Final Project

2025-08-21

```
#gene: ABCB1
#continuous covariate: charlson score
#categorical covariates: disease status & sex
```

```
#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   3.5.2     v tibble    3.3.0
## v lubridate 1.9.4     v tidyr     1.3.1
## v purrr     1.1.0
## -- 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
```

```
genes <- read_csv("~/Fundations of Data Science 103/Submission 1/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‘
```

```
series <- read_csv("~/Fundations of Data Science 103/Submission 1/QBS103_GSE157103_series_matrix-1.csv"
```

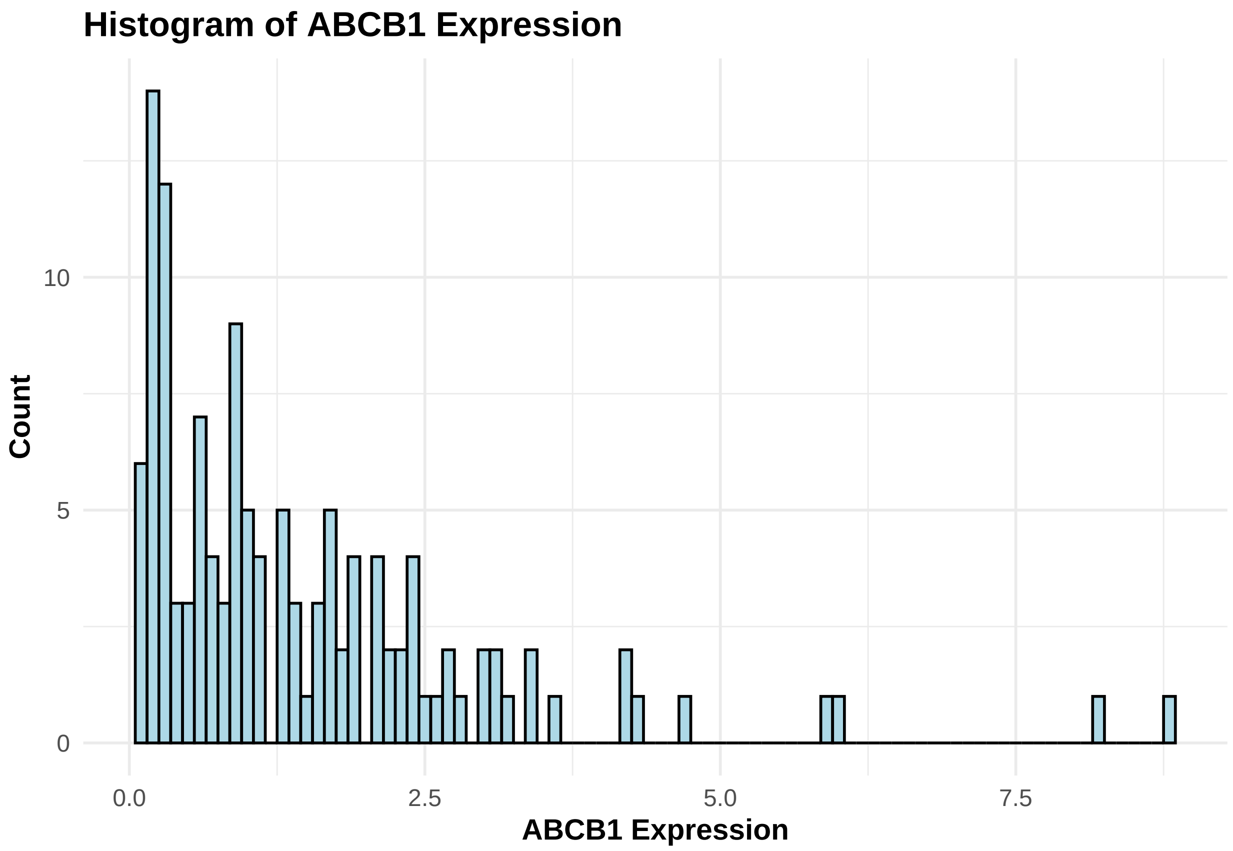
```
## Rows: 126 Columns: 25
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (21): participant_id, geo_accession, status, !Sample_submission_date, la...
## dbl  (4): channel_count, charlson_score, ventilator-free_days, hospital-free...
##
## 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
genes_long <- genes %>%
  pivot_longer(
    cols = -`...1`,
    names_to = "participant_id",
    values_to = "expression"
  ) %>%
  rename(gene = `...1`)

gene_of_interest <- "ABCB1"

gene_expr <- genes_long %>%
  filter(gene == gene_of_interest) %>%
  select(participant_id, expression)

data_merged <- series %>%
  left_join(gene_expr, by = "participant_id")

data_merged <- data_merged %>%
  rename(ventilator_free_days = `ventilator-free_days`) %>%
  mutate(
    expression           = as.numeric(expression),
    charlson_score       = as.numeric(charlson_score),
    age                  = as.numeric(age),
    ventilator_free_days = as.numeric(ventilator_free_days),

    disease_status = factor(
      disease_status,
      levels = c("disease state: COVID-19", "disease state: non-COVID-19"),
      labels = c("COVID-19", "Non-COVID-19")
    ),
    sex = factor(
      tolower(sex),
      levels = c("male","female","unknown"),
      labels = c("Male","Female","Unknown")
    ),
    icu_status = factor(
      tolower(icu_status),
      levels = c("yes","no"),
      labels = c("Yes","No")
    )
  )
```

```
## Warning: There was 1 warning in ‘mutate()‘.
## i In argument: ‘age = as.numeric(age)‘.
## Caused by warning:
## ! NAs introduced by coercion
```

```r
fab_theme <- theme_minimal(base_size = 11) +
  theme(
    axis.title = element_text(face = "bold"),
    plot.title = element_text(face = "bold"),
    plot.caption = element_text(size = 9, colour = "grey35")
  )
```

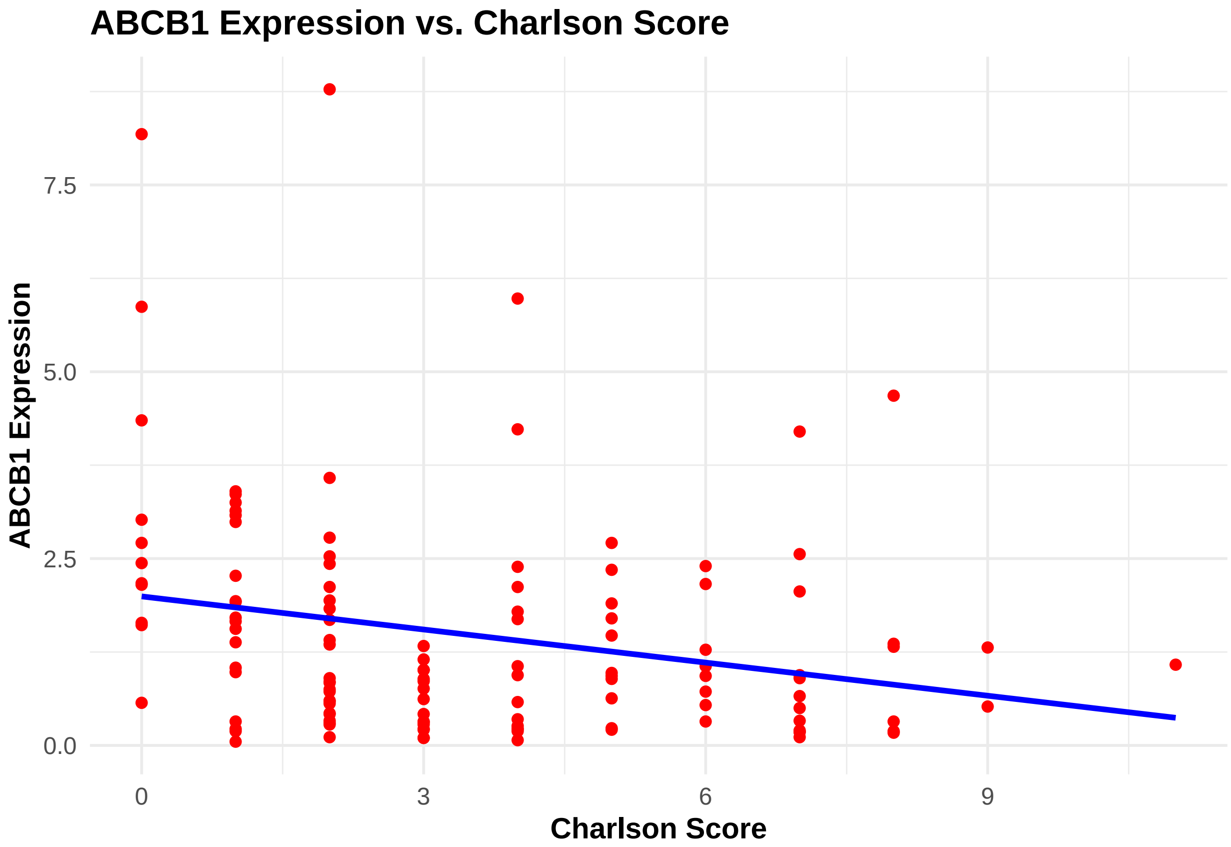
```r
library(ggplot2)

ggplot(data_merged, aes(x = expression)) +
  geom_histogram(binwidth = 0.1, fill = "lightblue", color = "black") +
  labs(
    title = "Histogram of ABCB1 Expression",
    x = "ABCB1 Expression",
    y = "Count"
  ) +
  fab_theme
```



```r
ggsave("Histogram of ABCB1 Expression.pdf", plot = ggplot(data_merged, aes(x = expression)) +
  geom_histogram(binwidth = 0.1, fill = "lightblue", color = "black") +
  labs(
    title = "Histogram of ABCB1 Expression",
    x = "ABCB1 Expression",
    y = "Count"
  ) +
  fab_theme, width = 6, height = 4, units = "in", dpi = 300)
```

```r
#Scatterplot
ggplot(data_merged, aes(x = charlson_score, y = expression)) +
  geom_point(color = "red") +
  geom_smooth(method = "lm", se = FALSE, color = "blue") +
  labs(
    title = "ABCB1 Expression vs. Charlson Score",
    x = "Charlson Score",
    y = "ABCB1 Expression"
  ) +
  fab_theme
```

