Final project

2025-08-20

Read in data

```
### Read in files.

setwd("/Users/mengyingxia/Desktop/QBS103/Github/Final-project")

#check where we are
getwd()
```

```
## [1] "/Users/mengyingxia/Desktop/QBS103/Github/Final-project"
```

```
#use read.csv
gene_exp  <- read.csv(file ="QBS103_GSE157103_genes.csv",row.names=1)
meta_data <- read.csv(file = "QBS103_GSE157103_series_matrix-1.csv",row.names=1)


#my rationale to choose the gene - the range of expression is wide across sample

#calculate range for each row
diff_exp <- apply(gene_exp, 1, function(x) max(x) - min(x))

#sort the range
diff_exp <- sort(diff_exp)
print(diff_exp)
```

```
##        AADAC       ABCA12        ABCA8         A1CF       ABCC12      AADACL2
##         0.00         0.01         0.01         0.02         0.02         0.03
##        ABCC8      AADACL4        AADAT        ABCG8      AADACL3        ABCG4
##         0.03         0.04         0.06         0.06         0.07         0.07
##        A2ML1        ABCG5        A4GNT        ABCB5       ABCB11        ABHD1
##         0.10         0.10         0.19         0.22         0.40         0.47
##         AARD       ABCA10        ABCA9       A4GALT        ABCG2       ABCC11
##         0.55         0.65         0.65         0.72         0.74         0.77
##          A2M ABHD14A-ACY1         AASS      ABHD16B        ABCA3        AANAT
##         0.88         1.08         1.10         1.19         1.21         1.46
##        ABCA4        ABCB4      A3GALT2      ABHD12B        ABCA6         A1BG
##         1.51         1.65         1.75         2.17         2.67         2.75
##        AARS2        ABCC9        ABCC2        ABHD8        ABCB8         AACS
##         4.93         4.95         5.00         5.16         5.19         5.28
##        ABCA5        ABCD2        ABCB6       ABCA13        ABHD6        ABCB9
##         5.96         6.12         6.29         8.09         8.23         8.31
##         ABI2      ABHD17C        ABCB1        ABCC6       ABCC10        ABCB7
```

```
##         8.54         8.72         8.73         8.85         8.87         9.28
##       ABHD18       ABHD15        ABCF2        ABCD1        AASDH      ABHD17B
##         9.37         9.93        10.41        10.46        11.23        12.65
##        ABCC1         AAK1        ABCC4        ABCD3       ABHD11       ABHD12
##        12.97        13.05        13.56        14.98        15.31        15.84
##        ABCD4       ABCB10        ABCC3         ABAT ABCF2-H2BE1       ABHD10
##        17.53        17.81        18.24        18.50        18.62        19.13
##         AATK        AAMDC       AARSD1         AAAS      ABHD14A        ABCA2
##        19.48        19.93        21.04        24.29        24.55        25.04
##         AAR2     AASDHPPT        AAGAB       ABHD13        ABCF3        AARS1
##        25.78        26.04        26.30        26.48        28.32        29.11
##        ABCF1        ABCC5        ABCE1        ABCG1      ABHD17A        ABCA1
##        30.91        31.31        37.09        43.76        43.92        47.82
##        ABHD4         AATF        ABCA7      ABHD14B      ABHD16A         AAMP
##        47.84        49.79        53.35        58.87        62.25        70.13
##        ABHD2         ABI1        ABHD5        ABHD3
##        88.22        93.54       173.79       202.71
```

```
#The gene I choose is ABHD5 and the reference is https://www.ncbi.nlm.nih.gov/gene/51099
```

```
#reshape the gene expression data
exp_ABHD5 <- as.data.frame(t(gene_exp["ABHD5",]))

#there is one row named differently in two data sets, locate and rename it
setdiff(rownames(meta_data),rownames(exp_ABHD5))
```

```
## [1] "COVID_06_:y_male_NonICU"
```

```
setdiff(rownames(exp_ABHD5),rownames(meta_data))
```

```
## [1] "COVID_06_.y_male_NonICU"
```

```
#the age was missing for this participant
rownames(meta_data)[rownames(meta_data) == "COVID_06_:y_male_NonICU"] <- "COVID_06_missing_male_NonICU"
rownames(exp_ABHD5)[rownames(exp_ABHD5) == "COVID_06_.y_male_NonICU"] <- "COVID_06_missing_male_NonICU"

#check if the rowname is identical after renaming
identical(rownames(meta_data),rownames(exp_ABHD5))
```

```
## [1] TRUE
```

```
#merge
df <- merge(meta_data, exp_ABHD5, by = "row.names")

#change the rowname and drop the first column which is rowname
rownames(df) <- df[,1]
df <- df[,-1]

#check the spelling of each category, pay attention to the blank in front of the character
table(df$sex)
```

```
##
##   female     male  unknown
##       51       74        1
```

```
levels(df$sex)
```

```
## NULL
```

```
#drop missing sex
df <- df[which(!df$sex == " unknown"),]
#change the name of each level
df$sex[df$sex == " female"] <- "Female"
df$sex[df$sex == " male"] <- "Male"
df$sex<-as.factor(df$sex)


#check the spelling of each category
table(df$disease_status)
```

```
##
##     disease state: COVID-19 disease state: non-COVID-19
##                         100                         25
```

```
#change the name
df$disease_status[df$disease_status == "disease state: COVID-19"] <- "COVID-19"
df$disease_status[df$disease_status == "disease state: non-COVID-19"] <- "Non-COVID-19"
df$disease_status<-as.factor(df$disease_status)

#check each category again
table(df$disease_status)
```

```
##
##     COVID-19 Non-COVID-19
##          100           25
```

