QBS103_Final_Project

Isabelle Kressy

2025-08-14

Importing packages

```
# call packages
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   3.5.2     v tibble    3.3.0
## v lubridate 1.9.4     v tidyr     1.3.1
## v purrr     1.0.4
## -- 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(reshape2)
```

```
##
## Attaching package: 'reshape2'
##
## The following object is masked from 'package:tidyr':
##
##     smiths
```

```
library(kableExtra)
```

```
##
## Attaching package: 'kableExtra'
##
## The following object is masked from 'package:dplyr':
##
##     group_rows
```

```
library(pheatmap)
```

Importing and tidying data

```r
# set working directory
setwd("~/Desktop/Dartmouth/Foundations of Data Science")

# import csv files
genes_df <- read_csv(file = 'QBS103_GSE157103_genes.csv')
```

```
## New names:
## Rows: 100 Columns: 127
## -- Column specification
## ------------------------------------------------------- Delimiter: "," chr
## (1): ...1 dbl (126): COVID_01_39y_male_NonICU, COVID_02_63y_male_NonICU,
## COVID_03_33y_...
## i Use `spec()` to retrieve the full column specification for this data. i
## Specify the column types or set `show_col_types = FALSE` to quiet this message.
## * `` -> `...1`
```

```r
meta_df <- read_csv(file = 'QBS103_GSE157103_series_matrix-1.csv')
```

```
## Rows: 126 Columns: 25
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (21): participant_id, geo_accession, status, !Sample_submission_date, la...
## dbl  (4): channel_count, charlson_score, ventilator-free_days, hospital-free...
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```r
# combine dataframes and subset for plotting

## tidy genes_df to a format for df merging
genes_df_tidy <- genes_df %>%
  pivot_longer(cols = c(-'...1'), names_to = 'participant_id',
               values_to = 'Gene Expression') %>%
  mutate('Gene' = ...1) %>%
  select(-'...1') %>%
  pivot_wider(values_from = 'Gene Expression', names_from = 'Gene')

## join data frames by participant_id
full_df <- genes_df_tidy %>%
  left_join(meta_df, by = "participant_id")

## tidy continuous variables - ferritin, lactate, age
full_df$`ferritin(ng/ml)`[full_df$`ferritin(ng/ml)` == "unknown"] <- 0
full_df$`ferritin(ng/ml)` <- as.numeric(full_df$`ferritin(ng/ml)`)

full_df$`lactate(mmol/l)`[full_df$`lactate(mmol/l)` == "unknown"] <- 0
full_df$`lactate(mmol/l)` <- as.numeric(full_df$`lactate(mmol/l)`)

full_df$age <- as.numeric(full_df$age)
```

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

```r
## create factors of categorical variables for correct plotting - sex, disease status, icu status
full_df$sex <- factor(full_df$sex, levels = c('female', 'male'))

full_df$disease_status <- factor(full_df$disease_status,
                                 levels = c('disease state: COVID-19',
                                            'disease state: non-COVID-19'))

full_df$icu_status <- factor(full_df$icu_status, levels = c('yes', 'no'))

## remove na values
full_df <- full_df %>% drop_na() %>% na.omit()
```