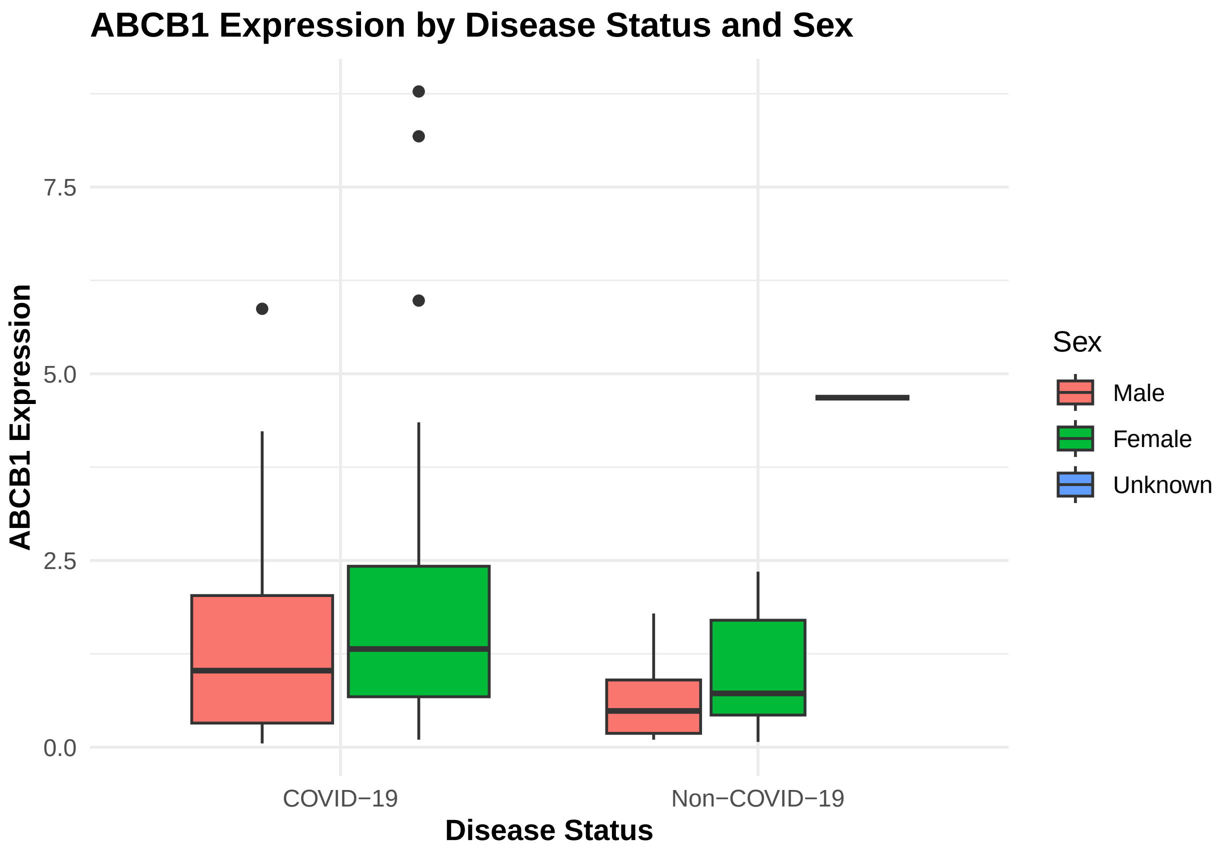
```
## ‘geom_smooth()‘ using formula = ’y ~ x’
```



```
ggsave("Scatterplot of ABCB1 Expression.pdf", plot = ggplot(data_merged, aes(x = charlson_score, y = exp
  geom_point(color = "red") +
  geom_smooth(method = "lm", se = FALSE, color = "blue") +
  labs(
    title = "ABCB1 Expression vs. Charlson Score",
    x = "Charlson Score",
    y = "ABCB1 Expression"
  ) +
  fab_theme, width = 6, height = 4, units = "in", dpi = 300)
```

```
## ‘geom_smooth()‘ using formula = ’y ~ x’
```

```
#Boxplot
ggplot(data_merged, aes(x = disease_status, y = expression, fill = sex)) +
  geom_boxplot() +
  labs(
    title = "ABCB1 Expression by Disease Status and Sex",
    x = "Disease Status",
    y = "ABCB1 Expression",
    fill = "Sex"
  ) +
  fab_theme
```



```
ggsave("Boxplot of ABCB1 Expression.pdf", plot = ggplot(data_merged, aes(x = disease_status, y = expres
  geom_boxplot() +
  labs(
    title = "ABCB1 Expression by Disease Status and Sex",
    x = "Disease Status",
    y = "ABCB1 Expression",
    fill = "Sex"
  ) +
  fab_theme, width = 6, height = 4, units = "in", dpi = 300)
```

Build a function to create the plots you made for Presentation 1, incorporating any feedback you received on your submission. 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 (10 pts)

