Final Project 3

Annie Glenning

2024-08-19

```
setwd("/Users/annieglenning/Documents/Dartmouth/SU24/QBS_103/Data") # set my working directory to where
original_gene_data <- read.table("QBS103_GSE157103_genes.csv", header = TRUE, sep = ",") # reading in t
series_data <- read.table("QBS103_GSE157103_series_matrix.csv", header = TRUE, sep = ",") # reading in
```

```
# transposing the gene data
original_gene_data <- as.data.frame(original_gene_data)
if (is.character(original_gene_data[1, 1])) {
  gene_data <- t(original_gene_data)
  colnames(gene_data) <- gene_data[1, ]
  gene_data <- gene_data[-1, ]
  gene_data <- as.data.frame(gene_data)
}
```

Identify one gene, one continuous covariate, and two categorical covariates

```
AAGAB <- as.numeric(gene_data$AAGAB) # one gene
```

Info for Table of Summary Statistics

sex

```
# percent
length(series_data$age[series_data$sex == ' male'])/length(series_data$age)*100
```

```
## [1] 58.73016
```

```
# percent
length(series_data$age[series_data$sex == ' female'])/length(series_data$age)*100
```

```
## [1] 40.47619
```

```
length(series_data$age[series_data$sex == ' unknown'])/length(series_data$age)*100
```

```
## [1] 0.7936508
```

```r
mean(as.numeric(series_data$age[series_data$sex == ' male']), na.rm = TRUE) # note getting rid of >89
```

```
## Warning in mean(as.numeric(series_data$age[series_data$sex == " male"]), : NAs
## introduced by coercion

## [1] 62.27778
```

```r
sd(as.numeric(series_data$age[series_data$sex == ' male']), na.rm = TRUE)
```

```
## Warning in is.data.frame(x): NAs introduced by coercion

## [1] 14.41168
```

```r
mean(as.numeric(series_data$age[series_data$sex == ' female']), na.rm = TRUE)
```

```
## Warning in mean(as.numeric(series_data$age[series_data$sex == " female"]), :
## NAs introduced by coercion

## [1] 59.3
```

```r
sd(as.numeric(series_data$age[series_data$sex == ' female']), na.rm = TRUE)
```

```
## Warning in is.data.frame(x): NAs introduced by coercion

## [1] 17.92074
```

```r
mean(as.numeric(series_data$age[series_data$sex == ' unknown']), na.rm = TRUE)
```

```
## [1] 83
```

```r
sd(as.numeric(series_data$age[series_data$sex == ' unknown']), na.rm = TRUE)
```

```
## [1] NA
```

```r
mean(as.numeric(series_data$ferritin.ng.ml.[series_data$sex == ' male']), na.rm = TRUE) # 11 unknown
```

```
## Warning in mean(as.numeric(series_data$ferritin.ng.ml.[series_data$sex == : NAs
## introduced by coercion

## [1] 993.3492
```

```r
sd(as.numeric(series_data$ferritin.ng.ml.[series_data$sex == ' male']), na.rm = TRUE)
```

```
## Warning in is.data.frame(x): NAs introduced by coercion

## [1] 1013.052
```

```r
mean(as.numeric(series_data$ferritin.ng.ml.[series_data$sex == ' female']), na.rm = TRUE) # 4 unknown
```

```
## Warning in mean(as.numeric(series_data$ferritin.ng.ml.[series_data$sex == : NAs
## introduced by coercion
```

```
## [1] 619.2766
```

```r
sd(as.numeric(series_data$ferritin.ng.ml.[series_data$sex == ' female']), na.rm = TRUE)
```

```
## Warning in is.data.frame(x): NAs introduced by coercion
```

```
## [1] 1054.329
```

```r
series_data$ferritin.ng.ml.[series_data$sex == ' unknown'] # 1 unknown
```

```
## [1] " unknown"
```

```r
mean(as.numeric(series_data$procalcitonin.ng.ml..[series_data$sex == ' male']), na.rm = TRUE) # 14 unkn
```

```
## Warning in mean(as.numeric(series_data$procalcitonin.ng.ml..[series_data$sex ==
## : NAs introduced by coercion
```

```
## [1] 2.4745
```

```r
sd(as.numeric(series_data$procalcitonin.ng.ml..[series_data$sex == ' male']), na.rm = TRUE)
```

```
## Warning in is.data.frame(x): NAs introduced by coercion
```

```
## [1] 5.793494
```

```r
mean(as.numeric(series_data$procalcitonin.ng.ml..[series_data$sex == ' female']), na.rm = TRUE) # 9 unk
```

```
## Warning in mean(as.numeric(series_data$procalcitonin.ng.ml..[series_data$sex ==
## : NAs introduced by coercion
```

```
## [1] 3.939524
```

```r
sd(as.numeric(series_data$procalcitonin.ng.ml..[series_data$sex == ' female']), na.rm = TRUE)
```

```
## Warning in is.data.frame(x): NAs introduced by coercion
```

```
## [1] 13.65469
```

```r
series_data$procalcitonin.ng.ml..[series_data$sex == ' unknown'] # 1 unknown
```

```
## [1] "unknown"
```

