final_assignment

2024-08-13

```
library(pheatmap)
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 errors
```

```
library(dplyr)
library(ggplot2)
library(rlang)
```

```
##
## Attaching package: 'rlang'
##
## The following objects are masked from 'package:purrr':
##
##     %@%, flatten, flatten_chr, flatten_dbl, flatten_int, flatten_lgl,
##     flatten_raw, invoke, splice
```

```
library(corrplot)
```

```
## corrplot 0.92 loaded
```

```
#install.packages("table1")
library(table1)
```

```
##
## Attaching package: 'table1'
##
## The following objects are masked from 'package:base':
##
##     units, units<-
```

```r
#--------------------------INITIALIZING DATA-------------------------
datafiles <- list.files(path = "/Users/amyw/Documents/QBS103/DSProject", pattern = ".csv")
print(datafiles)
```

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

```r
setwd("/Users/amyw/Documents/QBS103/DSProject")

for (file in datafiles){
  the_file <- read.csv(file)
}

gene_df <- read.csv('QBS103_GSE157103_genes.csv', header = TRUE,row.names = 1)
meta_df <- read.csv('QBS103_GSE157103_series_matrix.csv', header = TRUE, row.names = 1)

working_genes <- as.data.frame(t(gene_df)) #pivoting dataframe to select AAGAB as a column

new_comp <- merge(working_genes, meta_df, by = 'row.names')
#merging the two dataframes using the row names
final_comp <- data.frame(new_comp, row.names = 1)
#renaming the row names as the participant ID
```

```r
#----------------------SEPARATING BY CAT VARIABLE-----------------------
ventData <- data.frame(final_comp[final_comp$mechanical_ventilation == ' yes', ])
nonventData <- data.frame(final_comp[final_comp$mechanical_ventilation == ' no', ])

#------------------GETTING INDEX NUMBERS FOR CONT VARIABLES--------------------
contVars <- c('age', 'lactate.mmol.l.', 'fibrinogen')
contVarsIndex <- list()

for (i in 1:length(final_comp)) {
  for (j in 1:length(contVars)) {
    if (colnames(final_comp)[i] == contVars[j])
      print(paste('Index for', contVars[j] ,'is', i))
      contVarsIndex <- append(contVarsIndex, i)
    }
}
```

```
## [1] "Index for age is 110"
## [1] "Index for lactate.mmol.l. is 122"
## [1] "Index for fibrinogen is 123"
```

```r
contVarsIndex = c(110, 122, 123)

#---------------------------CHECKING DISTRIBUTION--------------------
for (index in contVarsIndex){
  #checking distribution
  xAxis = 0

  if (colnames(final_comp[index]) == 'age'){
    xAxis <- 'Age'
  }
```

```
  else if (colnames(final_comp[index]) == 'lactate.mmol.l.'){
    xAxis <- 'Lactate (mmol/l)'
  }
  else {
    xAxis <- 'Fibrinogen'
  }

  final_comp[[index]]<- as.numeric(final_comp[[index]])

  #plotting histograms for each continuous variable to check uniform/non-uniform distribution
  plot<- ggplot(data = final_comp,aes(x = final_comp[[index]])) +
                geom_histogram() +
                labs(x = xAxis, y = 'Frequency') +
                theme_classic()
  print(plot)

}
```

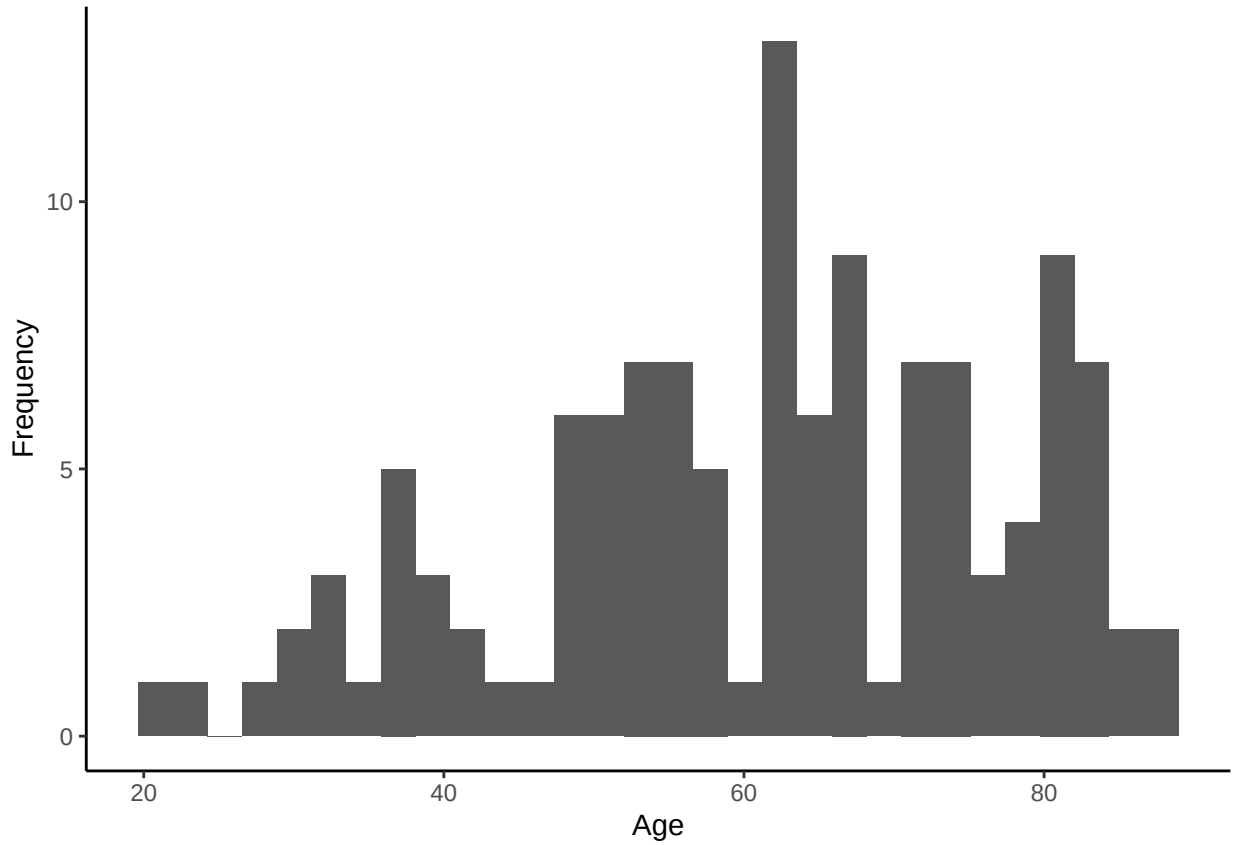
```
## Warning: NAs introduced by coercion
```

```
## Warning: Use of `final_comp[[index]]` is discouraged.
## i Use `.data[[index]]` instead.
```