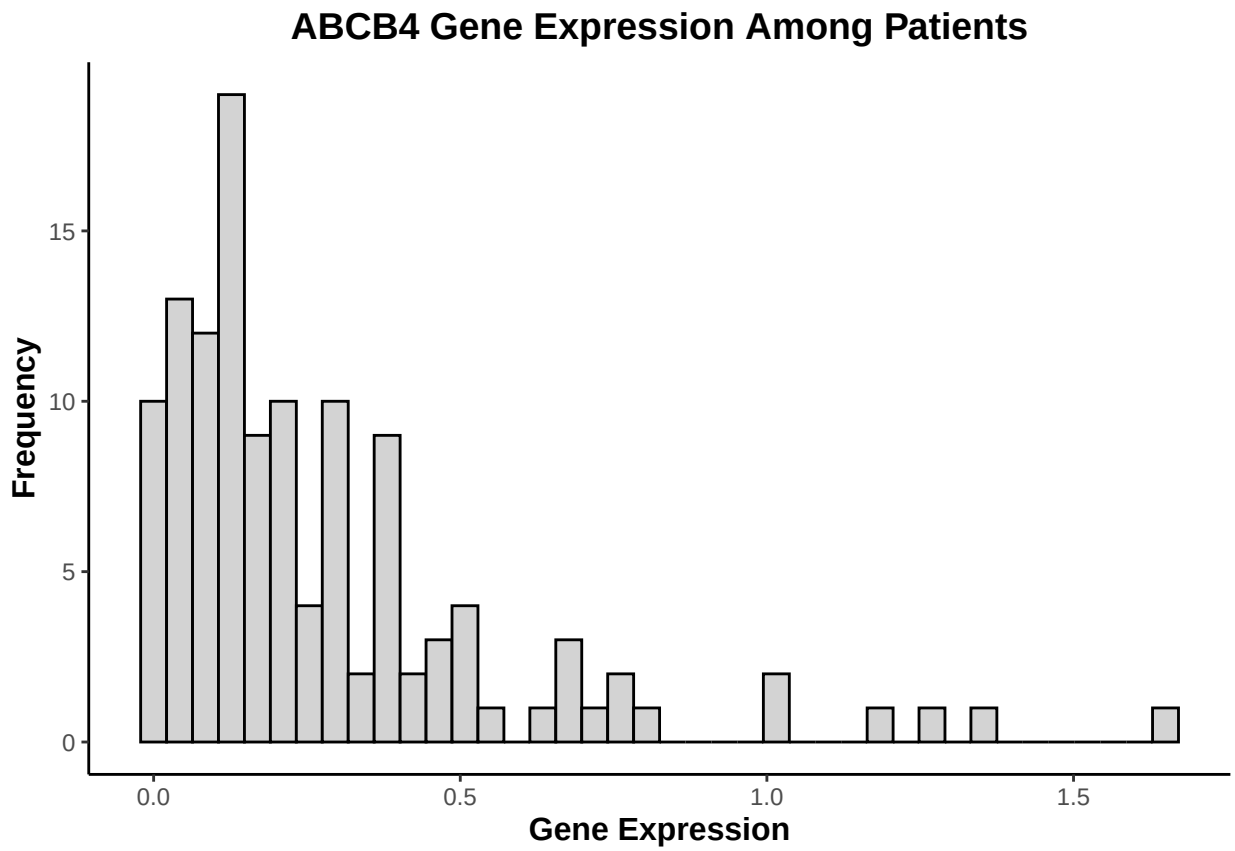
Latex Table Generation. Code is commented to remove outputs.

```r
# calculate summary stats

# check for normality in continuous variables
#hist(full_df$`ferritin(ng/ml)`, main = "Histogram", xlab = "Ferritin") # not normal
#hist(full_df$`lactate(mmol/l)`, main = "Histogram", xlab = "Lactate") # not normal
#hist(full_df$age, main = "Histogram", xlab = "Age") # pretty normal

# calculate mean/SD or median/IQR or % of each variable by gender
# females <- full_df %>%
#   subset(sex == 'female')
# males <- full_df %>%
#   subset(sex == 'male')
#
# round(mean(full_df$age,2))
# round(sd(full_df$age,2))
# round(mean(females$age),2)
# round(sd(females$age),2)
# round(mean(males$age),2)
# round(sd(males$age),2)
#
#
# round(median(full_df$`ferritin(ng/ml)`),2)
# round(quantile(full_df$`ferritin(ng/ml)`),2)
# round(median(females$`ferritin(ng/ml)`),2)
# round(quantile(females$`ferritin(ng/ml)`),2)
# round(median(males$`ferritin(ng/ml)`),2)
# round(quantile(males$`ferritin(ng/ml)`),2)
#
#
# round(median(full_df$`lactate(mmol/l)`),2)
# round(quantile(full_df$`lactate(mmol/l)`),2)
# round(median(females$`lactate(mmol/l)`),2)
# round(quantile(females$`lactate(mmol/l)`),2)
# round(median(males$`lactate(mmol/l)`),2)
# round(quantile(males$`lactate(mmol/l)`),2)
#
# summary(full_df$disease_status == 'disease state: COVID-19')
# summary(females$disease_status == 'disease state: COVID-19')
# summary(males$disease_status == 'disease state: COVID-19')
```