Table1 for statistics

```
table1 <- data.frame(
  Variable = c("\\textbf{Sex} n (\\%)",
               "\\quad Female",
               "\\quad Male",
               "\\textbf{ICU status} n (\\%)",
               "\\quad No",
               "\\quad Yes",
               "\\textbf{Charlson score} Median [IQR]",
               "\\textbf{Ferritin (ng/ml)} Median [IQR]",
               "\\quad Missing",
               "\\textbf{Procalcitonin (ng/ml)} Median [IQR]",
               "\\quad Missing"),
  `COVID-19\\\\(N = 100, 80\\%)` = c("", "38 (38.0\\%)", "62 (62.0\\%)",
```

```
                                    "", "50 (50.0\\%)", "50 (50.0\\%)",
                                    "3.0 [1.0, 5.0]",
                                    "652.0 [307.5, 1179.0]",
                                    "6 (6.0\\%)",
                                    "0.6 [0.2, 1.8]",
                                    "13 (13.0\\%)"),
  `Non-COVID-19\\\\(N = 25, 20\\%)` = c("", "13 (52.0\\%)", "12 (48.0\\%)",
                                        "", "10 (40.0\\%)", "15 (60.0\\%)",
                                        "4.0 [2.0, 6.0]",
                                        "142.0 [83.5, 325.5]",
                                        "9 (36.0\\%)",
                                        "0.4 [0.3, 0.7]",
                                        "10 (40.0\\%)"),
  `Overall\\\\(N = 125, 100\\%)` = c("", "51 (40.8\\%)", "74 (59.2\\%)",
                                     "", "60 (48.0\\%)", "65 (52.0\\%)",
                                     "3.0 [2.0, 5.0]",
                                     "573.0 [222.0, 1091.5]",
                                     "15 (12.0\\%)",
                                     "0.5 [0.2, 1.6]",
                                     "23 (18.4\\%)"),
  stringsAsFactors = FALSE
)

kable(
  table1, format = "latex", booktabs = TRUE, escape = FALSE,
  col.names = c(
  "",
  "COVID-19 (N = 100, 80.0\\%)",
  "Non-COVID-19 (N = 25, 20.0\\%)",
  "Overall (N = 125, 100.0\\%)"),
  caption = "Baseline Characteristics Stratified by COVID Status",
  linesep = ""
) %>%
  kable_styling(latex_options = c("hold_position"), full_width = FALSE) %>%
  column_spec(1, width = "3cm") %>%
  column_spec(2, width = "4.5cm") %>%
  column_spec(3, width = "5cm") %>%
  column_spec(4, width = "4.5cm")
```