icu_status

```
length(series_data$age[series_data$icu_status == ' yes'])/length(series_data$age) *100
```

```
## [1] 52.38095
```

```
length(series_data$age[series_data$icu_status == ' no'])/length(series_data$age) *100
```

```
## [1] 47.61905
```

```
mean(as.numeric(series_data$age[series_data$icu_status == ' yes']), na.rm = TRUE)
```

```
## [1] 63.45455
```

```
sd(as.numeric(series_data$age[series_data$icu_status == ' yes']), na.rm = TRUE)
```

```
## [1] 13.9958
```

```
mean(as.numeric(series_data$age[series_data$icu_status == ' no'] ), na.rm = TRUE) # gets rid of " :" an
```

```
## Warning in mean(as.numeric(series_data$age[series_data$icu_status == " no"]), :
## NAs introduced by coercion
```

```
## [1] 58.66667
```

```
sd(as.numeric(series_data$age[series_data$icu_status == ' no']), na.rm = TRUE)
```

```
## Warning in is.data.frame(x): NAs introduced by coercion
```

```
## [1] 17.82287
```

```
mean(as.numeric(series_data$ferritin.ng.ml.[series_data$icu_status == ' yes']), na.rm = TRUE)  # 7 unkn
```

```
## Warning in mean(as.numeric(series_data$ferritin.ng.ml.[series_data$icu_status
## == : NAs introduced by coercion
```

```
## [1] 935.322
```

```
sd(as.numeric(series_data$ferritin.ng.ml.[series_data$icu_status == ' yes']), na.rm = TRUE)
```

```
## Warning in is.data.frame(x): NAs introduced by coercion
```

```
## [1] 1019.02
```

```
mean(as.numeric(series_data$ferritin.ng.ml.[series_data$icu_status == ' no']), na.rm = TRUE)  # 9 unkno
```

```
## Warning in mean(as.numeric(series_data$ferritin.ng.ml.[series_data$icu_status
## == : NAs introduced by coercion
```

```
## [1] 715.7451
```

```r
sd(as.numeric(series_data$ferritin.ng.ml.[series_data$icu_status == ' no']), na.rm = TRUE)
```

```
## Warning in is.data.frame(x): NAs introduced by coercion
```

```
## [1] 1067.554
```

```r
mean(as.numeric(series_data$procalcitonin.ng.ml..[series_data$icu_status == ' yes']), na.rm = TRUE)  #
```

```
## Warning in
## mean(as.numeric(series_data$procalcitonin.ng.ml..[series_data$icu_status == :
## NAs introduced by coercion
```

```
## [1] 4.067414
```

```r
sd(as.numeric(series_data$procalcitonin.ng.ml..[series_data$icu_status == ' yes']), na.rm = TRUE)
```

```
## Warning in is.data.frame(x): NAs introduced by coercion
```

```
## [1] 11.98576
```

```r
mean(as.numeric(series_data$procalcitonin.ng.ml..[series_data$icu_status == ' no']), na.rm = TRUE)  # 1
```

```
## Warning in
## mean(as.numeric(series_data$procalcitonin.ng.ml..[series_data$icu_status == :
## NAs introduced by coercion
```

```
## [1] 1.773182
```

```r
sd(as.numeric(series_data$procalcitonin.ng.ml..[series_data$icu_status == ' no']), na.rm = TRUE)
```

```
## Warning in is.data.frame(x): NAs introduced by coercion
```

```
## [1] 5.618867
```