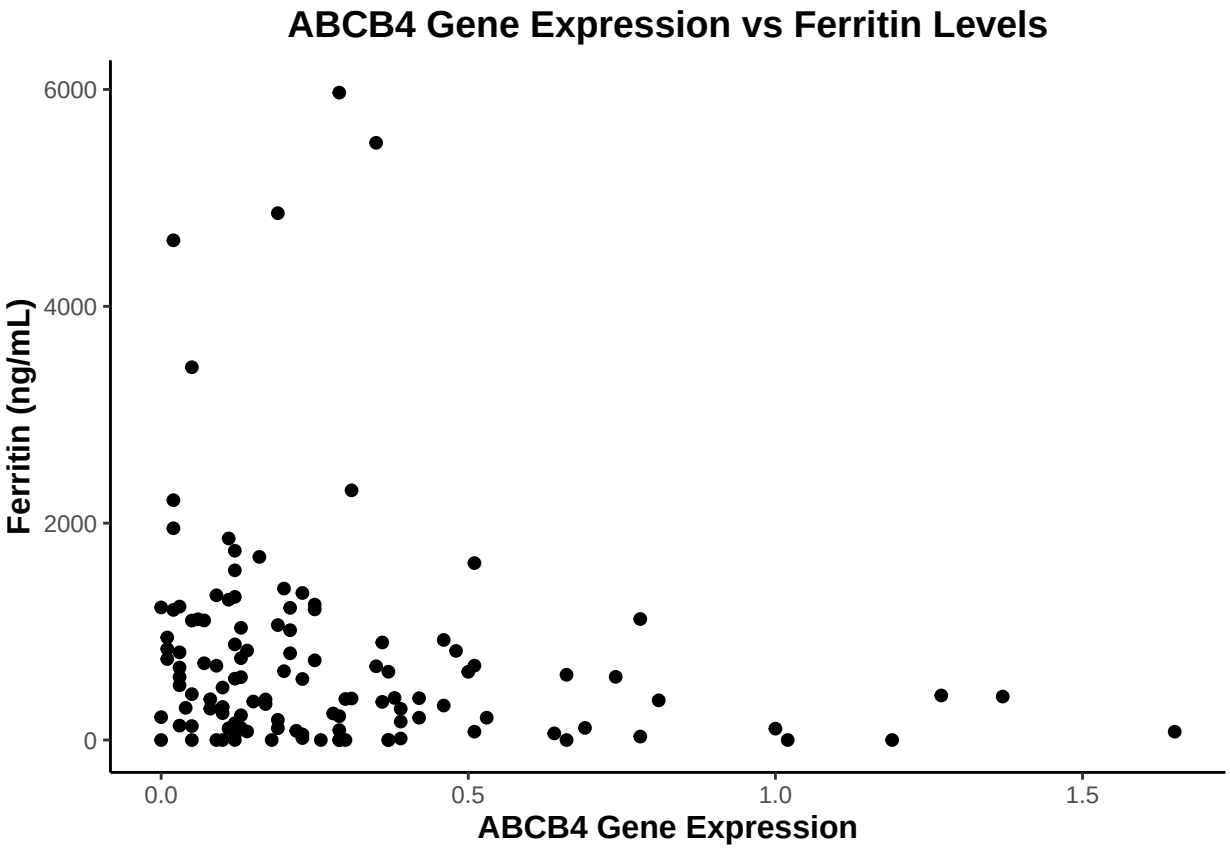
```
#
# summary(full_df$icu_status == 'yes')
# summary(females$icu_status == 'yes')
# summary(males$icu_status == 'yes')
#
#
#
#
# table1 <- data.frame(
#   Variable = c('Age, mean (sd)', 'Ferritin levels, median [IQR]',
#                'Lactate levels, median [IQR]', 'Disease status - COVID, n (%)',
#                'Disease status - non-COVID, n (%)', 'ICU status - Yes, n (%)', 'ICU status - No, n (%
#   Total = c('62.0 (16.0)', '406.0 [111.3, 996.3]', '0.88 [0, 1.35]', '98 (80.3)', '24 (19.7)', '65 (5
#   Female = c('59.3 (17.9)', '302.5 [86.5, 617.5]', '0.81 [0, 1.26]', '37 (74.0)',
#                '13 (26.0)', '24 (48.0)', '26 (52.0)'),
#   Male = c('62.28 (14.41)', '652.0 [217.8, 1201.3]', '1.00 [0, 1.45]', '61 (84.7)',
#            '11 (15.3)', '41 (56.9)', '31 (43.1)')
# )
#
# kable(table1,
#       format = "latex",
#       booktabs = TRUE,
#       col.names = c("Variable", 'Total (n = 22)', 'Female (n = 50)', 'Male (n = 72)'),
#       caption = "Summary Table",
#       align = c("l","l", "l", "l"),
#       escape = TRUE)
```