Histogram for gene expression

```
library(ggplot2)

# Create histogram
ggplot(df, aes(x = ABHD5)) +
  geom_histogram(aes(y=..density..*100),
                 binwidth = 5,
                 fill = "lightblue",
                 color = "white",
                 boundary = 0) +
  geom_smooth(aes(y = ..density..*100),
```

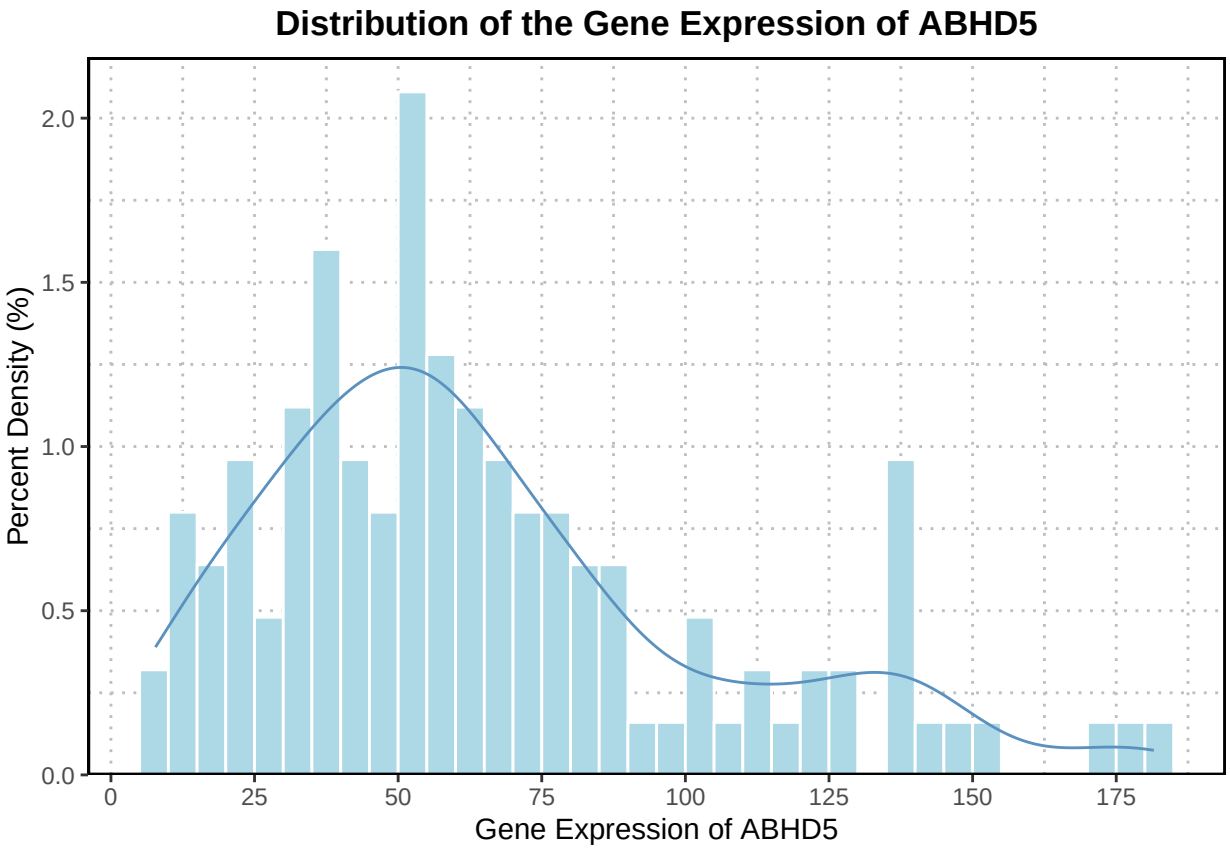
Table 1: Baseline Characteristics Stratified by COVID Status