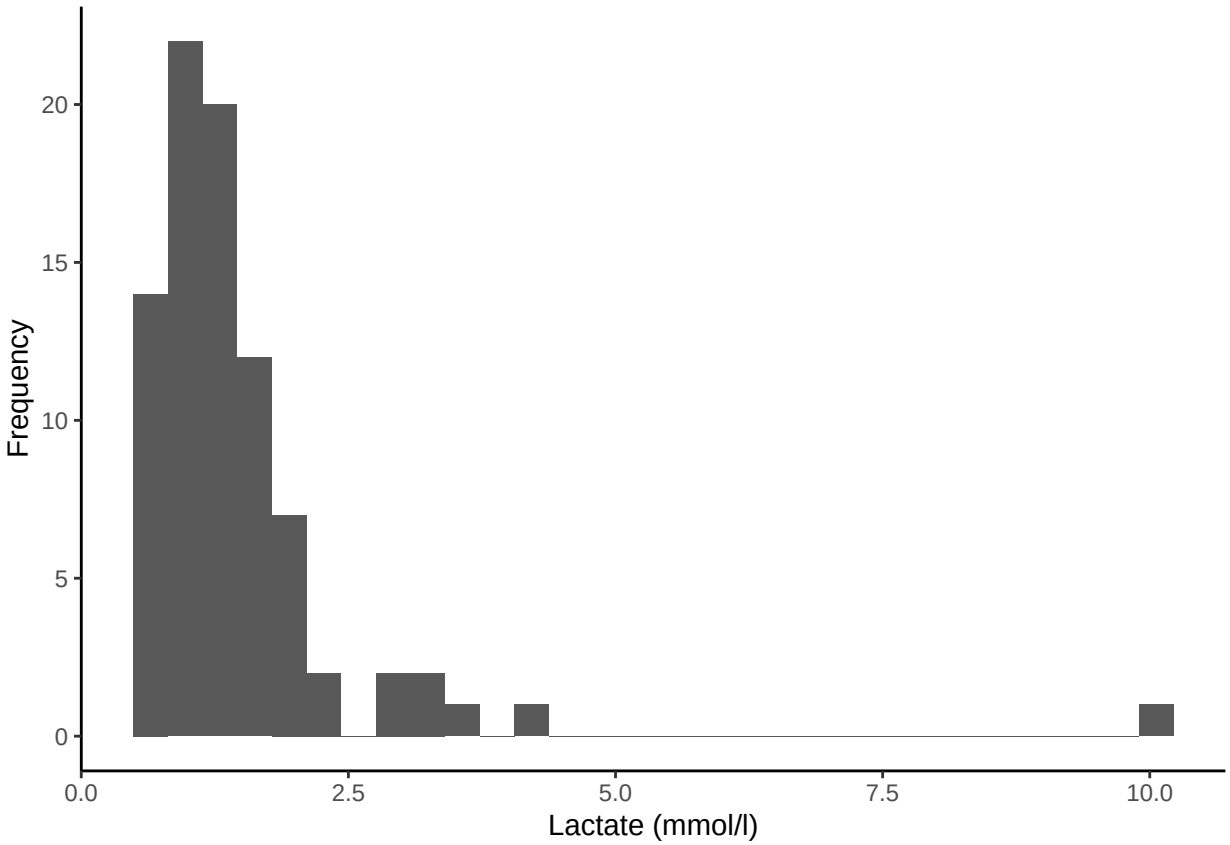
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

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

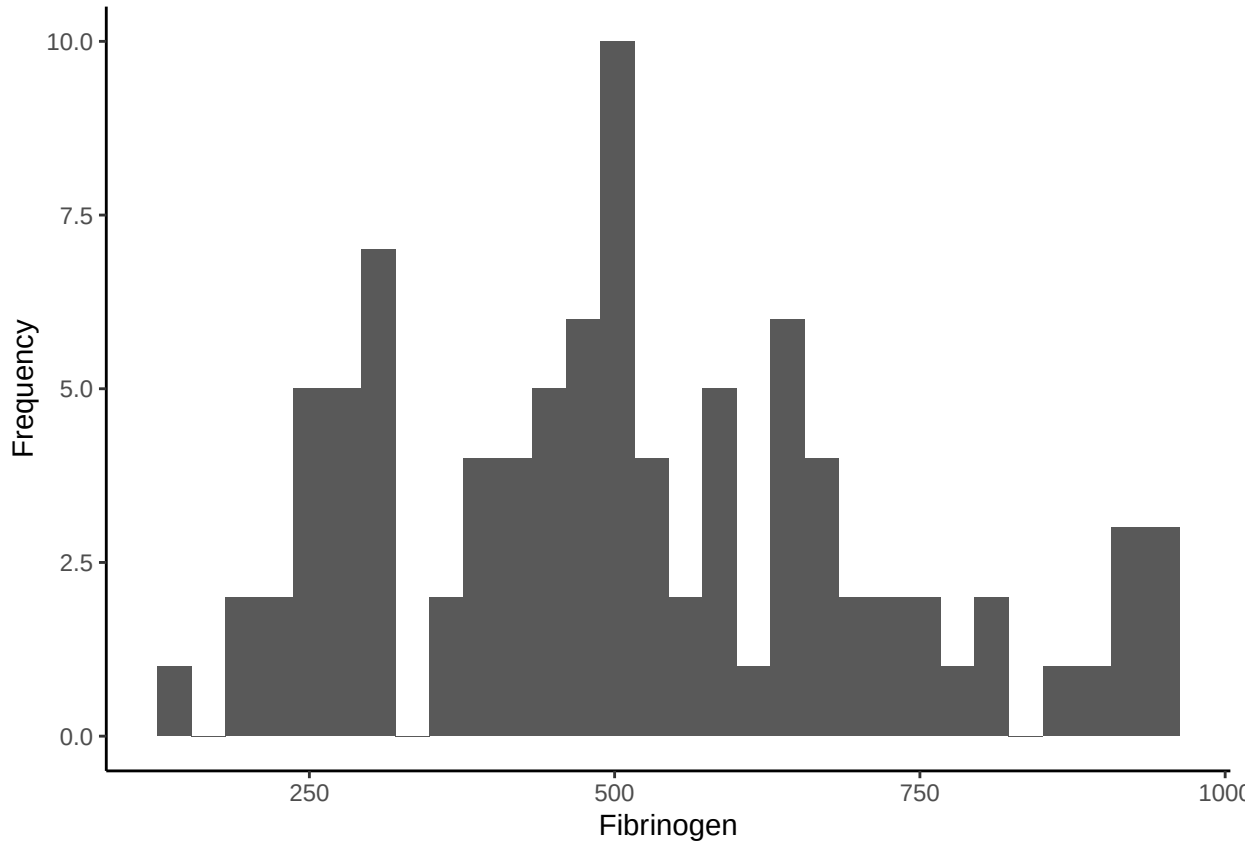
```
## Warning: NAs introduced by coercion
```




```
## Warning: Use of ‘final_comp[[index]]‘ is discouraged.
## i Use ‘.data[[index]]‘ instead.

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

## Warning: Removed 41 rows containing non-finite outside the scale range (‘stat_bin()‘).
## NAs introduced by coercion
```




```
## Warning: Use of `final_comp[[index]]` is discouraged.
## i Use `.data[[index]]` instead.

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

## Warning: Removed 33 rows containing non-finite outside the scale range
## (`stat_bin()`).
```




```
#-----------------PART 1: CHECKING VALUES FOR CONT VAR FOR VENTILATED PATIENTS--------------
#double checking my values with the table1 function

#lactate non uniformly distributed
ventData$lactate.mmol.l. <- as.double(ventData$lactate.mmol.l.)
```

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

```
ventData$age <- as.double(ventData$age)
ventData$fibrinogen <- as.double(ventData$fibrinogen)
```