Histogram, scatterplot, and boxplot from submission 1 of gene ABCB4.

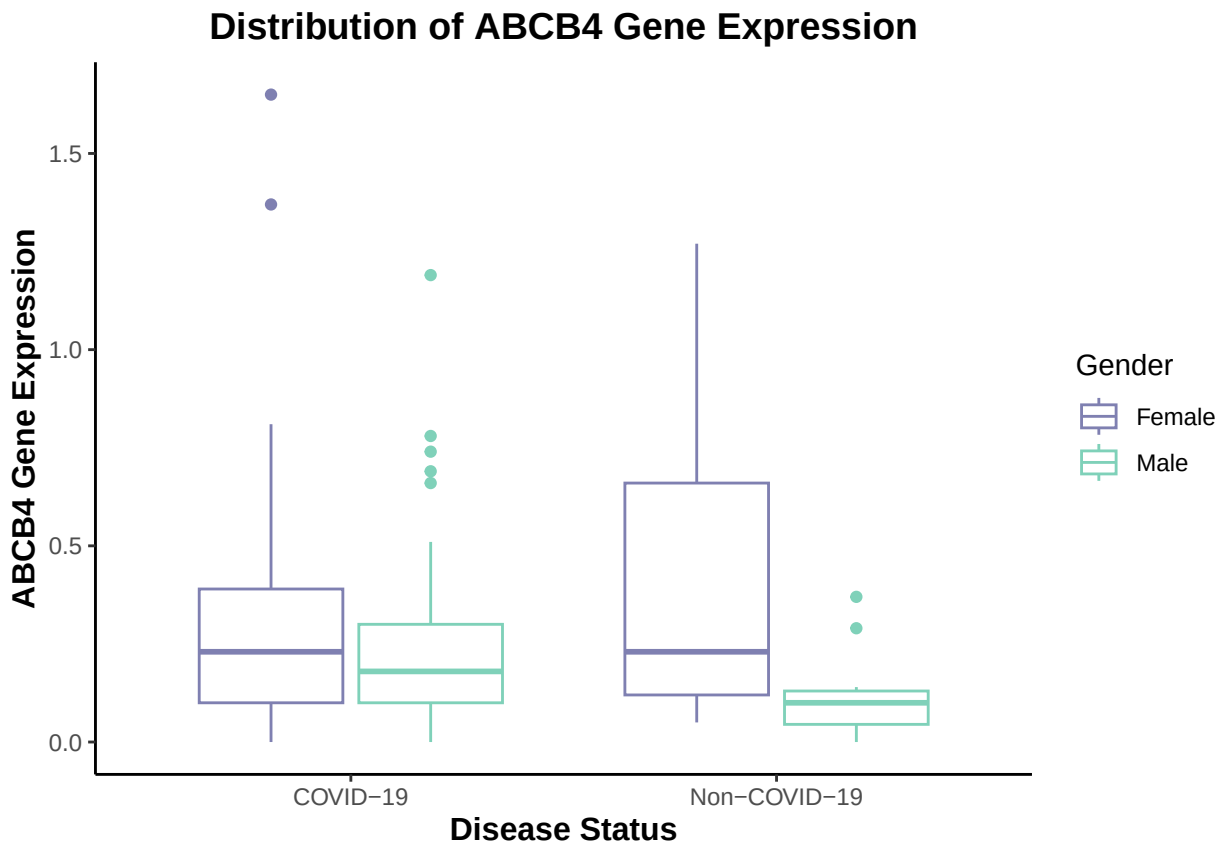
```
# define theme for plots
myTheme <- theme(panel.border = element_blank(),
                 panel.grid.major = element_blank(),
                 panel.grid.minor = element_blank(),
                 axis.line = element_line(colour = "black"),
                 plot.background = element_blank(),
                 panel.background = element_blank(),
                 plot.title = element_text(size = 14, face = 'bold', hjust = 0.5),
                 axis.title.x = element_text(size = 12, face = 'bold'),
                 axis.title.y = element_text(size = 12, face = 'bold'))


# histogram for gene expression
ggplot(data = full_df, aes(x = ABCB4)) +
  geom_histogram(bins = 40, fill = 'lightgrey', color = 'black') +
  labs(title = 'ABCB4 Gene Expression Among Patients',
       x = 'Gene Expression', y = 'Frequency') +
  myTheme
```




```
# scatterplot for gene expression and continuous covariate
ggplot(data = full_df, aes(x = ABCB4, y = `ferritin(ng/ml)`)) +
  geom_point(size = 1.75) +
  labs(title = 'ABCB4 Gene Expression vs Ferritin Levels',
       x = 'ABCB4 Gene Expression', y = 'Ferritin (ng/mL)') +
  myTheme
```




```
# boxplot of gene expression separated by both categorical covariates
ggplot(full_df,aes(x = disease_status, y = ABCB4, color = sex)) +
  geom_boxplot() +
  scale_color_manual(values = c('#7F80B1', '#7FD1B9'),
                     labels = c('Female', 'Male')) +
  labs(x = 'Disease Status', y = 'ABCB4 Gene Expression',
       title = 'Distribution of ABCB4 Gene Expression', color = 'Gender') +
  scale_x_discrete(labels = c('COVID-19', 'Non-COVID-19')) +
  myTheme
```