	COVID-19 (N = 100, 80.0%)	Non-COVID-19 (N = 25, 20.0%)	Overall (N = 125, 100.0%)
Sex n (%)			
Female	38 (38.0%)	13 (52.0%)	51 (40.8%)
Male	62 (62.0%)	12 (48.0%)	74 (59.2%)
ICU status n (%)			
No	50 (50.0%)	10 (40.0%)	60 (48.0%)
Yes	50 (50.0%)	15 (60.0%)	65 (52.0%)
Charlson score Median [IQR]	3.0 [1.0, 5.0]	4.0 [2.0, 6.0]	3.0 [2.0, 5.0]
Ferritin (ng/ml) Median [IQR]	652.0 [307.5, 1179.0]	142.0 [83.5, 325.5]	573.0 [222.0, 1091.5]
Missing	6 (6.0%)	9 (36.0%)	15 (12.0%)
Procalcitonin (ng/ml) Median [IQR]	0.6 [0.2, 1.8]	0.4 [0.3, 0.7]	0.5 [0.2, 1.6]
Missing	13 (13.0%)	10 (40.0%)	23 (18.4%)

```
            stat = "density",
            color = "#5B91BE" , linewidth = 0.5) +
 labs(title = "Distribution of the Gene Expression of ABHD5",
      x = "Gene Expression of ABHD5",
      y = "Percent Density (%)") +
 scale_x_continuous(breaks = c(0, 25, 50, 75, 100, 125, 150, 175)) +
 scale_y_continuous(expand = expansion(mult = c(0, 0.05))) +
 theme(
   plot.title = element_text(hjust = 0.5, face = "bold"),
   panel.grid.major = element_line(color = "gray",