disease_status

```r
length(series_data$age[series_data$disease_status == "disease state: COVID-19"])/length(series_data$age
```

```
## [1] 79.36508
```

```r
length(series_data$age[series_data$disease_status == "disease state: non-COVID-19"])/length(series_data
```

```
## [1] 20.63492
```

```
mean(as.numeric(series_data$age[series_data$disease_status == "disease state: COVID-19"]), na.rm = TRUE
```

```
## Warning in mean(as.numeric(series_data$age[series_data$disease_status == : NAs
## introduced by coercion
```

```
## [1] 60.83673
```

```
sd(as.numeric(series_data$age[series_data$disease_status == "disease state: COVID-19"]), na.rm = TRUE)
```

```
## Warning in is.data.frame(x): NAs introduced by coercion
```

```
## [1] 16.14924
```

```
mean(as.numeric(series_data$age[series_data$disease_status == "disease state: non-COVID-19"]), na.rm =
```

```
## Warning in mean(as.numeric(series_data$age[series_data$disease_status == : NAs
## introduced by coercion
```

```
## [1] 62.8
```

```
sd(as.numeric(series_data$age[series_data$disease_status == "disease state: non-COVID-19"]), na.rm = TRU
```

```
## Warning in is.data.frame(x): NAs introduced by coercion
```

```
## [1] 15.60983
```

```
mean(as.numeric(series_data$ferritin.ng.ml.[series_data$disease_status == "disease state: COVID-19"]),
```

```
## Warning in
## mean(as.numeric(series_data$ferritin.ng.ml.[series_data$disease_status == : NAs
## introduced by coercion
```

```
## [1] 932.7553
```

```
sd(as.numeric(series_data$ferritin.ng.ml.[series_data$disease_status == "disease state: COVID-19"]), na
```

```
## Warning in is.data.frame(x): NAs introduced by coercion
```

```
## [1] 1094.042
```

```
mean(as.numeric(series_data$ferritin.ng.ml.[series_data$disease_status == "disease state: non-COVID-19"
```

```
## Warning in
## mean(as.numeric(series_data$ferritin.ng.ml.[series_data$disease_status == : NAs
## introduced by coercion
```

```
## [1] 250.5
```

```
sd(as.numeric(series_data$ferritin.ng.ml.[series_data$disease_status == "disease state: non-COVID-19"])
```

```
## Warning in is.data.frame(x): NAs introduced by coercion

## [1] 238.208
```

```
mean(as.numeric(series_data$procalcitonin.ng.ml..[series_data$disease_status == "disease state: COVID-1
```

```
## Warning in
## mean(as.numeric(series_data$procalcitonin.ng.ml..[series_data$disease_status ==
## : NAs introduced by coercion

## [1] 3.242989
```

```
sd(as.numeric(series_data$procalcitonin.ng.ml..[series_data$disease_status == "disease state: COVID-19"
```

```
## Warning in is.data.frame(x): NAs introduced by coercion

## [1] 10.44837
```

```
mean(as.numeric(series_data$procalcitonin.ng.ml..[series_data$disease_status == "disease state: non-COVI
```

```
## Warning in
## mean(as.numeric(series_data$procalcitonin.ng.ml..[series_data$disease_status ==
## : NAs introduced by coercion

## [1] 2.119333
```

```
sd(as.numeric(series_data$procalcitonin.ng.ml..[series_data$disease_status == "disease state: non-COVID-
```

```
## Warning in is.data.frame(x): NAs introduced by coercion

## [1] 4.417311
```

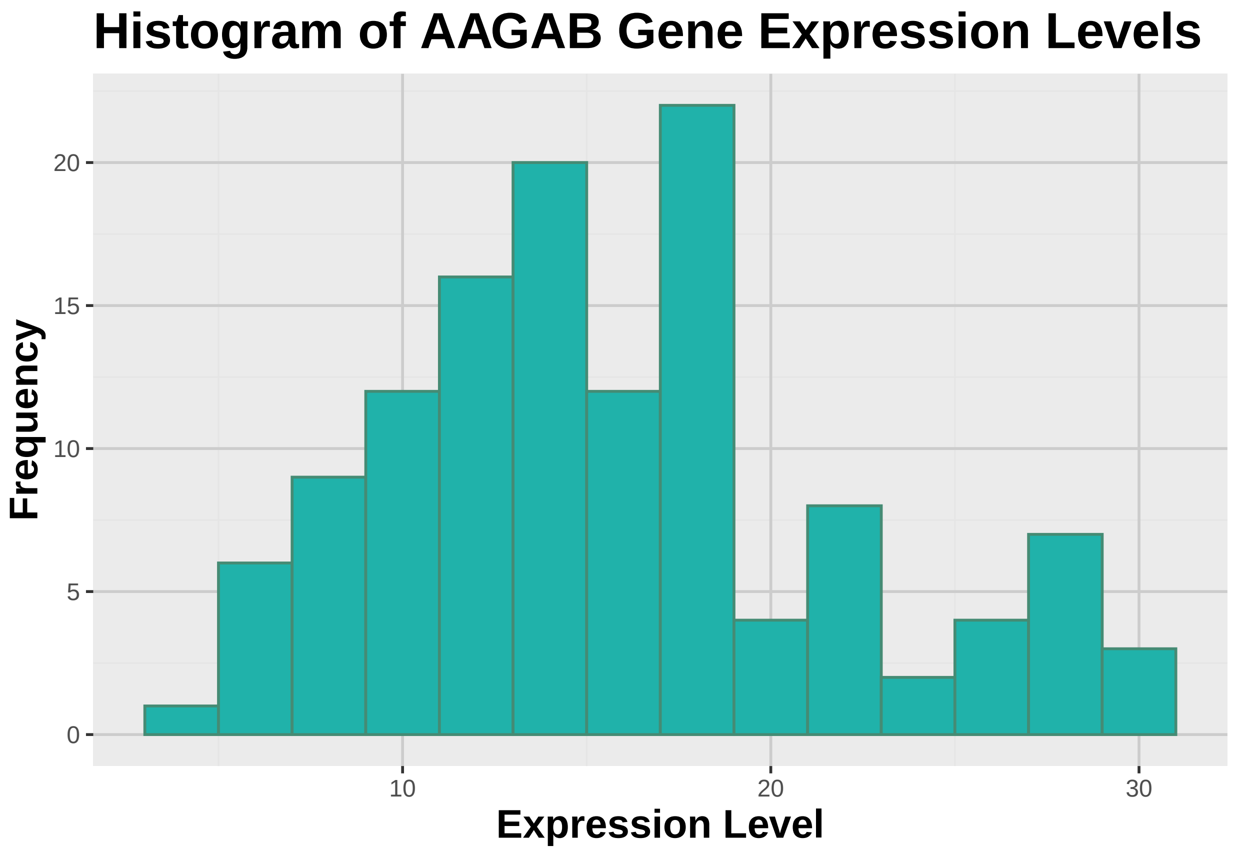
Finalizing Histogram

```
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 error
```

```r
library(ggplot2)
```

```r
ggplot(gene_data, aes(x = as.numeric(AAGAB))) +
  geom_histogram(binwidth = 2, fill = "lightseagreen", color = "aquamarine4") +
  labs( # labeling the title and axis
    title = "Histogram of AAGAB Gene Expression Levels",
    x = "Expression Level",
    y = "Frequency"
  ) +
  theme(
    plot.title = element_text(size = 19, face = "bold"), # title
    axis.title = element_text(size = 15, face = "bold"), # axis
    panel.grid.major = element_line(color = "grey80"), # background
    panel.grid.minor = element_line(color = "grey90")
    )
```