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

```
medVar <- median(ventData$lactate.mmol.l., na.rm = TRUE)
# Quartile values
quantileVar <- quantile(ventData$lactate.mmol.l., na.rm = TRUE)

# IQR (Q3 - Q1)
IQRVar <- IQR(ventData$lactate.mmol.l., na.rm = TRUE)

lowerQua <- quantile(ventData$lactate.mmol.l., 1/4, na.rm = TRUE)

higherQua <- quantile(ventData$lactate.mmol.l., 3/4, na.rm = TRUE)
```

```r
#print median and IQR statement
print(paste('Median [IQR] of lactate of age in ventilated patients:',
            round((medVar),digits = 2),
             '[',round((lowerQua), digits = 2), ',', round((higherQua), digits = 2),']'))
```

```
## [1] "Median [IQR] of lactate of age in ventilated patients: 1.3 [ 0.92 , 1.65 ]"
```

```r
#for age and fibrinogen which are uniformly distributed
print(paste('Mean (sd) of age in ventilated patients: ',
            round(mean(ventData$age),digits = 2),' (',
             round(sd(ventData$age),digits = 2),')'))
```

```
## [1] "Mean (sd) of age in ventilated patients:  61.16  ( 14.32 )"
```

```r
print(paste('Mean (sd) of fibrinogen in ventilated patients: ',
            round(mean(ventData$fibrinogen, na.rm=TRUE),digits = 2),' (',
             round(sd(ventData$fibrinogen, na.rm=TRUE),digits = 2),')'))
```

```
## [1] "Mean (sd) of fibrinogen in ventilated patients:  527.72  ( 210.24 )"
```

```r
#-----------------PART 1: CHECKING VALUES FOR CONT VAR FOR NONVENTILATED PATIENTS--------------

nonventData$lactate.mmol.l. <- as.double(nonventData$lactate.mmol.l.)
```

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

```r
nonventData$age <- as.double(nonventData$age)
```

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

```r
nonventData$fibrinogen <- as.double(nonventData$fibrinogen)
```

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

```r
#lactate non uniformly distributed
nonventData$lactate.mmol.l. <- as.double(nonventData$lactate.mmol.l.)

medVar <- median(nonventData$lactate.mmol.l., na.rm = TRUE)
# Quartile values
quantileVar <- quantile(nonventData$lactate.mmol.l., na.rm = TRUE)

# IQR (Q3 - Q1)
IQRVar <- IQR(nonventData$lactate.mmol.l., na.rm = TRUE)

lowerQua <- quantile(nonventData$lactate.mmol.l., 1/4, na.rm = TRUE)

higherQua <- quantile(nonventData$lactate.mmol.l., 3/4, na.rm = TRUE)

#print median and IQR statement
print(paste('Median [IQR] of lactate in non-ventilated patients:',round((medVar),digits = 2),
             '[',round((lowerQua), digits = 2), ',', round((higherQua), digits = 2),']'))
```

```
## [1] "Median [IQR] of lactate in non-ventilated patients: 1.17 [ 0.87 , 1.49 ]"
```

```
#for age and fibrinogen which are uniformly distributed
nonventData$age <- as.double(nonventData$age)
nonventData$fibrinogen <- as.double(nonventData$fibrinogen)


print(paste('Mean (sd) of age in non-ventilated patients: ',
            round(mean(nonventData$age, na.rm=TRUE),digits = 2),' (',
             round(sd(nonventData$age, na.rm=TRUE),digits = 2),')'))
```

```
## [1] "Mean (sd) of age in non-ventilated patients:  61.29  ( 17.19 )"
```

```
print(paste('Mean (sd) of fibrinogen in non-ventilated patients: ',
            round(mean(nonventData$fibrinogen, na.rm=TRUE),digits = 2),'
            (', round(sd(nonventData$fibrinogen, na.rm=TRUE),digits = 2),')'))
```

```
## [1] "Mean (sd) of fibrinogen in non-ventilated patients:  501.53  \n            ( 192.83 )"
```

```
#-------------------PART 1: CHECKING CATEGORICAL DATA---------------------------------
data.frame('n' = c(table(ventData$sex)),
          'percent' = c(round(table(ventData$sex)/(length(ventData$sex))*100,digits = 2)))
```

```
##          n percent
##  female 16   31.37
##  male   35   68.63
```

```
data.frame('n' = c(table(nonventData$sex)),
          'percent' = c(round(table(nonventData$sex)/(length(nonventData$sex))*100,digits = 2)))
```

```
##           n percent
##  female  35   47.30
##  male    38   51.35
##  unknown  1    1.35
```

```
#--------------- PART 1: for ICU status-------------------------
data.frame('n' = c(table(ventData$icu_status)),
          'percent' = c(round(table(ventData$icu_status)/(length(ventData$icu_status))*100,digits = 2)))
```

```
##       n percent
##  no   5     9.8
##  yes 46    90.2
```

```
data.frame('n' = c(table(nonventData$icu_status)),
          'percent' = c(round(table(nonventData$icu_status)/(length(nonventData$icu_status))*100,digits
```

```
##       n percent
##  no  54   72.97
##  yes 20   27.03
```

```r
#------------------------GENERATING TABLE 1-----------------------------
#https://einsteinmed.edu/uploadedfiles/centers/ictr/new/p3-r4-table-1-package-in-r.pdf
#https://cran.r-project.org/web/packages/table1/vignettes/table1-examples.html


table1Data <- data.frame(final_comp$mechanical_ventilation, #stratifying
                         final_comp$lactate.mmol.l.,
                         final_comp$age,
                         final_comp$fibrinogen,
                         final_comp$sex, #categorical
                         final_comp$icu_status) #categorical



#renaming the column names so the table1 output will look nice
colnames(table1Data) = c('mech_vent', 'Lactate','Age','Fibrinogen','Sex','icu_status')

#have to change data type because otherwise numeric columns appear as chr
table1Data$Lactate <- as.double(table1Data$Lactate)
table1Data$Fibrinogen <- as.double(table1Data$Fibrinogen)


#renaming some of the values for nice output
table1Data$mech_vent <-
  factor(table1Data$mech_vent,
         levels=c(' no',' yes'),
         labels=c("No Mechanical Ventilation", # Reference
                  "Mechanical Ventilation"))

table1Data$icu_status <-
  factor(table1Data$icu,
         levels=c(' no',' yes'),
         labels=c("No ICU stay", # Reference
                  "ICU"))


#https://stackoverflow.com/questions/73965608/remove-the-mean-row-from-table1-function-in-r
#https://github.com/benjaminrich/table1/issues/126


label(table1Data$Age) <- "Age (years)"
label(table1Data$Lactate) <- "Lactate (mmol/l)"
label(table1Data$Fibrinogen) <- "Fibrinogen (mg/dL)"
label(table1Data$Sex) <- "Sex"
label(table1Data$icu_status) <- "ICU Status"

head(table1Data)
```