                                   linewidth = 0.5,
                                   linetype = "dotted"),
   panel.grid.minor = element_line(color = "gray",
                                   linewidth = 0.5,
                                   linetype = "dotted"),
   panel.border = element_rect(color = "black", fill = NA, size = 0.8),
   panel.background = element_rect(fill = "white"),
   #plot.background = element_rect(fill = "lightgray"),
   axis.line = element_blank()
 )
```

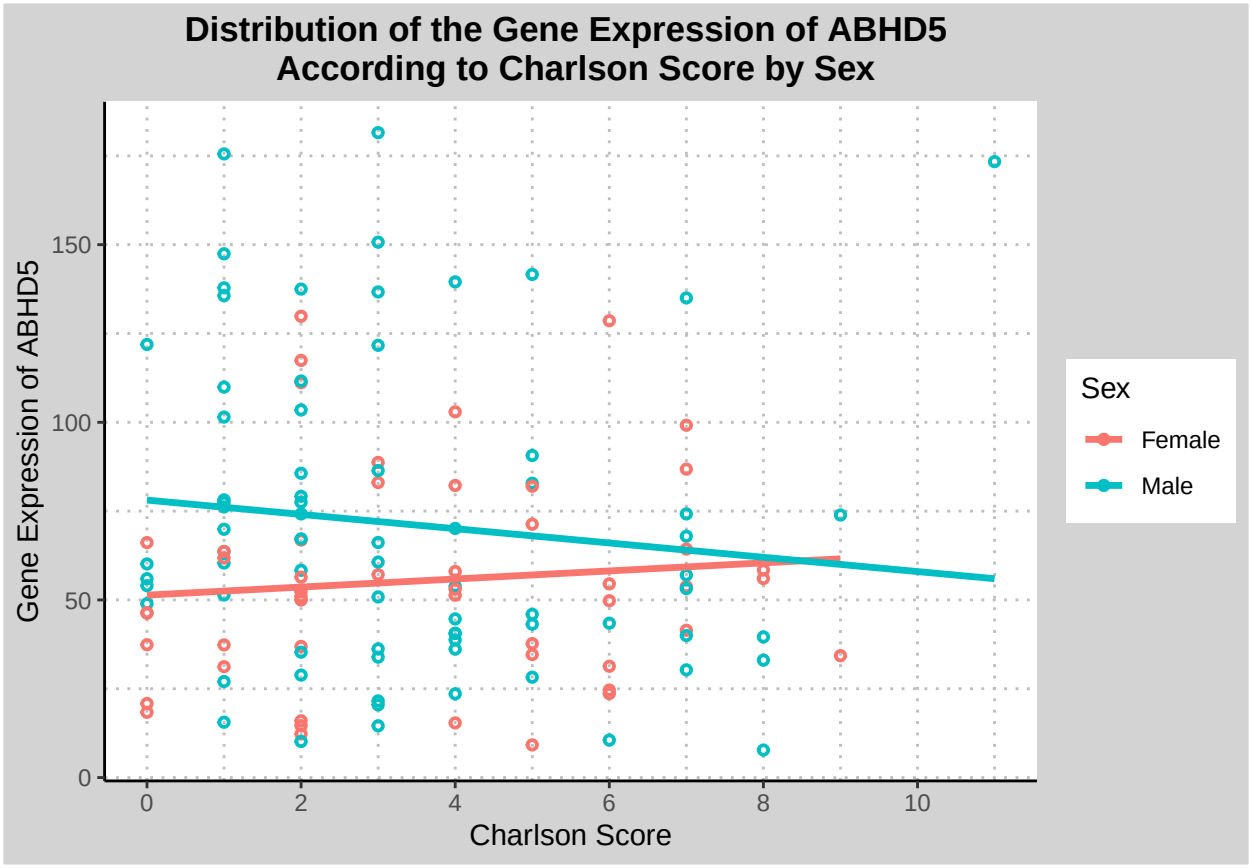
```
## Warning: The ‘size‘ argument of ‘element_rect()‘ is deprecated as of ggplot2 3.4.0.
## i Please use the ‘linewidth‘ argument instead.
## This warning is displayed once every 8 hours.
## Call ‘lifecycle::last_lifecycle_warnings()‘ to see where this warning was
## generated.
```

```
## Warning: The dot-dot notation (‘..density..‘) was deprecated in ggplot2 3.4.0.
## i Please use ‘after_stat(density)‘ instead.
## This warning is displayed once every 8 hours.
## Call ‘lifecycle::last_lifecycle_warnings()‘ to see where this warning was
## generated.
```