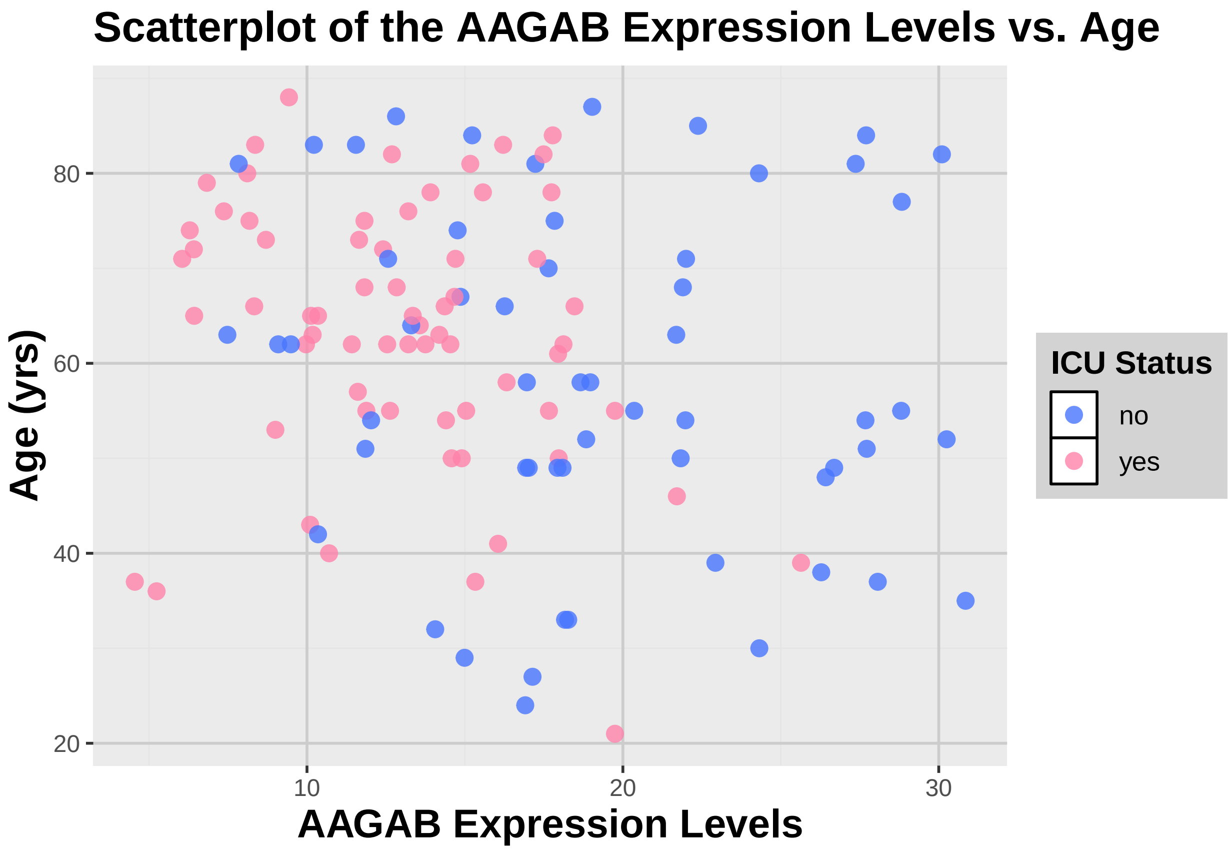


Finalizing Scatterplot

```r
ggplot(series_data, aes(x = as.numeric(AAGAB), y = as.numeric(age),
                        color = factor(icu_status))) + # seperarting the points by icu status
  geom_point(size = 2.5, alpha = 0.8) + # size and transparency of the point
  labs( # labeling the title and axis
    title = "Scatterplot of the AAGAB Expression Levels vs. Age",
    x = "AAGAB Expression Levels",
    y = "Age (yrs)",
    color = "ICU Status" # legend title
    ) +
  scale_color_manual(values = c(" yes" = "palevioletred1", " no" = "royalblue1")) +  # setting the colo
  theme(
    plot.title = element_text(size = 16, face = "bold"),