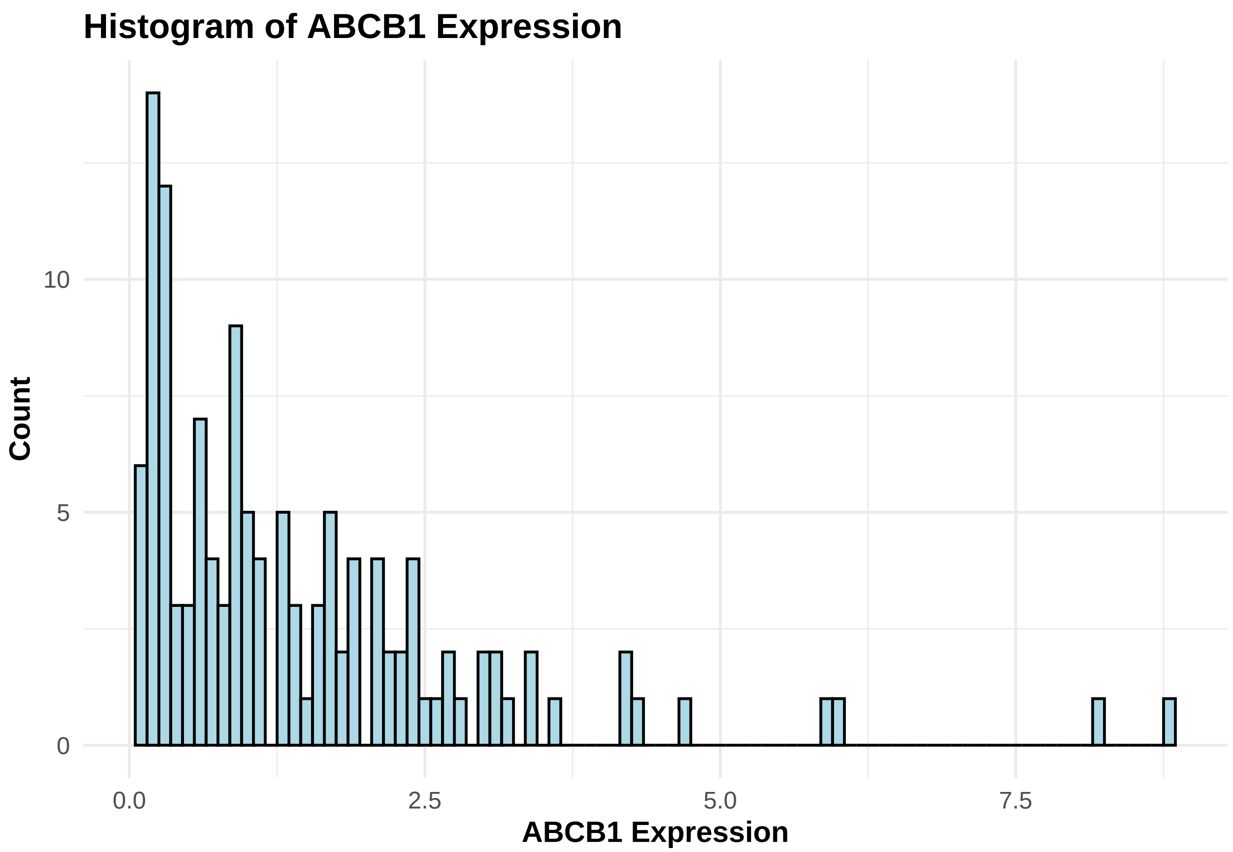
```
gene_plots <- function(genes_long, series, gene_list, cont_var, cate_var1, cate_var2){
  library(tidyverse)
  library(ggplot2)
```

```
plot_list <- list()

for (gene_of_interest in gene_list) {
  gene_expr <- genes_long %>%
  filter(gene == gene_of_interest) %>%
  select(participant_id, expression)

  data_merged <- series %>%
    left_join(gene_expr, by = "participant_id")

  data_merged <- data_merged %>%
    rename(ventilator_free_days = `ventilator-free_days`) %>%
    mutate(
      expression = as.numeric(expression),
      charlson_score = as.numeric(charlson_score),
      age = as.numeric(age),
      ventilator_free_days = as.numeric(ventilator_free_days),

      disease_status = factor(
        disease_status,
        levels = c("disease state: COVID-19", "disease state: non-COVID-19"),
        labels = c("COVID-19", "Non-COVID-19")
      ),
      sex = factor(
        tolower(sex),
        levels = c("male","female","unknown"),
       labels = c("Male","Female","Unknown")
     ),
     icu_status = factor(
       tolower(icu_status),
       levels = c("yes","no"),
        labels = c("Yes","No")
     )
   )

fab_theme <- theme_minimal(base_size = 11) +
theme(
  axis.title = element_text(face = "bold"),
  plot.title = element_text(face = "bold"),
  plot.caption = element_text(size = 9, colour = "grey35")
)

  p_hist <- ggplot(data_merged, aes(x = expression)) +
    geom_histogram(binwidth = 0.1, fill = "lightblue", color = "black") +
    labs(
      title = paste("Histogram of", gene_of_interest, "Expression"),
      x = paste(gene_of_interest, "Expression"),
      y = "Count"
    ) +
    fab_theme

  p_scatter <- ggplot(data_merged, aes(x = .data[[cont_var]], y = expression)) +
    geom_point(color = "red") +
      geom_smooth(method = "lm", se = FALSE, color = "blue") +
      labs(
        title = paste(gene_of_interest, "Expression vs.", cont_var),
        x = str_to_title(cont_var),
        y = paste(gene_of_interest, "Expression")
      ) +
      fab_theme

    p_box <- ggplot(data_merged, aes(x = .data[[cate_var1]], y = expression, fill = .data[[cate_var2]]))
      geom_boxplot() +
      labs(
        title = paste(gene_of_interest, "Expression by", cate_var1, "and", cate_var2),
        x = str_to_title(cate_var1),
        y = paste(gene_of_interest, "Expression"),
        fill = cate_var2
       ) +
       fab_theme

    plot_list[[gene_of_interest]] <- list(
      histogram = p_hist,
      scatterplot = p_scatter,
      boxplot = p_box
    )
  }
  return(plot_list)
}
```

```
plots <- gene_plots(
  genes_long = genes_long,
  series = series,
  gene_list = "ABCB1",
  cont_var = "charlson_score",
  cate_var1 = "disease_status",
  cate_var2 = "sex"
)
```

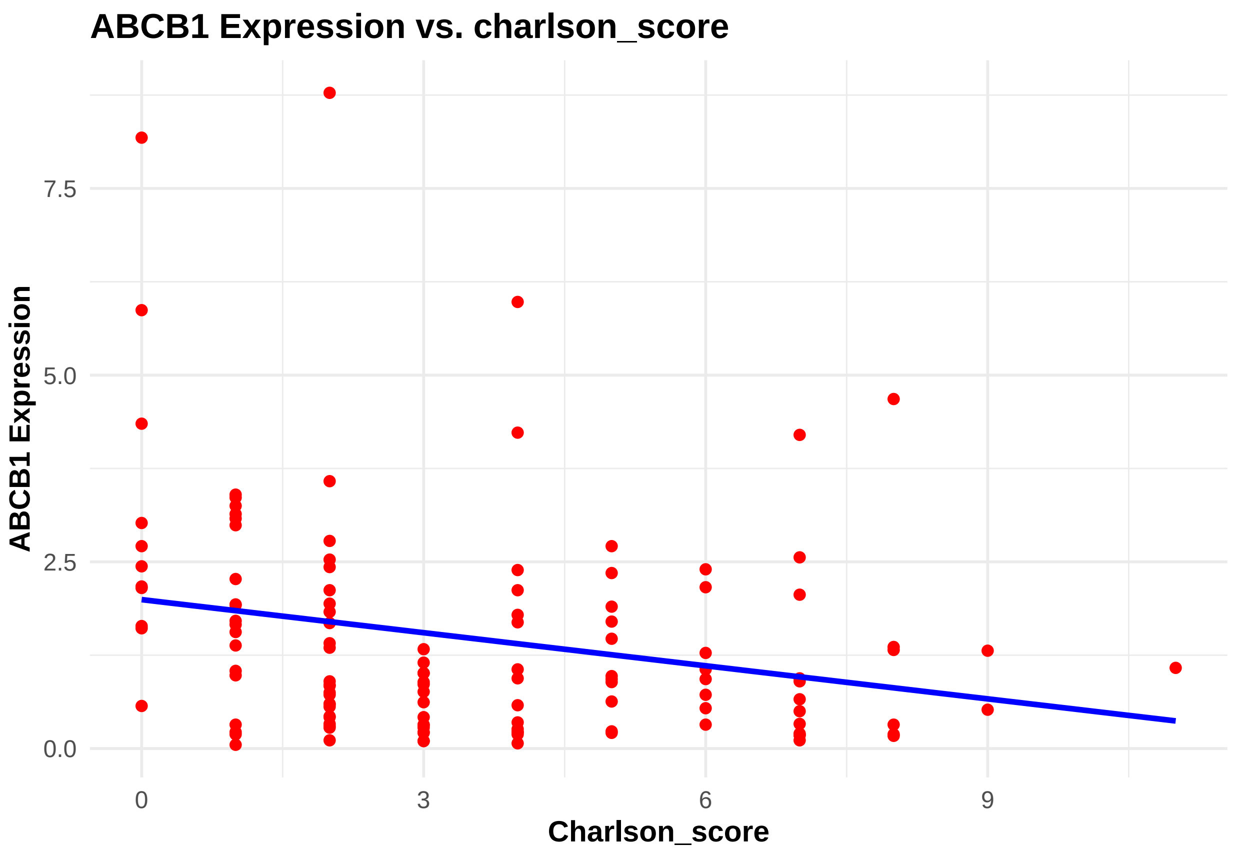
```
## Warning: There was 1 warning in ‘mutate()‘.
## i In argument: ‘age = as.numeric(age)‘.
## Caused by warning:
## ! NAs introduced by coercion
```

```
plots[["ABCB1"]][["histogram"]]
```

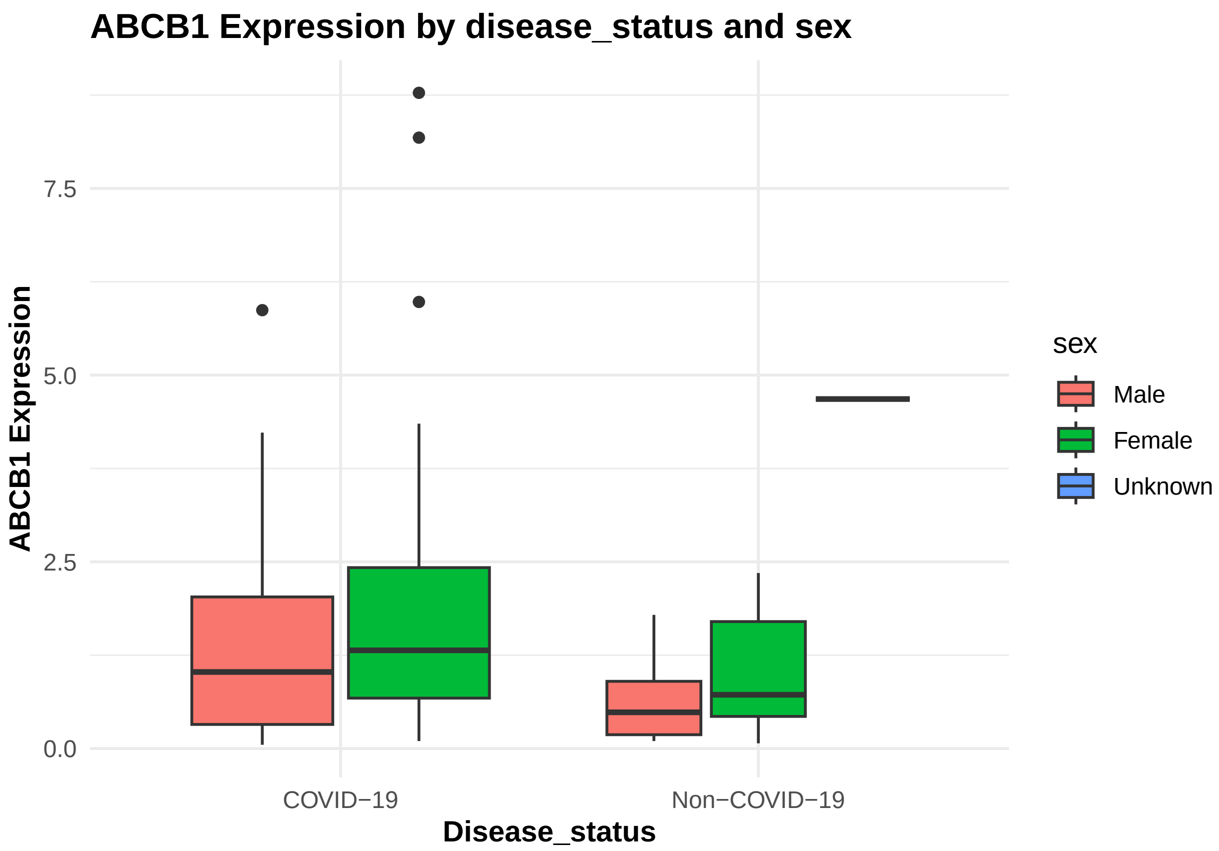


```
plots[["ABCB1"]][["scatterplot"]]
```

```
## ‘geom_smooth()‘ using formula = ’y ~ x’
```



```
plots[["ABCB1"]][["boxplot"]]
```