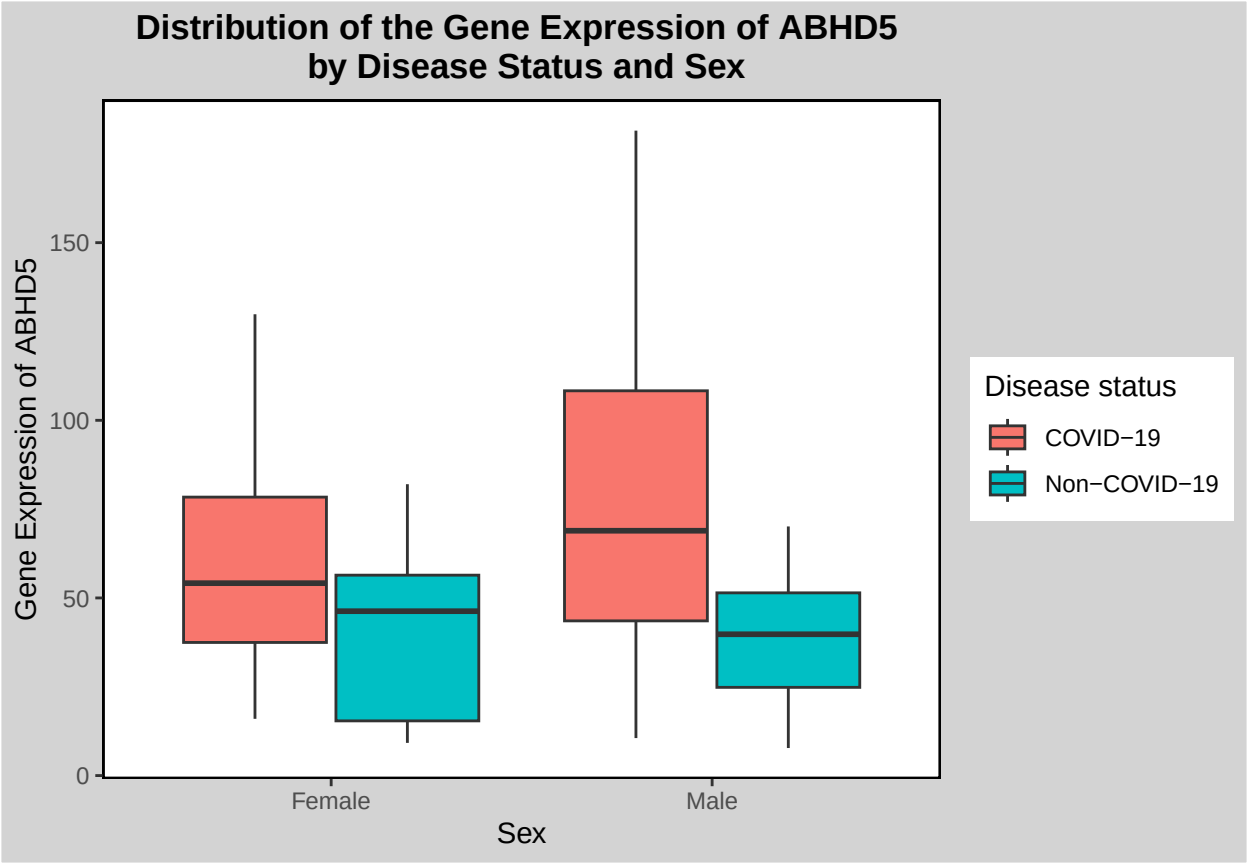
Scatterplot for gene expression and continuous covariate

```
ggplot(df, aes(x = charlson_score, y = ABHD5, color = sex)) +
    geom_point(shape = 1, size = 1, stroke = 1)+
    geom_smooth(method = "lm", size = 1.2, fill = NA)+
  labs(title = "Distribution of the Gene Expression of ABHD5 \n According to Charlson Score by Sex",
       x = "Charlson Score",
       y = "Gene Expression of ABHD5",
       color = "Sex") +
  scale_x_continuous(breaks = c(0,2,4,6,8,10))+
  theme(
    plot.title = element_text(hjust = 0.5, face="bold"),
    panel.background = element_rect(fill = "white"),
    plot.background = element_rect(fill = "lightgray"),
    panel.grid.major = element_line(color = "gray",
                                    size = 0.5,
                                    linetype = "dotted"),
    panel.grid.minor = element_line(color = "gray",
                                    size = 0.5,
                                    linetype = "dotted"),
    axis.line = element_line(color = "black"),
    legend.position = "right")
```