    axis.title = element_text(size = 15, face = "bold"),
    legend.title = element_text(size = 12, face = "bold"),  # editing the legend
    legend.text = element_text(size = 10),
    legend.background = element_rect(fill = "lightgray", color = NA),
    legend.key = element_rect(fill = "white", color = "black"),
    panel.grid.major = element_line(color = "grey80"), # background
    panel.grid.minor = element_line(color = "grey90")
    )
```

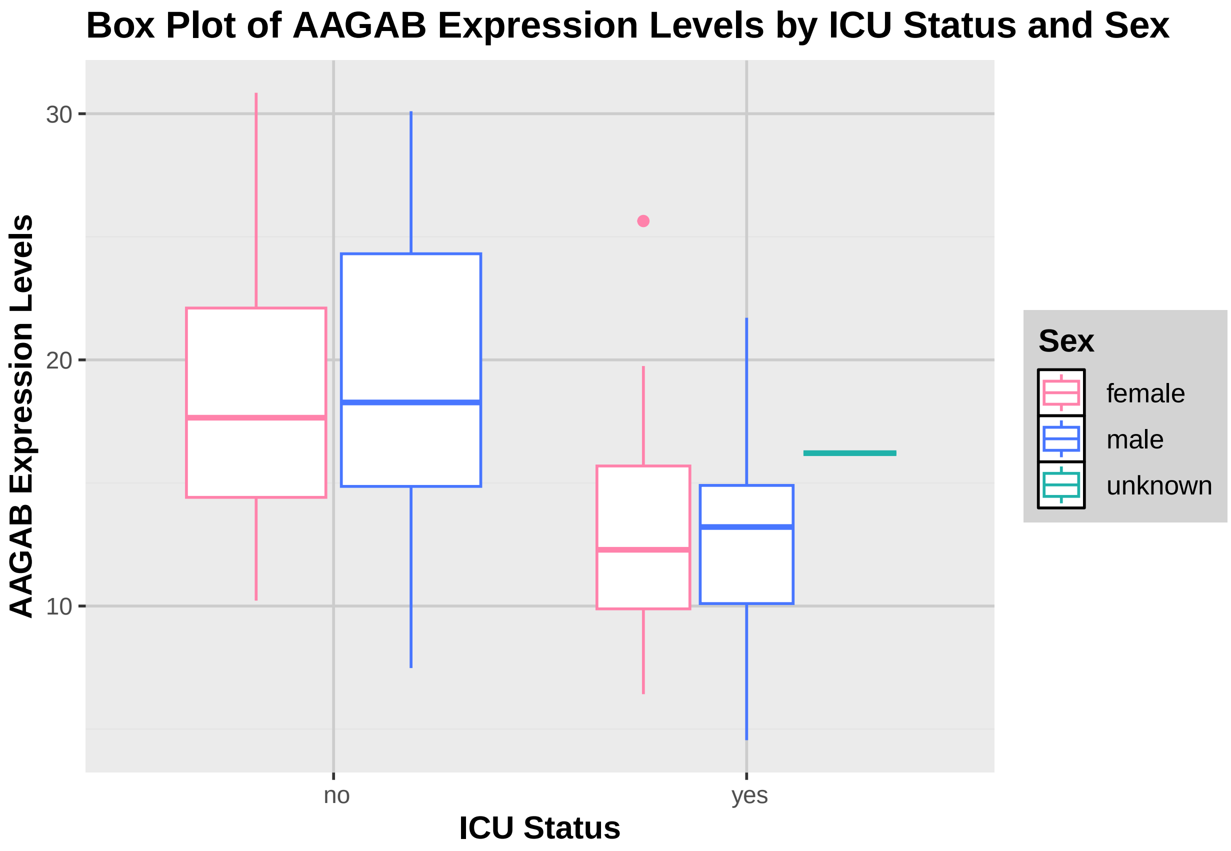
```
## Warning in FUN(X[[i]], ...): NAs introduced by coercion
```

```
## Warning: Removed 3 rows containing missing values or values outside the scale range
## (‘geom_point()‘).
```