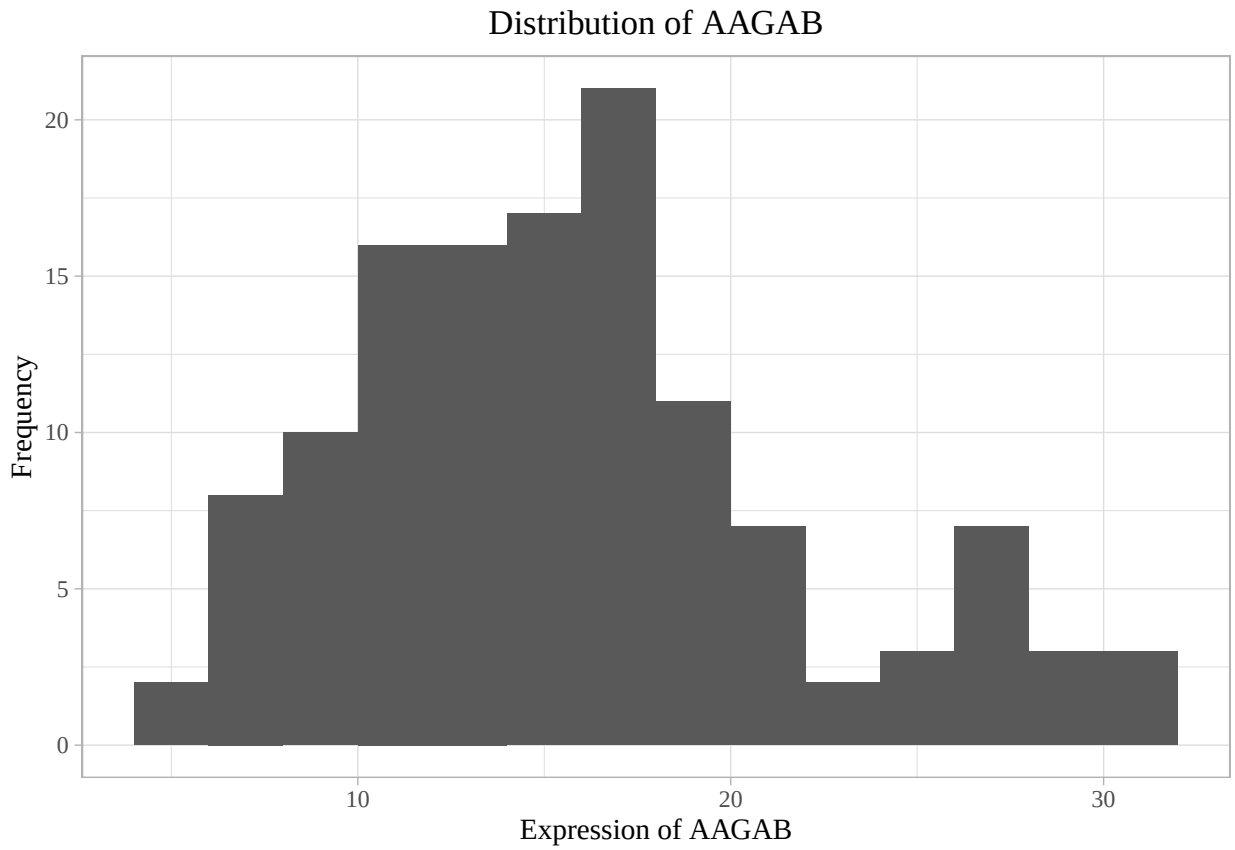
```
##                   mech_vent Lactate Age Fibrinogen     Sex  icu_status
## 1    Mechanical Ventilation    0.90  39        513    male No ICU stay
## 2 No Mechanical Ventilation      NA  63         NA    male No ICU stay
## 3 No Mechanical Ventilation      NA  33        513    male No ICU stay
## 4 No Mechanical Ventilation    0.87  49        949    male No ICU stay
```

```
## 5    Mechanical Ventilation    1.48  49        929    male No ICU stay
## 6 No Mechanical Ventilation    1.17  38        478  female No ICU stay
```

```
#creating table 1 stratified by mech_vent
#rendering all continuous data by median and IQR
table1(~ table1Data$Age+
         table1Data$Lactate+
         table1Data$Fibrinogen+
         table1Data$Sex+
         table1Data$icu_status| table1Data$mech_vent, data=table1Data,
         render.continuous=c(.="Median [Q1 &ndash; Q3]"))
```

	No Mechanical Ventilation	Mechanical Ventilation	Overall
	(N=74)	(N=51)	(N=125)
Age (years)			
Median [Q1 – Q3]	62.0 [50.0 – 77.3]	64.0 [53.5 – 71.5]	62.0 [50.5 – 74.0]
Missing	2 (2.7%)	0 (0%)	2 (1.6%)
Lactate (mmol/l)			
Median [Q1 – Q3]	1.17 [0.870 – 1.50]	1.30 [0.915 – 1.65]	1.23 [0.893 – 1.54]
Missing	30 (40.5%)	11 (21.6%)	41 (32.8%)
Fibrinogen (mg/dL)			
Median [Q1 – Q3]	489 [362 – 585]	513 [372 – 667]	490 [364 – 643]
Missing	25 (33.8%)	8 (15.7%)	33 (26.4%)
Sex			
female	35 (47.3%)	16 (31.4%)	51 (40.8%)
male	38 (51.4%)	35 (68.6%)	73 (58.4%)
unknown	1 (1.4%)	0 (0%)	1 (0.8%)
ICU Status			
No ICU stay	54 (73.0%)	5 (9.8%)	59 (47.2%)
ICU	20 (27.0%)	46 (90.2%)	66 (52.8%)

```
#--------------------------PART 2: HISTOGRAM OF AAGAB-------------------------
#plotting histogram of AAGAB
working_genes <- as.data.frame(t(gene_df)) #pivoting dataframe to select AAGAB as a column

ggplot(working_genes, aes(x=AAGAB))+
  geom_histogram(binwidth = 2, boundary = 0)+ #boundary at 0 to group data
  labs(title = 'Distribution of AAGAB', x = 'Expression of AAGAB', y = 'Frequency')+
  theme_light()+
  theme(plot.title = element_text(hjust=0.5),
        text = element_text(family = "serif")) #changing the font
```




```
new_comp <- merge(working_genes, meta_df, by = 'row.names') #merging the two dataframes using the row n
final_comp <- data.frame(new_comp, row.names = 1) #renaming the row names as the participant ID

final_comp$age<- as.numeric(final_comp$age) #converting the age column from characters to numerics
```