Boxplot of gene expression separated by both categorical covariates

```
ggplot(df, aes(x = sex, y = ABHD5, fill = disease_status)) +
  geom_boxplot(position = position_dodge(width = 0.8)) +
  labs(title = "Distribution of the Gene Expression of ABHD5 \n by Disease Status and Sex",
       x = "Sex",
       y = "Gene Expression of ABHD5",
       fill = "Disease status") +
  theme(
    plot.title = element_text(hjust = 0.5, face="bold"),
    panel.background = element_rect(fill = "white"),
    plot.background = element_rect(fill = "lightgray"),
    panel.border = element_rect(color = "black", fill = NA, size = 0.8),
    panel.grid = element_blank(),
    axis.line = element_blank(),
    legend.position = "right")
```



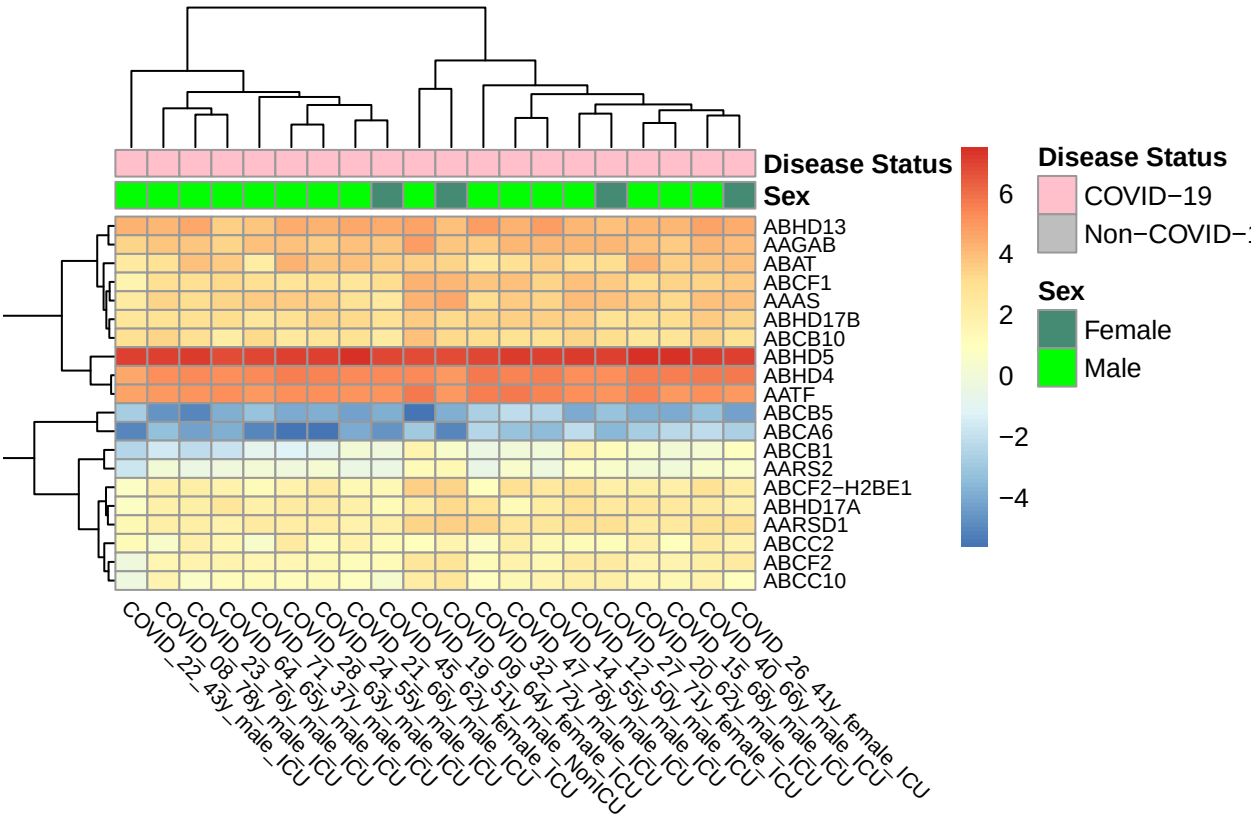
Heatmap

```
random <- c("ABHD17A", "ABCF2", "ABCB5", "ABCC2", "ABCC10", "ABHD4", "ABCF1", "ABHD17B", "AATF", "ABCB1

#reshape the gene expression data
exp_genes <- as.data.frame(t(gene_exp[random,]))

exp_genes <- exp_genes[rownames(exp_genes) != "NONCOVID_15_83y_unknown_ICU",]
rownames(exp_genes)[rownames(exp_genes) == "COVID_06_.y_male_NonICU"] <- "COVID_06_missing_male_NonICU"

#merge
df_htmp <- merge(df, exp_genes, by = "row.names")

#change the rowname and drop the first column which is rowname
rownames(df_htmp) <- df_htmp[,1]
df_htmp <- df_htmp[,-1]

htmp <- merge (df_htmp[,25:44], df_htmp[,c("sex","disease_status")], by = "row.names")
rownames(htmp) <- htmp[,1]
htmp <- htmp[,-1]

# Calculate variance
variance <- apply(df_htmp[,25:44], MARGIN = 1, FUN = var)
# Order rows of gene so that highest variance in expression is on top
ordered_htmp <- htmp[order(variance,decreasing = T),]
```

```r
library(dplyr)
```