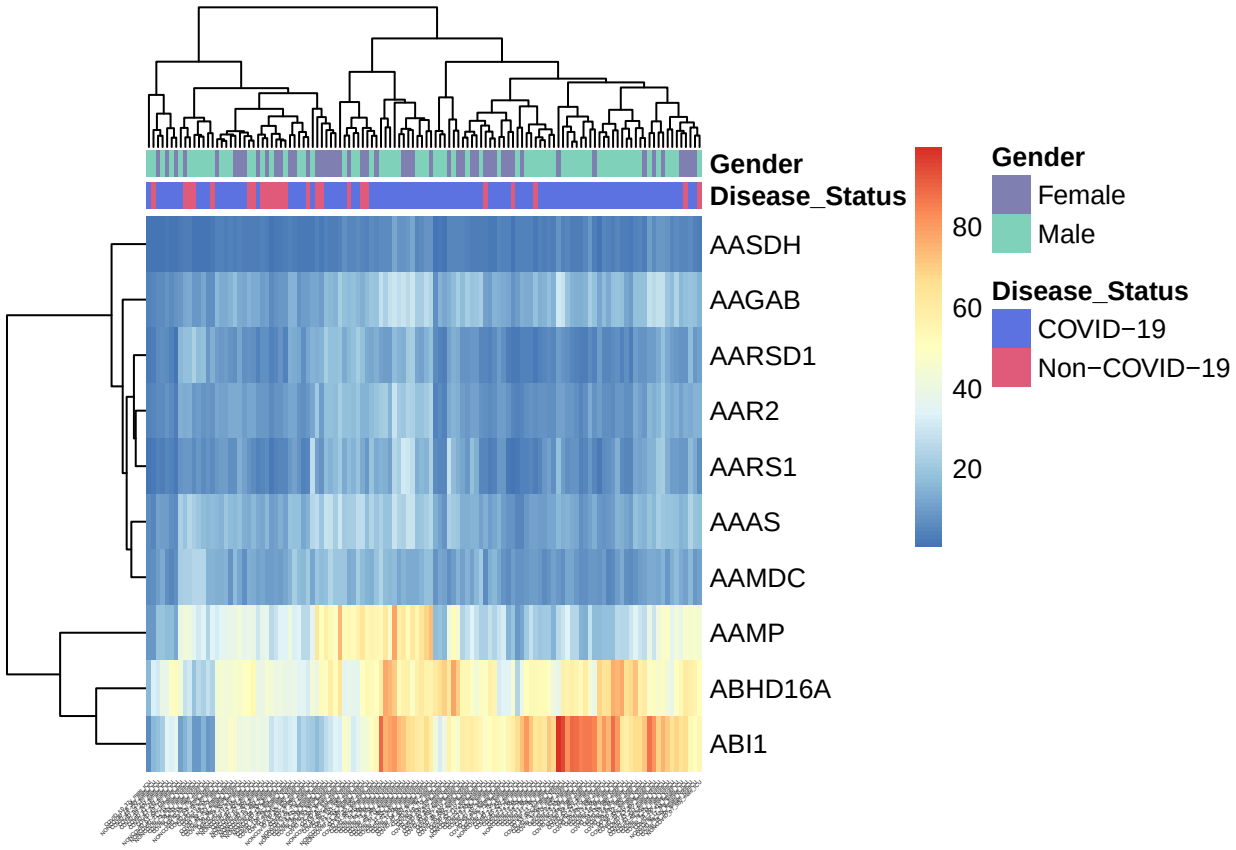
Heatmap of select genes

```
# 2 categorical covariates: sex and disease status
# 10 genes: "AAMP","AANAT","ABCA13","ABCA2","ABCB4","ABCB9","ABHD15","ABHD16A","ABHD8","ABI1"

# subset data for heatmap generation
sub_df <- full_df %>%
  select("AAMP","AAAS","AAGAB","ABHD16A","AAMDC","AAR2","AARS1",
         "AARSD1","AASDH","ABI1","disease_status","sex","participant_id")

genes <- sub_df %>%
  select("AAMP","AAAS","AAGAB","ABHD16A","AAMDC","AAR2","AARS1",
         "AARSD1","AASDH","ABI1")

# ensure rownames of genes df is the same of the participant id's
rownames(genes) <- sub_df$participant_id
```

```
## Warning: Setting row names on a tibble is deprecated.
```

```
# transpose genes for heatmap
genes_t <- t(genes)

# generate annotation data for the heatmap
annotationData <- data.frame(row.names = sub_df$participant_id,
                             'Disease_Status' =
                               ifelse(sub_df$disease_status == "disease state: COVID-19",
```

```
                                        "COVID-19", "Non-COVID-19"),
                                'Gender' = ifelse(sub_df$sex == "female", "Female", "Male"))

# assign colors to categorical variables
annotationColors <- list(Disease_Status = c('COVID-19' = '#5C72E0',
                                            'Non-COVID-19' = '#E05A79'),
                         Gender = c('Female' = '#7F80B1', 'Male' = '#7FD1B9'))

# generate heatmap using pheatmap
pheatmap(genes_t,
         clustering_distance_rows = "euclidean",
         clustering_distance_cols = "euclidean",
         annotation_col = annotationData,
         annotation_colors = annotationColors,
         fontsize_row = 10,
         fontsize_col = 2,
         angle_col = 45)
```



Jitter plot

```
# add geome_jitter layer to visualize gene expression by gender, colored by disease status
ggplot(data = full_df, aes(x = sex, y = ABCB4, color = disease_status)) +
  geom_jitter(width = 0.25, size = 2, alpha = 0.7) +
  scale_color_manual(values = c('#5C72E0', '#E05A79'),
                     labels = c('COVID-19', 'Non-COVID-19')) +
  labs(title = 'ABCB4 Expression Across Gender and Disease Status',
```

```
      x = 'Gender', y = 'ABCB4 Gene Expression', color = 'Disease Status') +
scale_x_discrete(labels = c('Female', 'Male')) +
myTheme
```