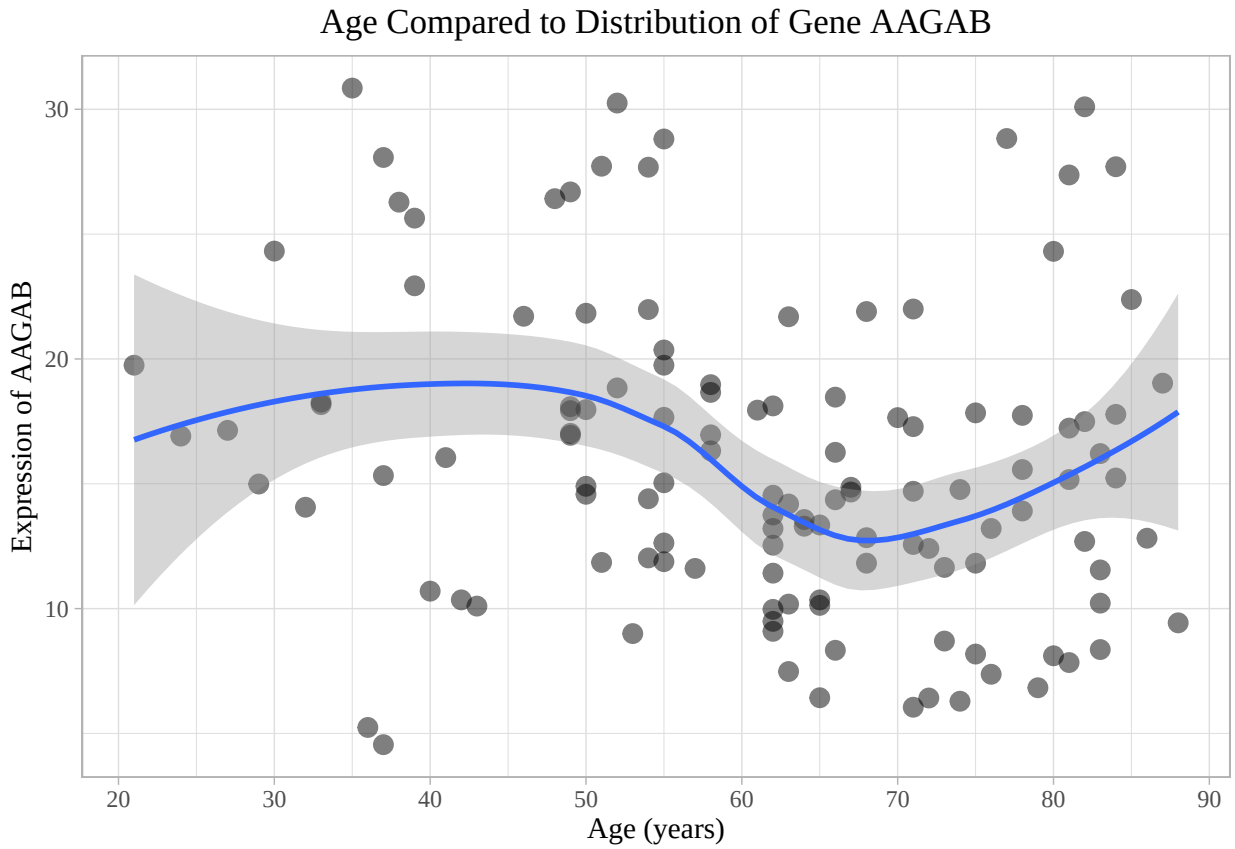
```
## Warning: NAs introduced by coercion
```

```
#-------------------------PART 2: SCATTERPLOT OF AAGAB--------------------------
ggplot(final_comp, aes(x= age, y=AAGAB))+
  #making alpha 0.5, meaning the dots are not fully opaque
  #This allows overlapping datapoints to show up darker than the others
  geom_point(alpha = 0.5, size = 3)+ #two age unknowns from the data
  labs(title = 'Age Compared to Distribution of Gene AAGAB',
       x = 'Age (years)', y = 'Expression of AAGAB')+
  scale_x_continuous(breaks=seq(0,150,by=10))+
  theme_light()+
  geom_smooth()+ #adding trend line
  theme(plot.title = element_text(hjust=0.5),
        text = element_text(family = "serif"))
```

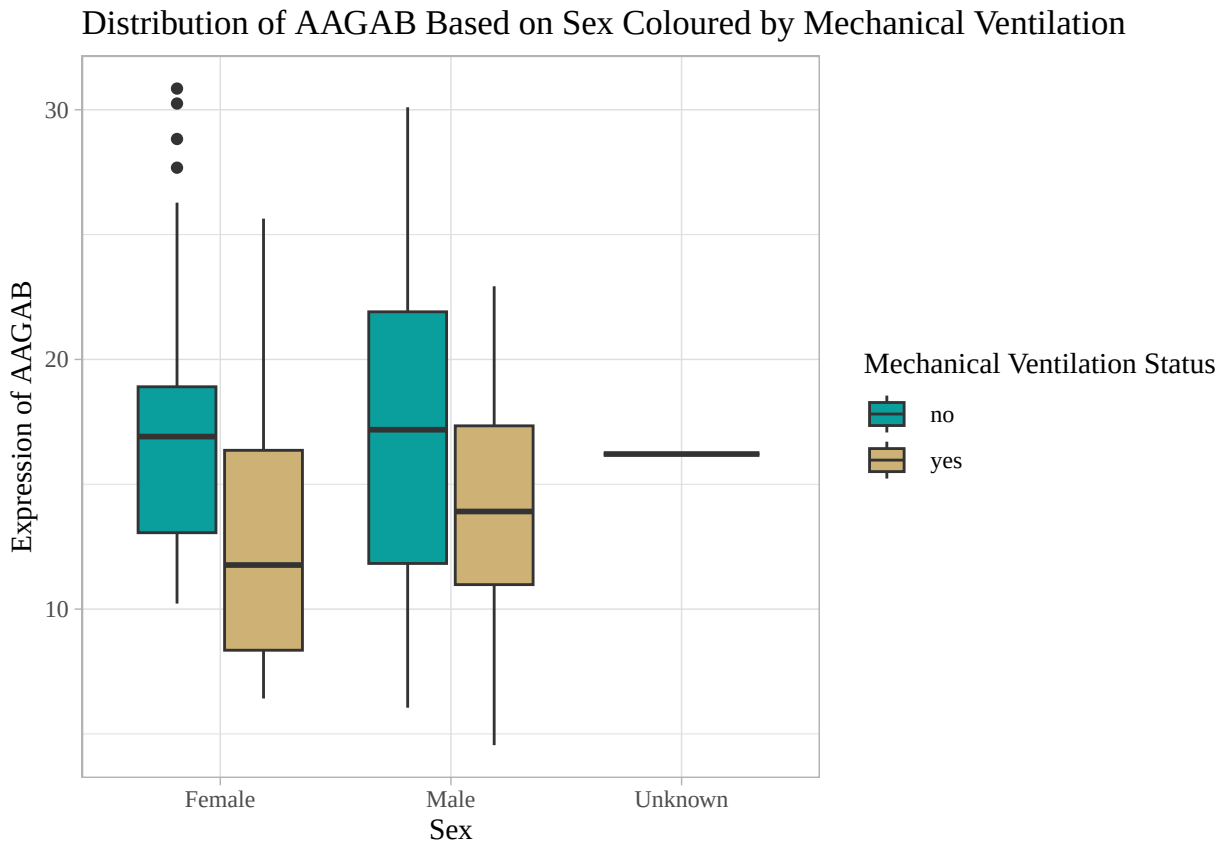
```
## `geom_smooth()` using method = 'loess' and formula = 'y ~ x'
```

```
## 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()`).
```