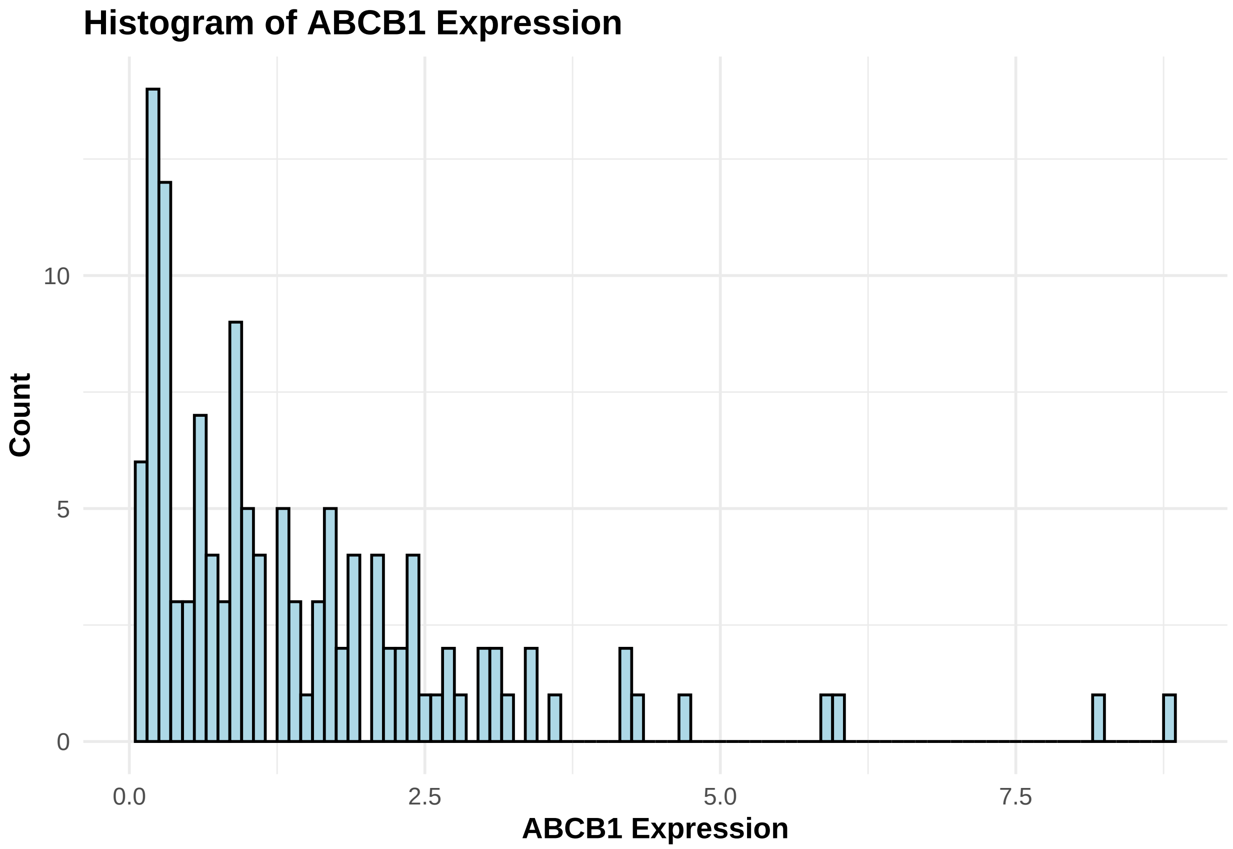


Select 2 additional genes (for a total of 3 genes) to look at and implement a loop to generate your figures using the function you created (10 pts)

```
gene_list <- c("ABCB1", "AAK1", "ABCD4")

plots <- gene_plots(
  genes_long = genes_long,
  series = series,
  gene_list = gene_list,
  cont_var = "charlson_score",
  cate_var1 = "disease_status",
  cate_var2 = "sex"
)
```

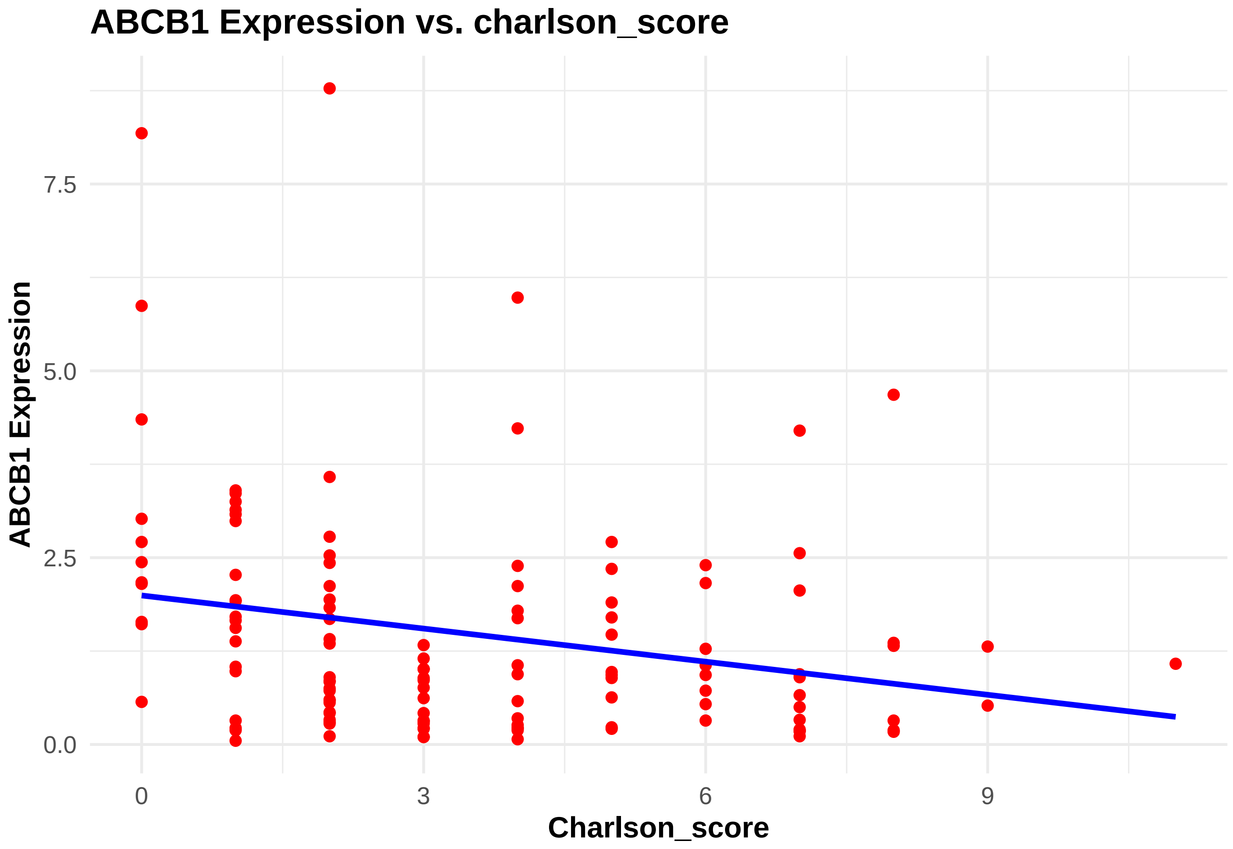
```
## Warning: There was 1 warning in ‘mutate()‘.
## i In argument: ‘age = as.numeric(age)‘.
## Caused by warning:
## ! NAs introduced by coercion
## There was 1 warning in ‘mutate()‘.
## i In argument: ‘age = as.numeric(age)‘.
## Caused by warning:
## ! NAs introduced by coercion
## There was 1 warning in ‘mutate()‘.
## i In argument: ‘age = as.numeric(age)‘.
## Caused by warning:
## ! NAs introduced by coercion
```

```r
plots[["ABCB1"]][["histogram"]]
```