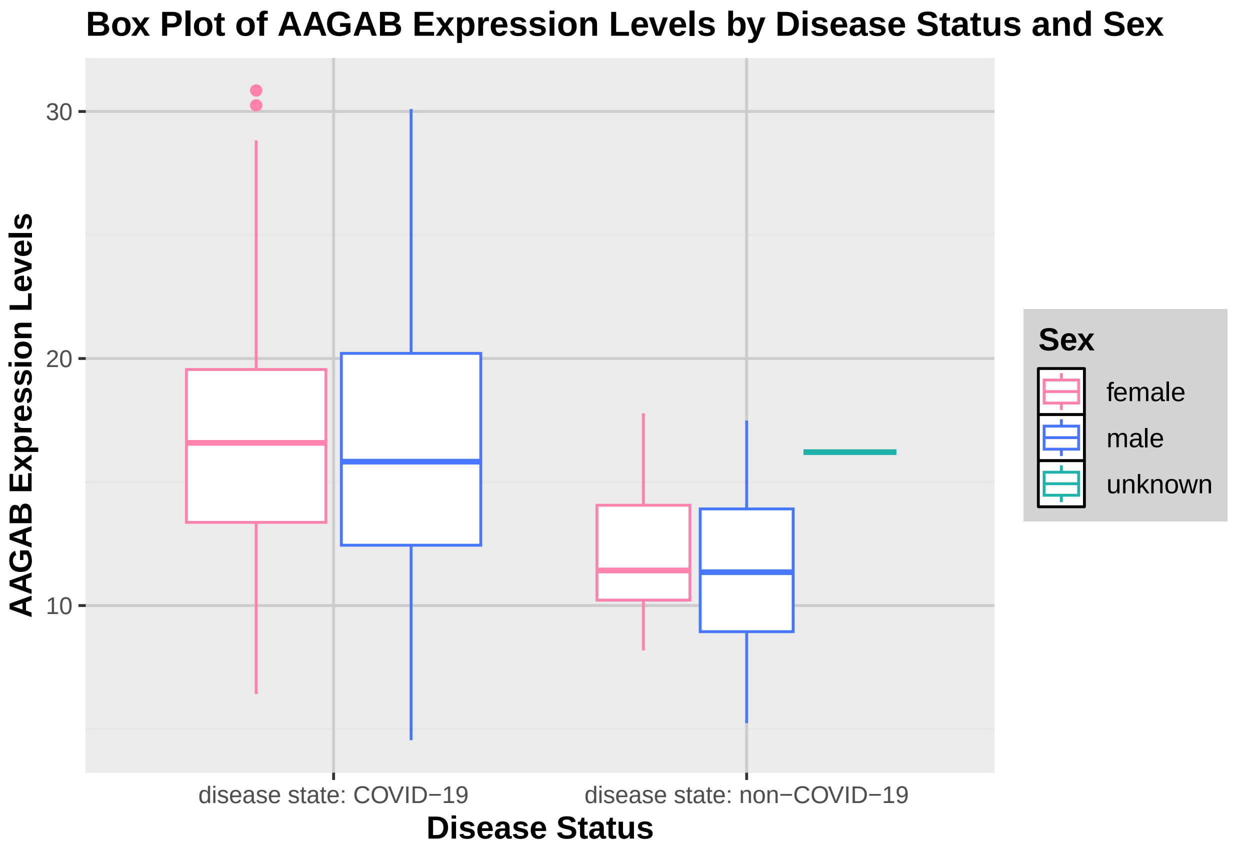


## Finalizing Box Plot

```
ggplot(series_data,aes(x = icu_status, y = as.numeric(AAGAB), color = sex)) +
  geom_boxplot() +
  labs( # labeling the title and axis
    title = "Box Plot of AAGAB Expression Levels by ICU Status and Sex",
    x = "ICU Status",
    y = "AAGAB Expression Levels",
    color = "Sex"
  ) +
  scale_color_manual(values = c(" female" = "palevioletred1", " male" = "royalblue1", " unknown" = "lig
  theme(
    plot.title = element_text(size = 14, face = "bold"),
    axis.title = element_text(size = 12, face = "bold"),
    legend.title = element_text(size = 12, face = "bold"), # editing the legend
    legend.text = element_text(size = 10),
    legend.background = element_rect(fill = "lightgray", color = NA),
    legend.key = element_rect(fill = "white", color = "black"),
    panel.grid.major = element_line(color = "grey80"), # background
    panel.grid.minor = element_line(color = "grey90")
  )
```



```
ggplot(series_data,aes(x = disease_status, y = as.numeric(AAGAB), color = sex)) +
  geom_boxplot() +
  labs( # labeling the title and axis
    title = "Box Plot of AAGAB Expression Levels by Disease Status and Sex",
    x = "Disease Status",
    y = "AAGAB Expression Levels",
    color = "Sex"
  ) +
  scale_color_manual(values = c(" female" = "palevioletred1", " male" = "royalblue1", " unknown" = "ligh
  theme(
    plot.title = element_text(size = 13, face = "bold"),
    axis.title = element_text(size = 12, face = "bold"),
    legend.title = element_text(size = 12, face = "bold"), # editing the legend
    legend.text = element_text(size = 10),
    legend.background = element_rect(fill = "lightgray", color = NA),
    legend.key = element_rect(fill = "white", color = "black"),
    panel.grid.major = element_line(color = "grey80"), # background
    panel.grid.minor = element_line(color = "grey90")
  )
```