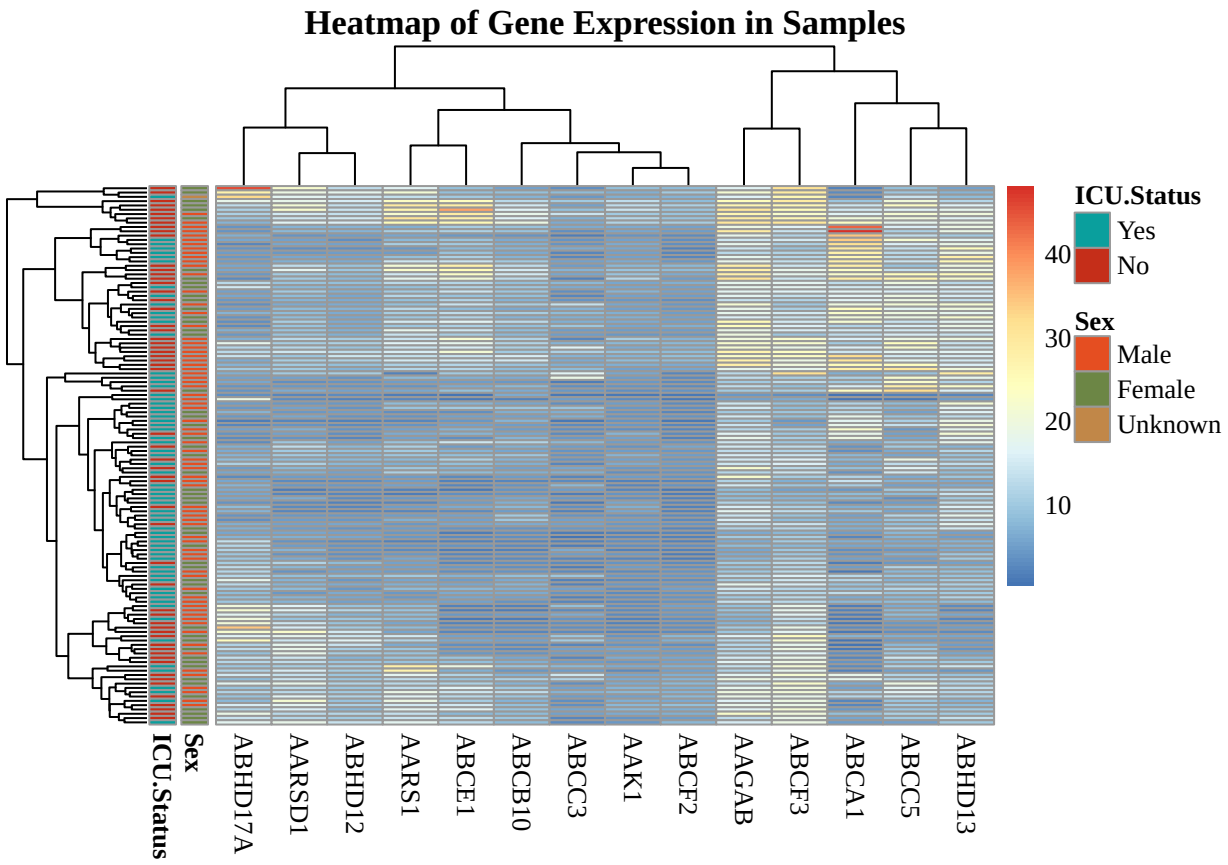



```
#----------------------------PART 2: BOXPLOT OF AAGAB--------------------------

colorPalette<-wesanderson::wes_palette('AsteroidCity1')

ggplot(final_comp, aes (sex, AAGAB, fill = mechanical_ventilation))+
  geom_boxplot()+
  theme_light()+
  labs(title = 'Distribution of AAGAB Based on Sex Coloured by Mechanical Ventilation',
       x = 'Sex', y = 'Expression of AAGAB')+
  theme(plot.title = element_text(hjust=0),
        text = element_text(family = "serif"))+
  scale_fill_manual(values = colorPalette, name = 'Mechanical Ventilation Status')+
  scale_x_discrete(labels = c('Female', 'Male', 'Unknown'))
```




```
#------------------------------PART 3: HEATMAP
annotationData <- data.frame(Sex = final_comp$sex,'ICU Status' = final_comp$icu_status)

#changing the values in the dataframe so the resulting heatmap is clean
annotationData$Sex <- replace(annotationData$Sex,
                              annotationData$Sex == ' female', 'Female')
annotationData$Sex <- replace(annotationData$Sex,
                              annotationData$Sex == ' male', 'Male')
annotationData$Sex <- replace(annotationData$Sex,
                              annotationData$Sex == ' unknown', 'Unknown')

annotationData$ICU.Status <- replace(annotationData$ICU.Status,
                                     annotationData$ICU.Status == ' yes', 'Yes')
annotationData$ICU.Status <- replace(annotationData$ICU.Status,
                                     annotationData$ICU.Status == ' no', 'No')

#equating the row names between the two databases to allow for annotating
rownames(annotationData) <- rownames(final_comp)

#redefining the colours for annotation to fit colour scheme
annotationColors <- list(Sex= c('Male' = '#E54E21',
                                'Female' = "#6C8645",
                                'Unknown' = '#C18748'),
                         ICU.Status = c('Yes' = "#0A9F9D",
                                        'No' = "#C52E19"))
```

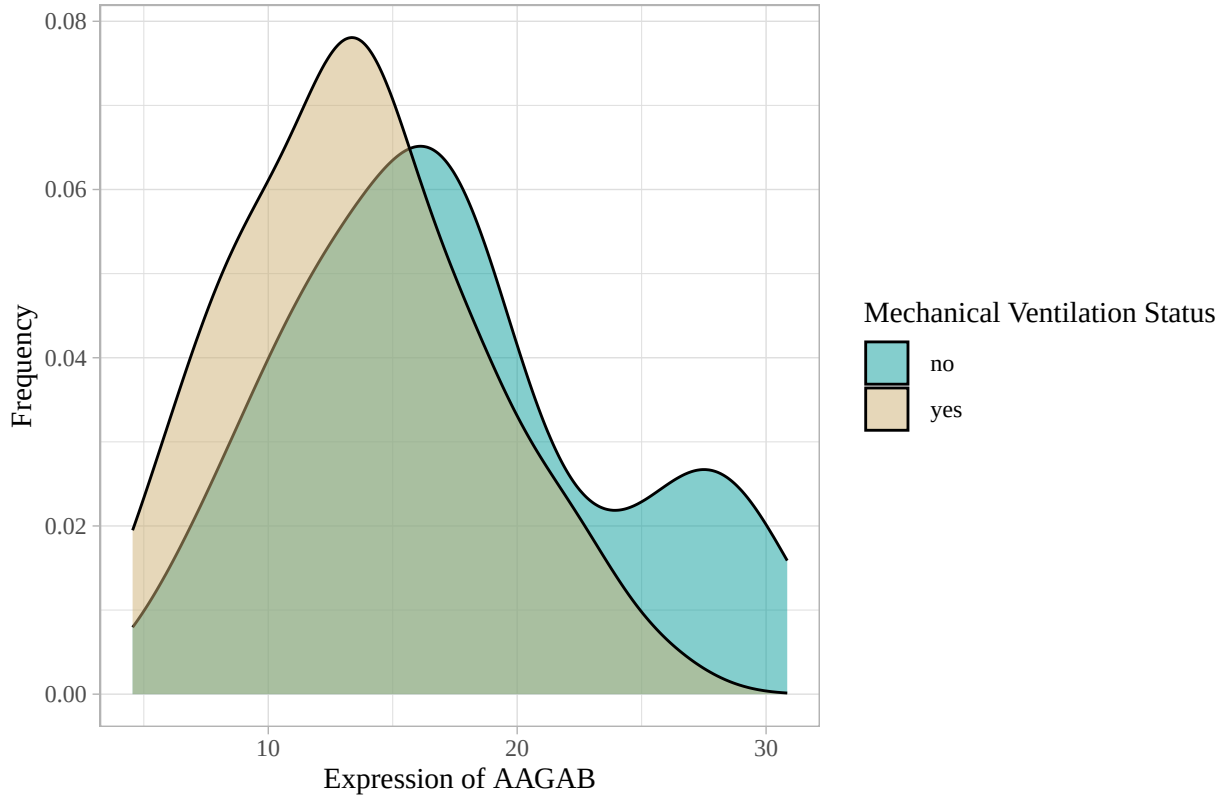
```r
#arbitrarily selecting variety of genes
f2 <-data.frame(final_comp['AAGAB'],
                final_comp['ABCC5'],
                final_comp['AAK1'],
                final_comp['AARS1'],
                final_comp['AARSD1'],
                final_comp['ABCA1'],
                final_comp['ABCB10'],
                final_comp['ABHD13'],
                final_comp['ABHD17A'],
                final_comp['ABCC3'],
                final_comp['ABCF3'],
                final_comp['ABCE1'],
                final_comp['ABCF2'],
                final_comp['ABHD12']
                )

##equating the row names between the two databases to allow for creation of heatmap
rownames(f2) = rownames(final_comp)

#generate heatmap
pheatmap(f2,
         show_rownames = F, #do not want all the sample names
         cluster_rows = T, #clustering both rows and columns
         cluster_cols = T,
         clustering_distance_cols = 'euclidean', #choosing default, euclidean
         clustering_distance_rows = 'euclidean',
         annotation_row = annotationData, #annotating the rows based on our dataframe
         annotation_colors = annotationColors, #using the specified colour scheme
         main = 'Heatmap of Gene Expression in Samples',
         fontfamily = "serif")
```




```
#---------------PART 3: NEW PLOT TYPE-------------------------------
plotData <- data.frame(final_comp['AAGAB'],
                       final_comp['ABHD10'],
                       final_comp['mechanical_ventilation'],
                       final_comp['icu_status'],
                       final_comp['sex'])

colorPalette<-wesanderson::wes_palette('AsteroidCity1')

ggplot(plotData, aes(x=AAGAB, fill = mechanical_ventilation))+
  geom_density(binwidth = 3, alpha = 0.5)+ #not fully opaque to allow vis
  labs(title = 'Expression of AAGAB Stratified by Mechanical Ventilation Status',
       x = 'Expression of AAGAB',
       y = 'Frequency')+
  theme_light()+
  theme(plot.title = element_text(hjust=0.5,),
        text = element_text(family = "serif"))+
  #colouring the curves based on mechanical vent status
  scale_fill_manual(values = colorPalette, name = 'Mechanical Ventilation Status')
```