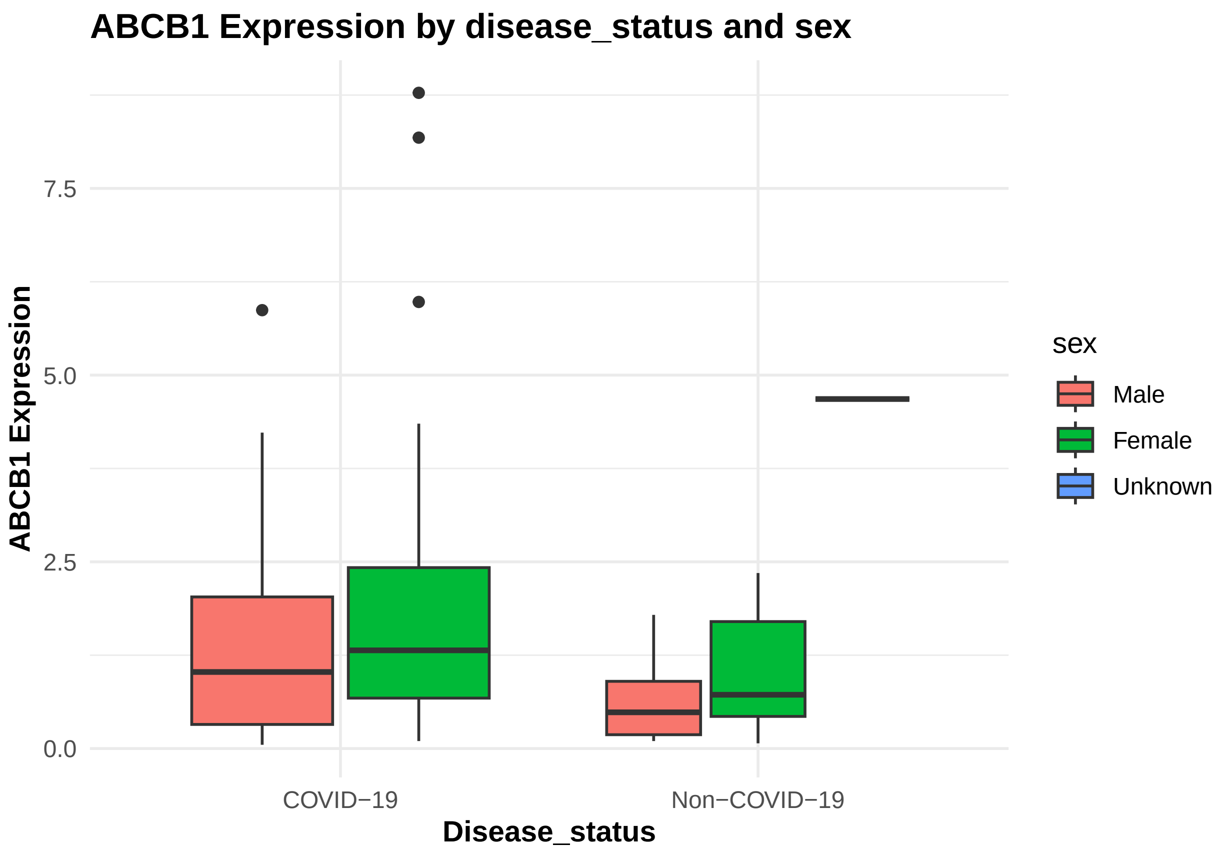


```r
plots[["ABCB1"]][["scatterplot"]]
```

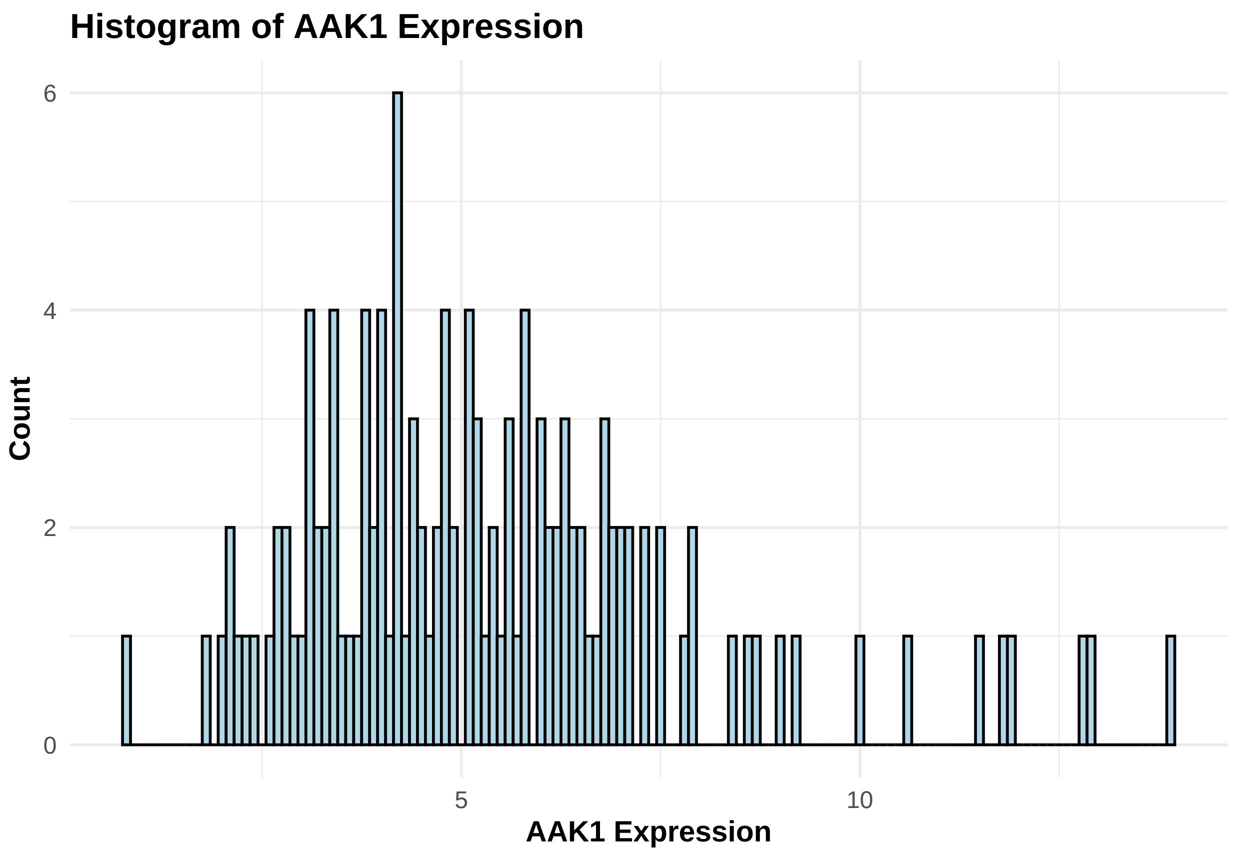
```
## `geom_smooth()` using formula = 'y ~ x'
```



```r
plots[["ABCB1"]][["boxplot"]]
```

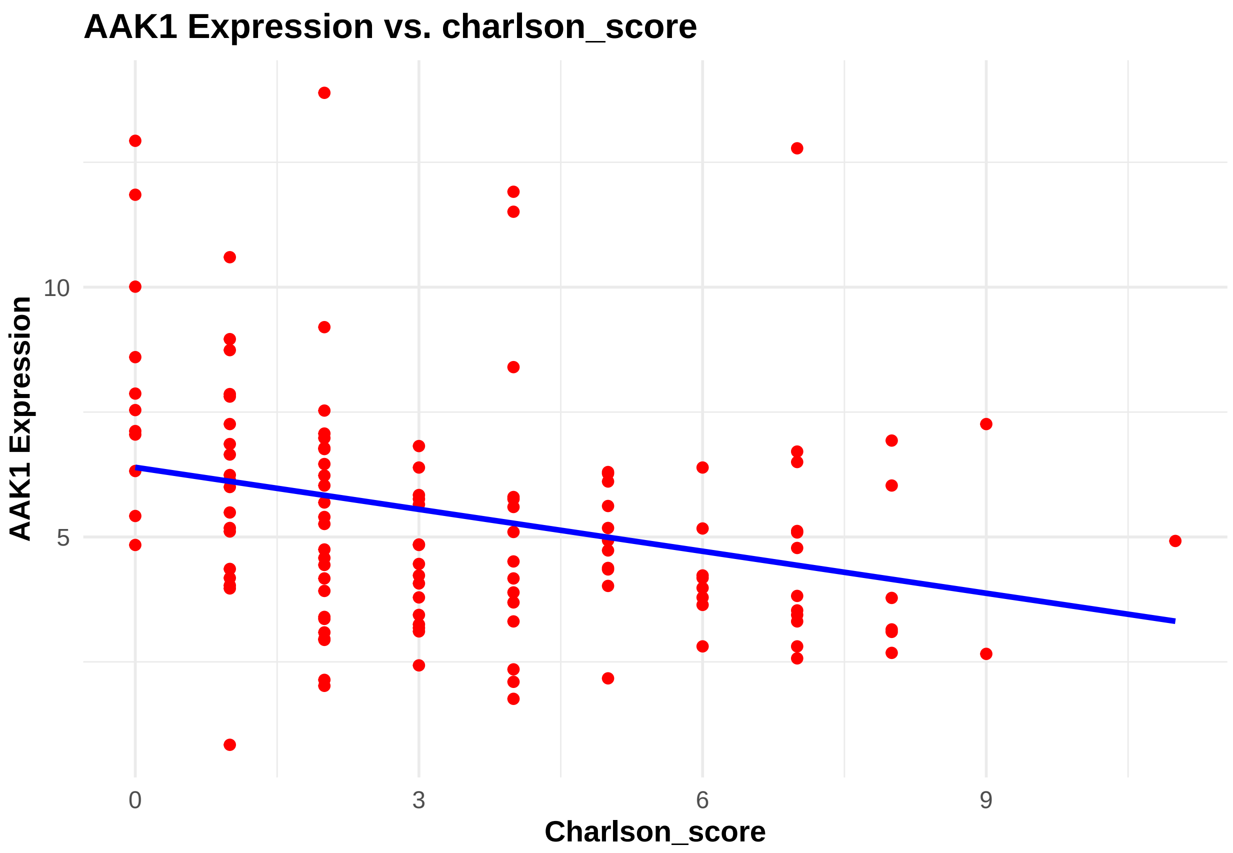


```
plots[["AAK1"]][["histogram"]]
```