New Plot Type

```
#series_data$sex_disease_status <- paste(series_data$sex, series_data$disease_status, sep = " & ")
```

```
# Density plots with semi-transparent fill
#ggplot(series_data, aes(x=as.numeric(age), fill=icu_status)) + geom_density(alpha=.3)
#ggplot(series_data, aes(x=as.numeric(ferritin), fill=disease_status)) + geom_density(alpha=.3)
#ggplot(series_data, aes(x=as.numeric(ferritin), fill=sex)) + geom_density(alpha=.3)
#ggplot(series_data, aes(x=as.numeric(ferritin), fill=sex_disease_status)) + geom_density(alpha=.3)
```

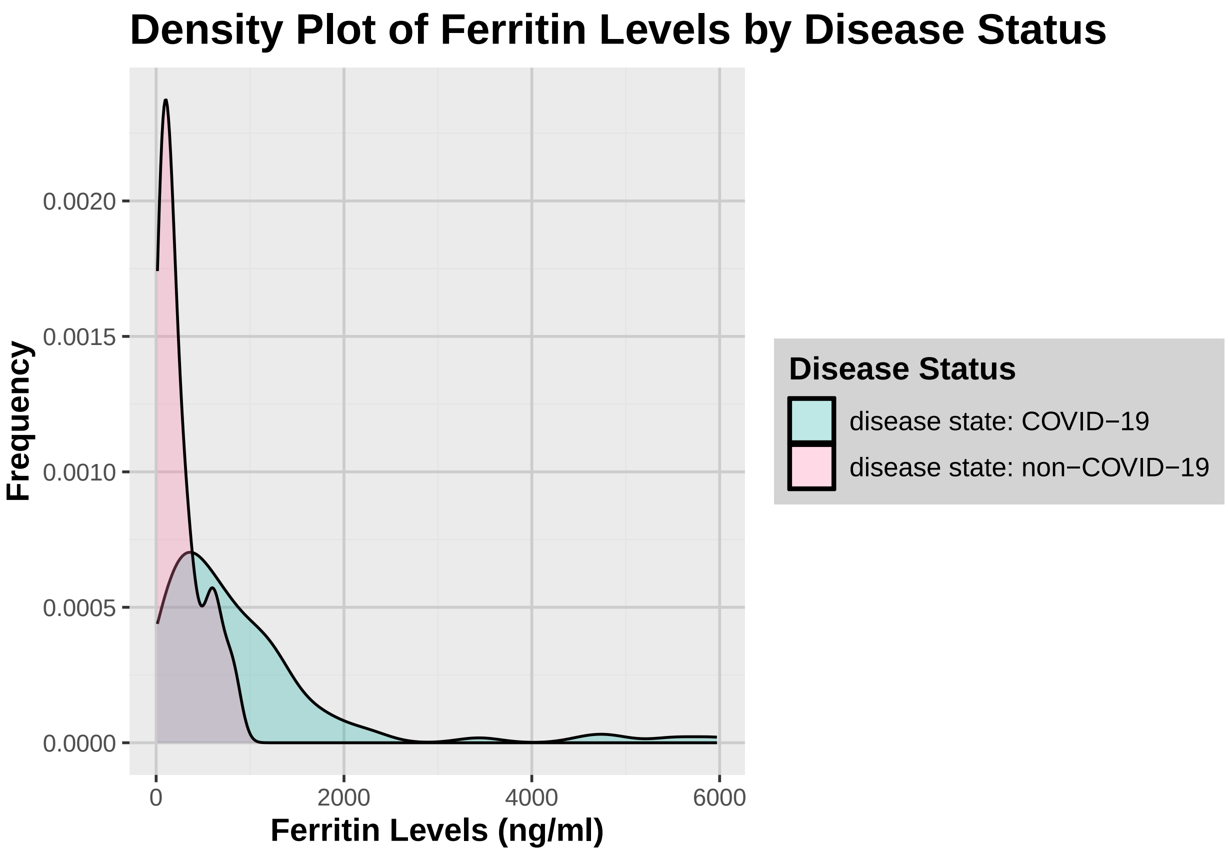
```r
# Density plots with semi-transparent fill
ggplot(series_data, aes(x=as.numeric(series_data$ferritin.ng.ml.), fill=series_data$disease_status)) + g
  labs( # labeling the title and axis
    title = "Density Plot of Ferritin Levels by Disease Status",
    x = "Ferritin Levels (ng/ml)",
    y = "Frequency",
    fill = "Disease Status"
  ) +
  scale_fill_manual(values = c("disease state: COVID-19" = "lightseagreen", "disease state: non-COVID-19
  theme(
    plot.title = element_text(size = 16, face = "bold"),
    axis.title = element_text(size = 12, face = "bold"),
    legend.title = element_text(size = 12, face = "bold"), # editing the legend
    legend.text = element_text(size = 10),
    legend.background = element_rect(fill = "lightgray", color = NA),
    legend.key = element_rect(fill = "white", color = "black"),
    panel.grid.major = element_line(color = "grey80"), # background
    panel.grid.minor = element_line(color = "grey90")
  )
```

```
## Warning in FUN(X[[i]], ...): NAs introduced by coercion
```

```
## Warning: Use of ‘series_data$ferritin.ng.ml.‘ is discouraged.
## i Use ‘ferritin.ng.ml.‘ instead.
```

```
## Warning: Use of ‘series_data$disease_status‘ is discouraged.
## i Use ‘disease_status‘ instead.
```