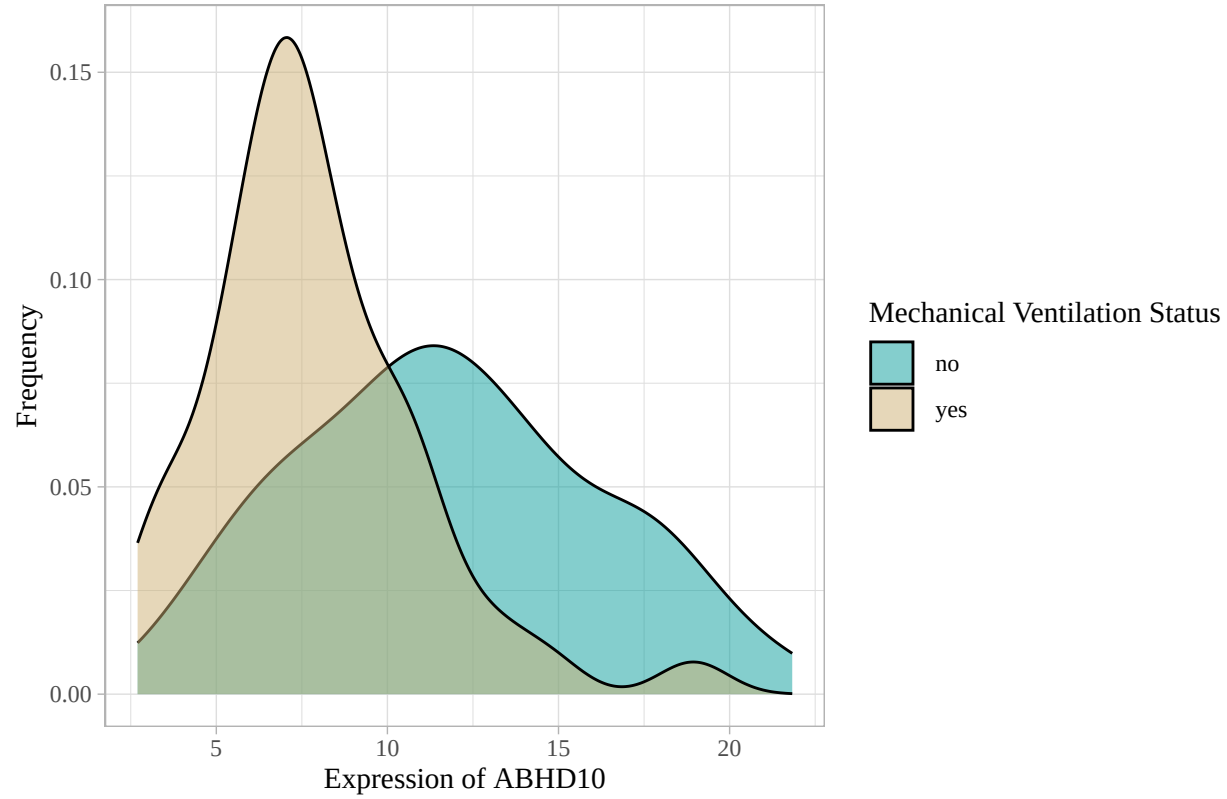
```
## Warning in geom_density(binwidth = 3, alpha = 0.5): Ignoring unknown
## parameters: `binwidth`
```