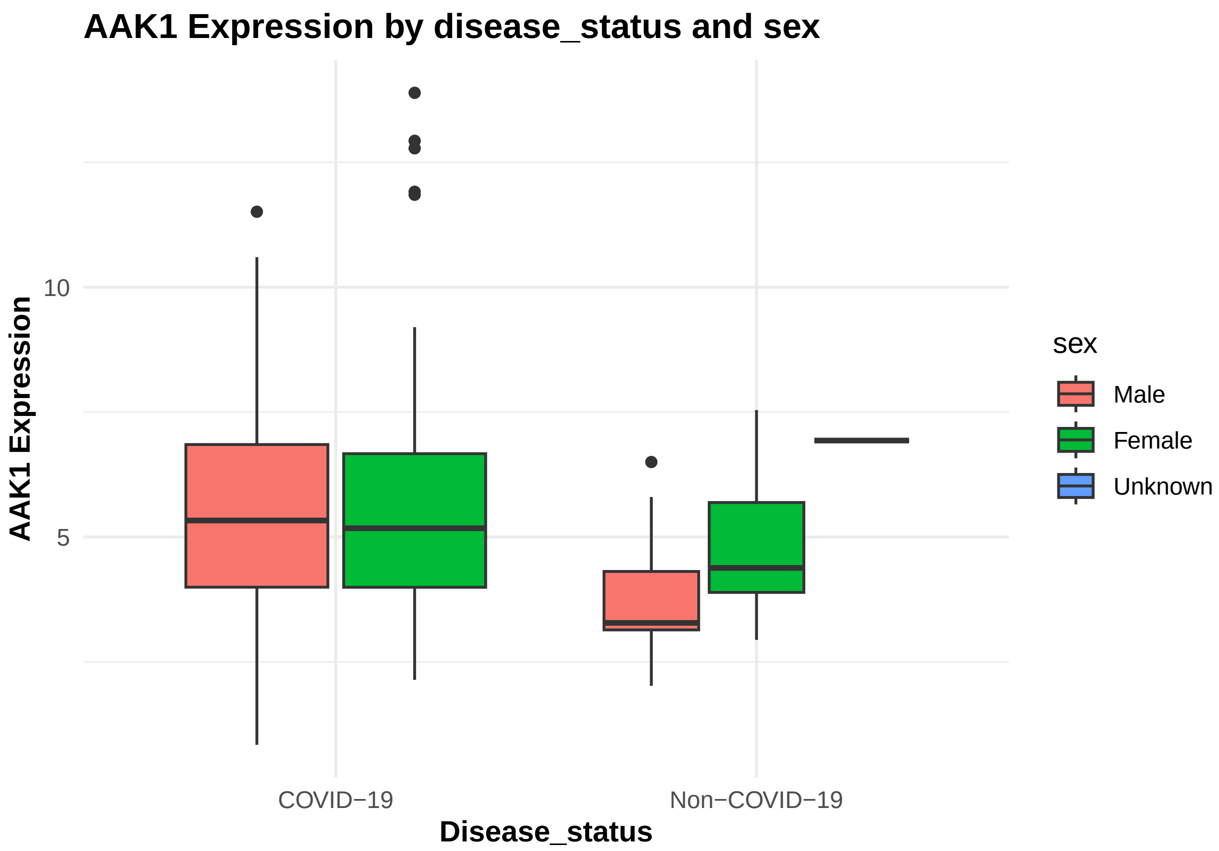


```
plots[["AAK1"]][["scatterplot"]]
```

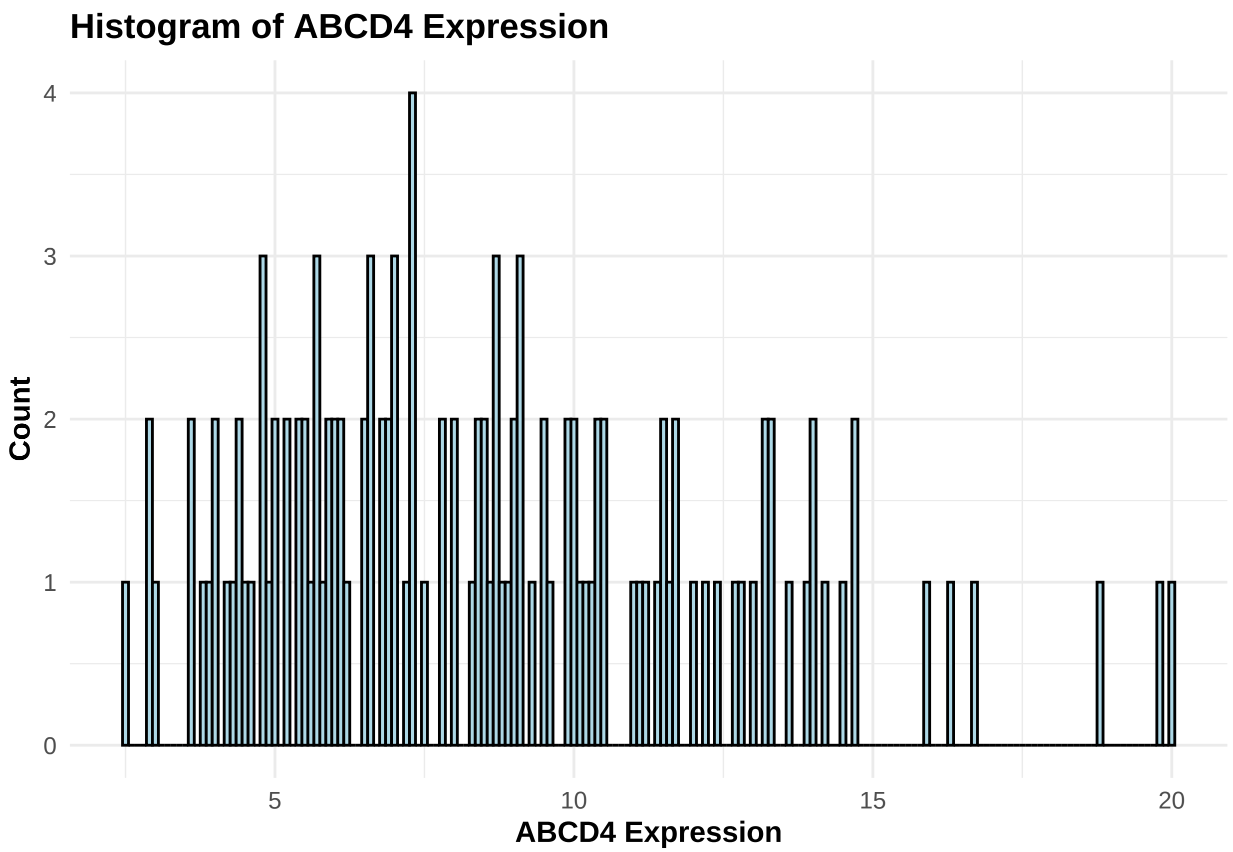
```
## ‘geom_smooth()‘ using formula = ’y ~ x’
```



```r
plots[["AAK1"]][["boxplot"]]
```