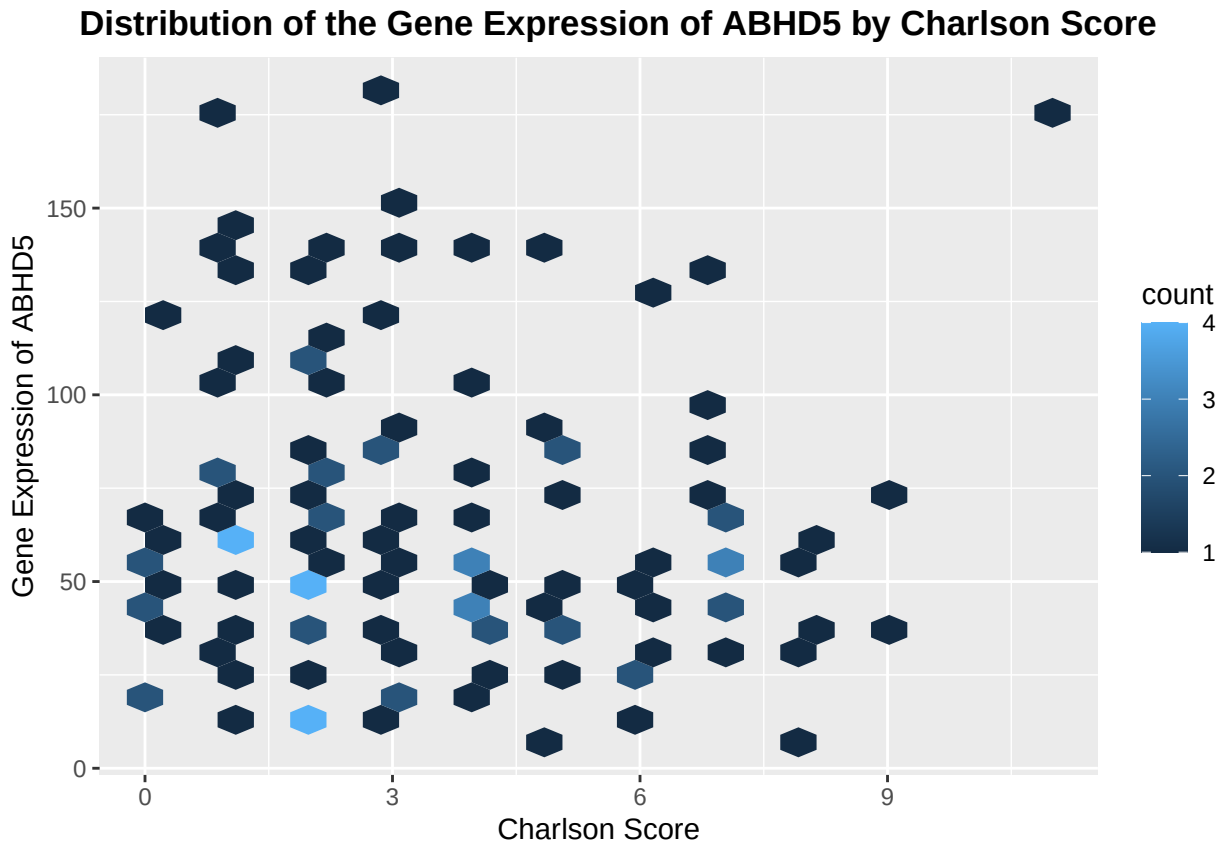
```
##
## Attaching package: 'dplyr'

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

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

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

```r
log2.htmp <- ordered_htmp %>%
  mutate(across(where(is.numeric), ~ log2(.)))

library(pheatmap)

# Define covariate for tracking bar
set.seed(1996)

annotationData <- data.frame(row.names = row.names(log2.htmp),
                             Sex = log2.htmp$sex,
                             `Disease Status` = log2.htmp$disease_status,
                             check.names = FALSE,
                             stringsAsFactors = FALSE
                             )

annotationColors <- list(Sex = c(Female = "aquamarine4", Male = "green"),
                         "Disease Status" = c("COVID-19" = "pink", "Non-COVID-19" = "grey")
                         )

pheatmap(t(log2.htmp[1:20,1:20]),
         clustering_distance_cols = 'euclidean',
         clustering_distance_rows = 'euclidean',
         angle_col = 315,
         annotation_col = annotationData,
         annotation_colors = annotationColors,
         fontsize = 10,
         fontsize_row = 8,
         fontsize_col = 8,
         cellwidth = 12,
         cellheight = 7)
```



Hexbin

```
library(hexbin)

ggplot(df,aes(x = charlson_score, y = ABHD5)) +
  stat_binhex(bins = 25)+
  labs(title = "    Distribution of the Gene Expression of ABHD5 by Charlson Score",
       x = "Charlson Score",
       y = "Gene Expression of ABHD5") +
  theme(plot.title = element_text(hjust = 0.5, face="bold"))
```


Distribution of the Gene Expression of ABHD5 by Charlson Score
Gene Expression of ABHD5
150
100
50
0
0
3
6
9
Charlson Score
count
4
3
2
1