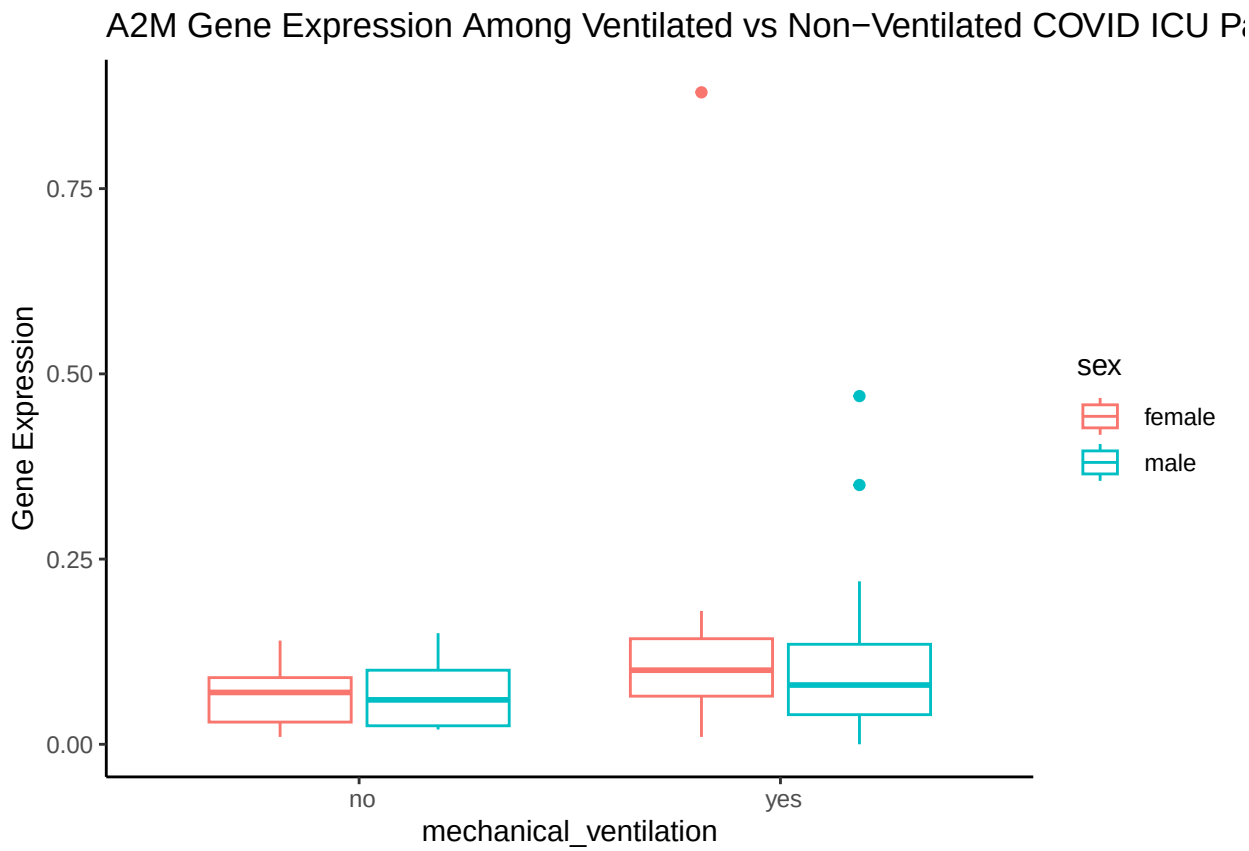



```
#boxplot of gene expression separated by both ventilator, sex
boxplot_func <- function(df, gene_name, expression, cat_covar1, cat_covar2) {
    df <- na.omit(df)
    df <- df %>%
    filter(gene == gene_name) #filter for just the gene in the loop

    p <- ggplot(df, aes_string(x = cat_covar1, y = expression, color = cat_covar2)) +
      geom_boxplot() +
      scale_fill_manual(values = c("male" = "royalblue1", "female" = "violetred1")) +
      labs(x = cat_covar1, y = 'Gene Expression', title = paste(gene_name, 'Gene Expression Among Venti
      theme_classic()

    print(p) #print each plot now that it's a function
}

genes <- c('A4GALT', 'A2M', 'A1CF') #set my 3 genes