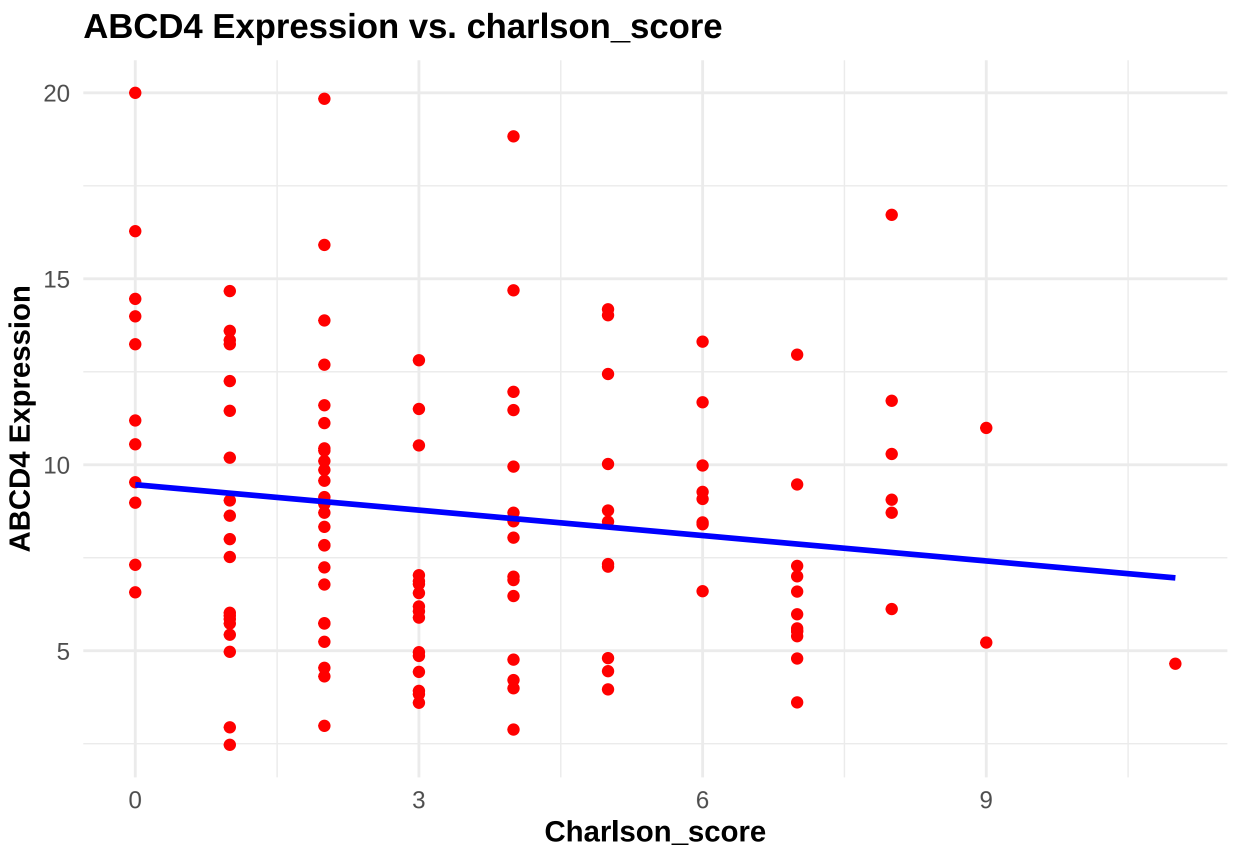


```
plots[["ABCD4"]][["histogram"]]
```

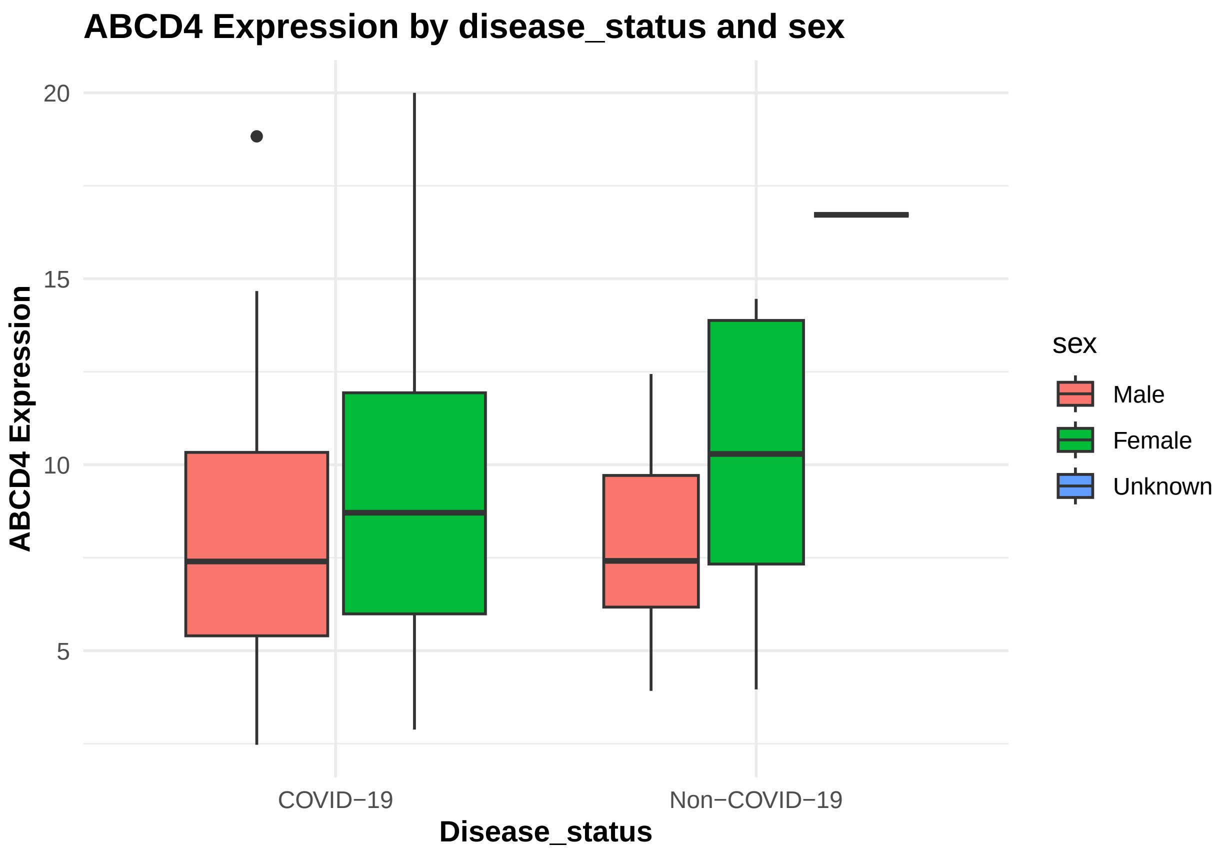


```
plots[["ABCD4"]][["scatterplot"]]
```

```
## ‘geom_smooth()‘ using formula = ’y ~ x’
```



```
plots[["ABCD4"]][["boxplot"]]
```