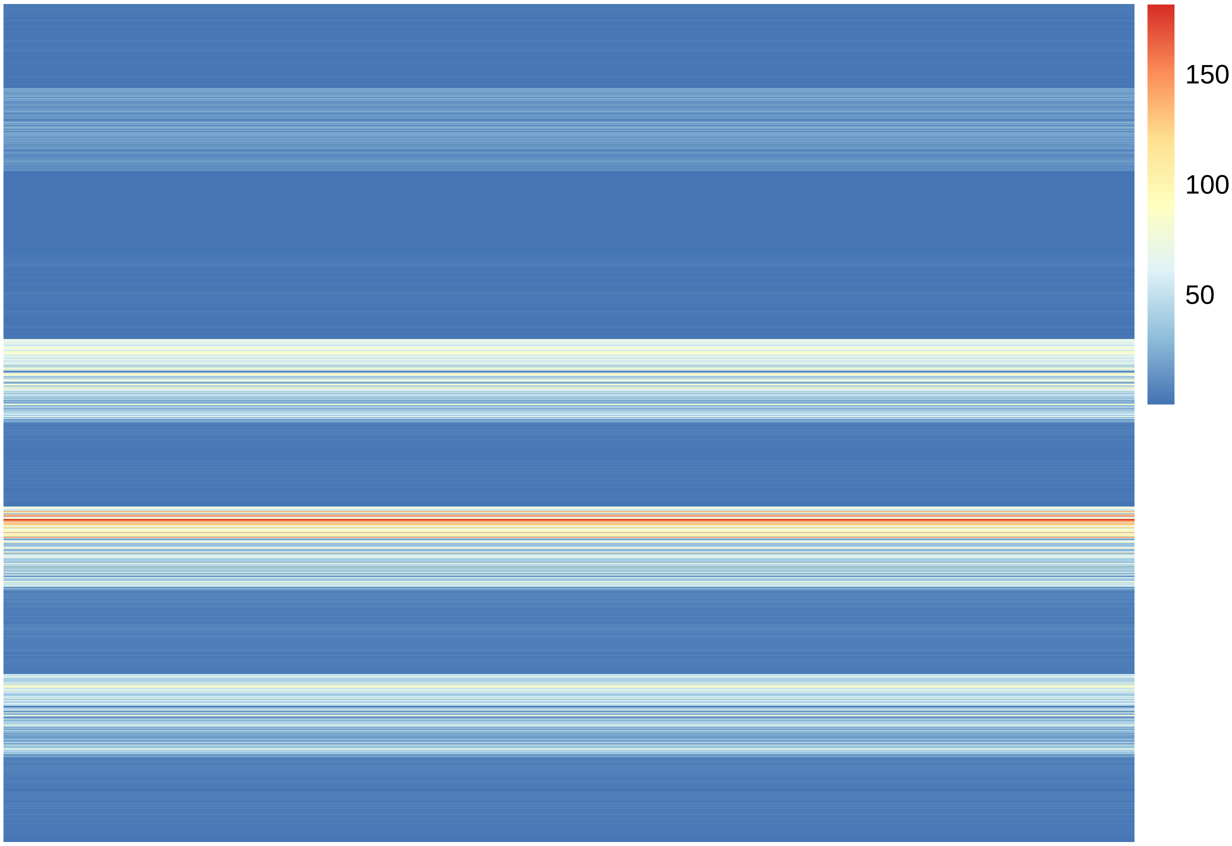
```
## Warning: Removed 16 rows containing non-finite outside the scale range
## (‘stat_density()‘).
```



Heatmap

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

```r
# at least 10 genes
heatmap_genes = c(gene_data$AACS, gene_data$AAGAB, gene_data$A1BG, gene_data$ABI2, gene_data$ABI1, gene_
# generating the heatmap
pheatmap(as.numeric(heatmap_genes),
         cluster_rows = F,
         cluster_cols = F)
```


150
100
50