# Loop through gene names to create 3 plots
for (gene in genes) {
  boxplot_func(df, gene, 'expression', 'mechanical_ventilation', 'sex')
}
```

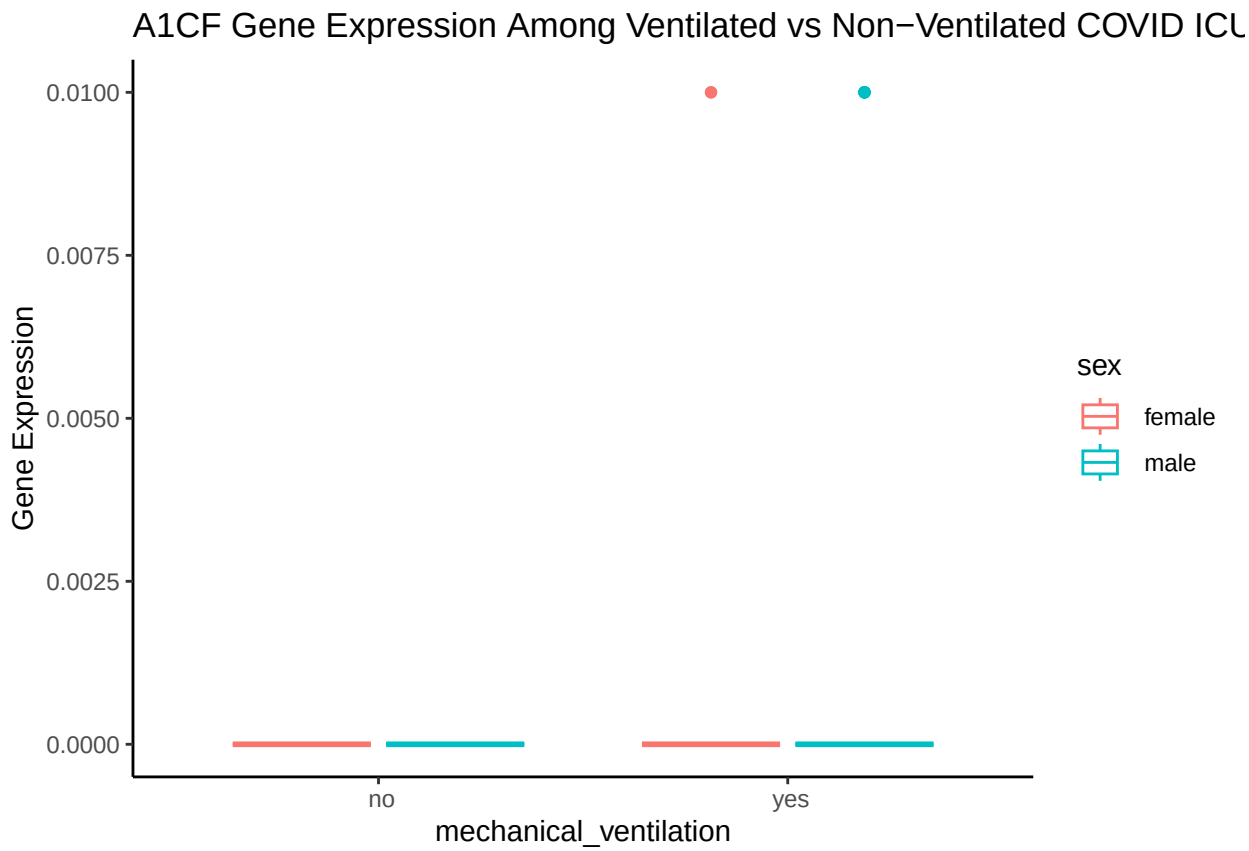
```
## Warning: No shared levels found between `names(values)` of the manual scale and the
## data's fill values.
```




```
## Warning: No shared levels found between ‘names(values)‘ of the manual scale and the
## data’s fill values.
```




```
## Warning: No shared levels found between ‘names(values)‘ of the manual scale and the
## data’s fill values.
```


A1CF Gene Expression Among Ventilated vs Non−Ventilated COVID ICU
Gene Expression
0.0100
0.0075
0.0050
0.0025
0.0000
no
yes
mechanical_ventilation
sex
female
male