Expression of AAGAB Stratified by Mechanical Ventilation Status



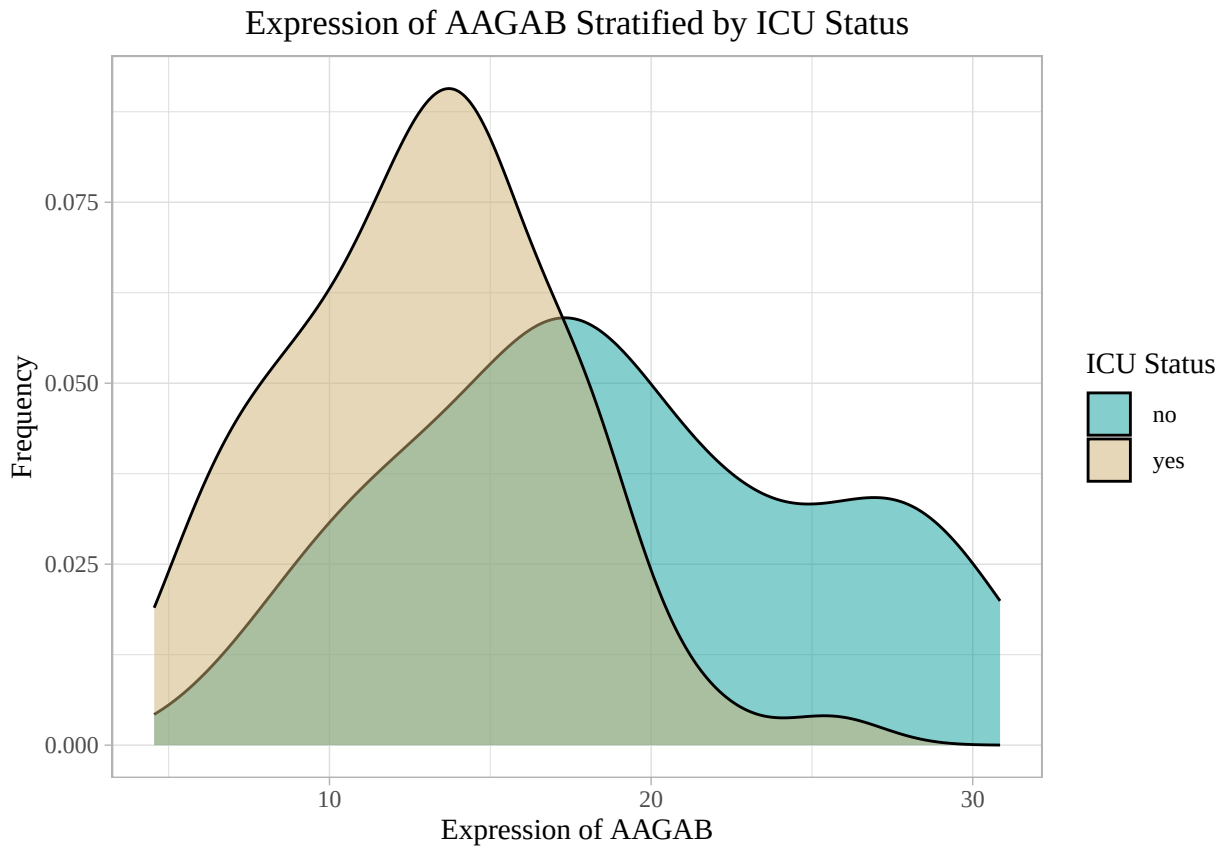

```
ggplot(plotData, aes(x=ABHD10, fill = mechanical_ventilation))+
  geom_density(binwidth = 3, alpha = 0.5)+
  labs(title = 'Expression of ABHD10 Stratified by Mechanical Ventilation Status',
       x = 'Expression of ABHD10',
       y = 'Frequency')+
  theme_light()+
  theme(plot.title = element_text(hjust=0.5,),
        text = element_text(family = "serif"))+
  scale_fill_manual(values = colorPalette, name = 'Mechanical Ventilation Status')
```

```
## Warning in geom_density(binwidth = 3, alpha = 0.5): Ignoring unknown
## parameters: ‘binwidth‘
```




```
ggplot(plotData, aes(x=AAGAB, fill = icu_status))+
  geom_density(binwidth = 3, alpha = 0.5)+ #not fully opaque to allow vis
  labs(title = 'Expression of AAGAB Stratified by ICU Status',
       x = 'Expression of AAGAB',
       y = 'Frequency')+
  theme_light()+
  theme(plot.title = element_text(hjust=0.5,),
        text = element_text(family = "serif"))+
  #colouring the curves based on mechanical vent status
  scale_fill_manual(values = colorPalette, name = 'ICU Status')
```

```
## Warning in geom_density(binwidth = 3, alpha = 0.5): Ignoring unknown
## parameters: ‘binwidth‘
```

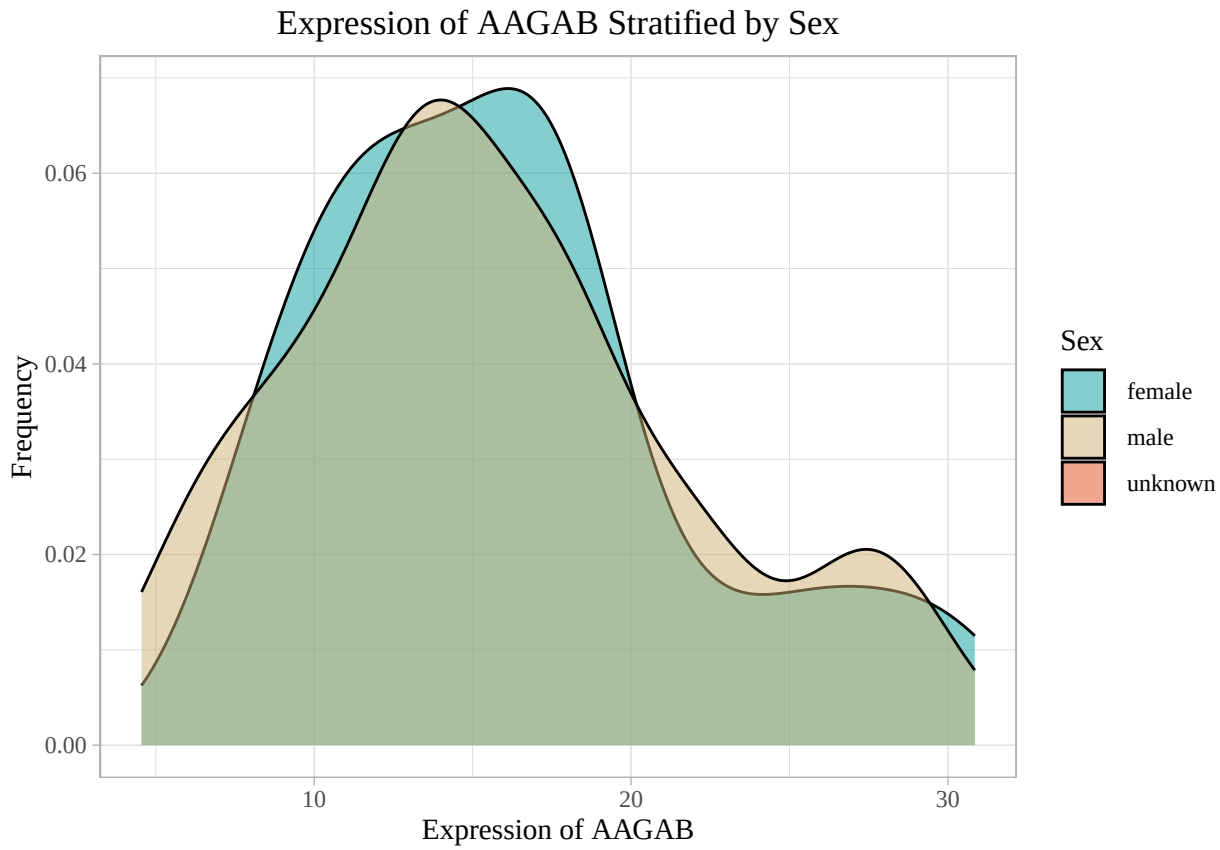



```
ggplot(plotData, aes(x=AAGAB, fill = sex))+
  geom_density(binwidth = 3, alpha = 0.5)+ #not fully opaque to allow vis
  labs(title = 'Expression of AAGAB Stratified by Sex',
       x = 'Expression of AAGAB',
       y = 'Frequency')+
  theme_light()+
  theme(plot.title = element_text(hjust=0.5,),
        text = element_text(family = "serif"))+
  #colouring the curves based on mechanical vent status
  scale_fill_manual(values = colorPalette, name = 'Sex')
```

```
## Warning in geom_density(binwidth = 3, alpha = 0.5): Ignoring unknown
## parameters: ‘binwidth‘
```

```
## Warning: Groups with fewer than two data points have been dropped.
```

```
## Warning in max(ids, na.rm = TRUE): no non-missing arguments to max; returning
## -Inf
```


Expression of AAGAB Stratified by Sex
Frequency
0.06
0.04
0.02
0.00
10
20
30
Expression of AAGAB
Sex
female
male
unknown