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

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

```
tapply(series$age, series$disease_status, mean, na.rm = TRUE)
```

```
##     disease state: COVID-19 disease state: non-COVID-19
##                    60.83673                    62.80000
```

```
tapply(series$age, series$disease_status, sd, na.rm = TRUE)
```

```
##     disease state: COVID-19 disease state: non-COVID-19
##                    16.14924                    15.60983
```

```
tapply(series$age, series$disease_status, median, na.rm = TRUE)
```

```
##     disease state: COVID-19 disease state: non-COVID-19
##                          62                          65
```

```
tapply(series$age, series$disease_status, quantile,
       probs = c(0.25, 0.75), na.rm = TRUE)
```

```
## $‘disease state: COVID-19‘
##   25%   75%
## 50.25 73.75
##
## $‘disease state: non-COVID-19‘
## 25% 75%
##  53  75
```

```
series$charlson_score <- as.numeric(series$charlson_score)

tapply(series$charlson_score, series$disease_status, mean, na.rm = TRUE)
```

```
##     disease state: COVID-19 disease state: non-COVID-19
##                    3.280000                    4.346154
```

```
tapply(series$charlson_score, series$disease_status, sd, na.rm = TRUE)
```

```
##     disease state: COVID-19 disease state: non-COVID-19
##                    2.478514                    2.415654
```

```
tapply(series$charlson_score, series$disease_status, median, na.rm = TRUE)
```

```
##     disease state: COVID-19 disease state: non-COVID-19
##                           3                           4
```

```
tapply(series$charlson_score, series$disease_status,
       quantile, probs = c(0.25, 0.75), na.rm = TRUE)
```

```
## $‘disease state: COVID-19‘
## 25% 75%
##   1   5
##
## $‘disease state: non-COVID-19‘
##  25%  75%
## 2.25 6.00
```

```
series$`ventilator-free_days` <- as.numeric(series$`ventilator-free_days`)

tapply(series$`ventilator-free_days`, series$disease_status, mean, na.rm = TRUE)
```

```
##     disease state: COVID-19 disease state: non-COVID-19
##                    19.81000                    22.42308
```

```
tapply(series$`ventilator-free_days`, series$disease_status, sd, na.rm = TRUE)
```

```
##     disease state: COVID-19 disease state: non-COVID-19
##                    11.56073                    10.06051
```

```r
tapply(series$`ventilator-free_days`, series$disease_status, median, na.rm = TRUE)
```

```
##     disease state: COVID-19 disease state: non-COVID-19
##                          28                          28
```

```r
tapply(series$`ventilator-free_days`, series$disease_status,
       quantile, probs = c(0.25, 0.75), na.rm = TRUE)
```

```
## $‘disease state: COVID-19‘
##  25%  75%
## 10.5 28.0
##
## $‘disease state: non-COVID-19‘
## 25% 75%
##  24  28
```

```r
mean(series$age, na.rm = TRUE)
```

```
## [1] 61.23577
```

```r
sd(series$age, na.rm = TRUE)
```

```
## [1] 15.99748
```

```r
median(series$age, na.rm = TRUE)
```

```
## [1] 62
```

```r
quantile(series$age, probs = c(0.25, 0.75), na.rm = TRUE)
```

```
##  25%  75%
## 50.5 74.0
```

```r
mean(series$charlson_score, na.rm = TRUE)
```

```
## [1] 3.5
```

```r
sd(series$charlson_score, na.rm = TRUE)
```

```
## [1] 2.493993
```

```r
median(series$charlson_score, na.rm = TRUE)
```

```
## [1] 3
```

```r
quantile(series$charlson_score, probs = c(0.25, 0.75), na.rm = TRUE)
```

```
## 25% 75%
##   2   5
```

```r
mean(series$`ventilator-free_days`, na.rm = TRUE)
```

```
## [1] 20.34921
```

```r
sd(series$`ventilator-free_days`, na.rm = TRUE)
```

```
## [1] 11.27923
```

```r
median(series$`ventilator-free_days`, na.rm = TRUE)
```

```
## [1] 28
```

```r
quantile(series$`ventilator-free_days`, probs = c(0.25, 0.75), na.rm = TRUE)
```

```
##   25%   75%
## 12.75 28.00
```

```r
sex_prop <- table(series$sex, series$disease_status)
sex_prop
```

```
##
##           disease state: COVID-19 disease state: non-COVID-19
##   female                       38                          13
##   male                         62                          12
##   unknown                       0                           1
```

```r
round(prop.table(sex_prop, margin = 2) * 100, 1)
```

```
##
##           disease state: COVID-19 disease state: non-COVID-19
##   female                     38.0                        50.0
##   male                       62.0                        46.2
##   unknown                     0.0                         3.8
```

```r
icu_prop <- table(series$icu_status, series$disease_status)
icu_prop
```

```
##
##       disease state: COVID-19 disease state: non-COVID-19
##   no                       50                          10
##   yes                      50                          16
```

```r
round(prop.table(icu_prop, margin = 2) * 100, 1)
```

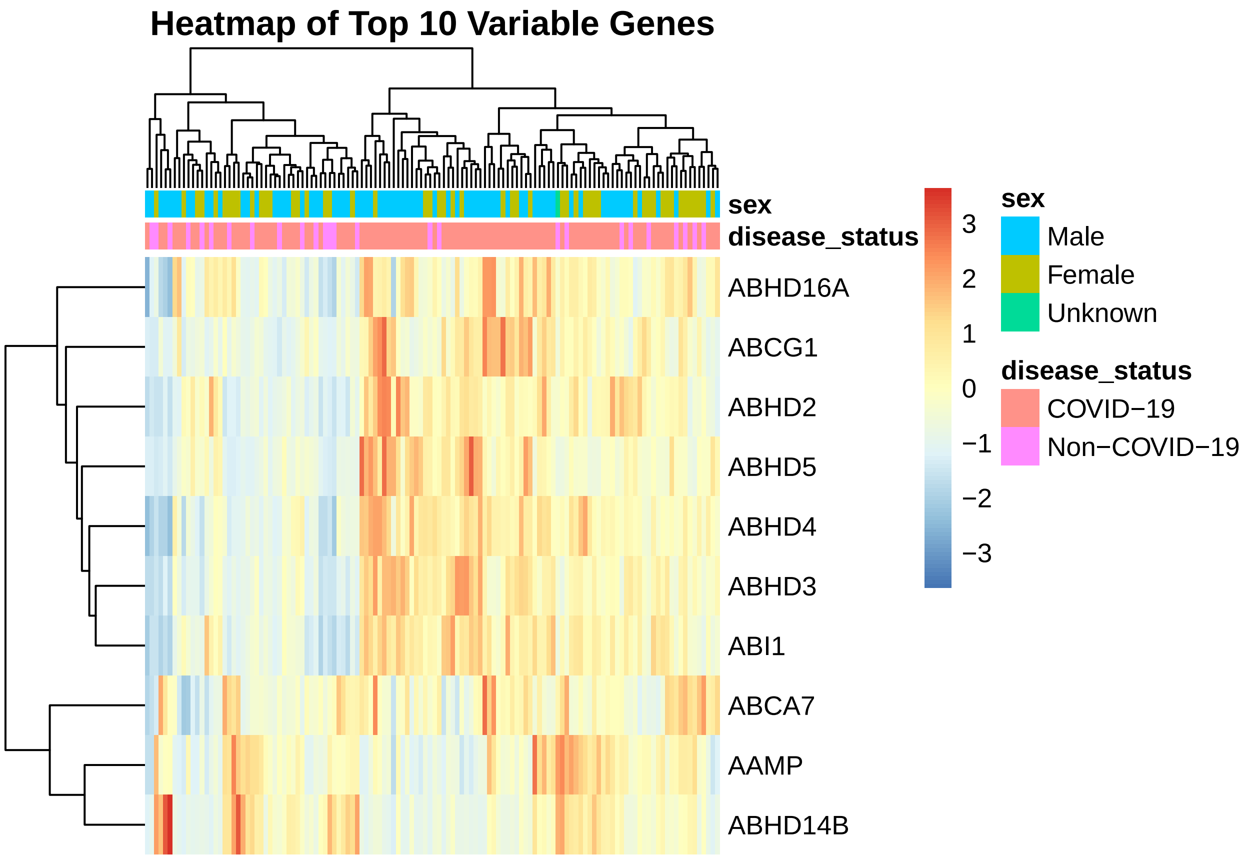
```
## 
##         disease state: COVID-19 disease state: non-COVID-19
##   no                       50.0                        38.5
##   yes                      50.0                        61.5
```

Table 1: Summary statistics of covariates stratified by `disease_status`.

	COVID-19	non-COVID-19	Total
Continuous variables[mean(sd)/median(IQR)]			
Age	60.8 (16.1) / 62 [50.3, 73.8]	62.8 (15.6) / 65 [53, 75]	61.2 (16.0) / 62 [50.5, 74.0]
Charlson score	3.28 (2.48) / 3 [1, 5]	4.35 (2.42) / 4 [2.25, 6]	3.50 (2.49) / 3 [2, 5]
Ventilator-free days	19.81 (11.56) / 28 [10.5, 28]	22.42 (10.06) / 28 [24, 28]	20.35 (11.28) / 28 [12.8, 28]
Categorical variables			
Sex			
Female	38 (38.0%)	13 (50.0%)	51 (40.5%)
Male	62 (62.0%)	12 (46.2%)	74 (58.7%)
Unknown	0 (0.0%)	1 (3.8%)	1 (0.8%)
ICU status			
No	50 (50.0%)	10 (38.5%)	60 (47.6%)
Yes	50 (50.0%)	16 (61.5%)	66 (52.4%)

Generate a heatmap (5 pts) Heatmap should include at least 10 genes Include tracking bars for the 2 categorical covariates in your boxplot Heatmaps should include clustered rows and columns

```r
library(dplyr)
library(tidyr)
library(pheatmap)

cate1 <- "disease_status"
cate2 <- "sex"

matri <- genes_long %>%
  select(participant_id, gene, expression) %>%
  mutate(expression = as.numeric(expression)) %>%
  distinct() %>%
  pivot_wider(names_from = participant_id, values_from = expression) %>%
  tibble::column_to_rownames("gene") %>%
  as.matrix()

selection <- names(sort(apply(matri, 1, var, na.rm = TRUE), decreasing = TRUE))[1:10]
matri10 <- matri[selection, , drop = FALSE]

annotation <- series %>%
  select(participant_id, disease_status, sex) %>%
  filter(participant_id %in% colnames(matri10)) %>%
  distinct()
annotation$disease_status <- factor(annotation$disease_status,
                                    levels = c("disease state: COVID-19", "disease state: non-COVID-19"
                                    labels = c("COVID-19", "Non-COVID-19"))
annotation$sex <- factor(annotation$sex,
                         levels = c("male", "female", "unknown"),
                         labels = c("Male", "Female", "Unknown"))
annotation <- as.data.frame(annotation)
rownames(annotation) <- annotation$participant_id
annotation$participant_id <- NULL
annotation <- annotation[colnames(matri10), , drop = FALSE]

matri10 <- apply(matri10, 2, as.numeric)
rownames(matri10) <- selection

pheatmap(
  matri10,
  scale = "row",
  annotation_col = annotation,
  clustering_distance_rows = "euclidean",
  clustering_distance_cols = "euclidean",
  clustering_method = "complete",
  show_rownames = TRUE,
  show_colnames = FALSE,
  main = "Heatmap of Top 10 Variable Genes"
)
```

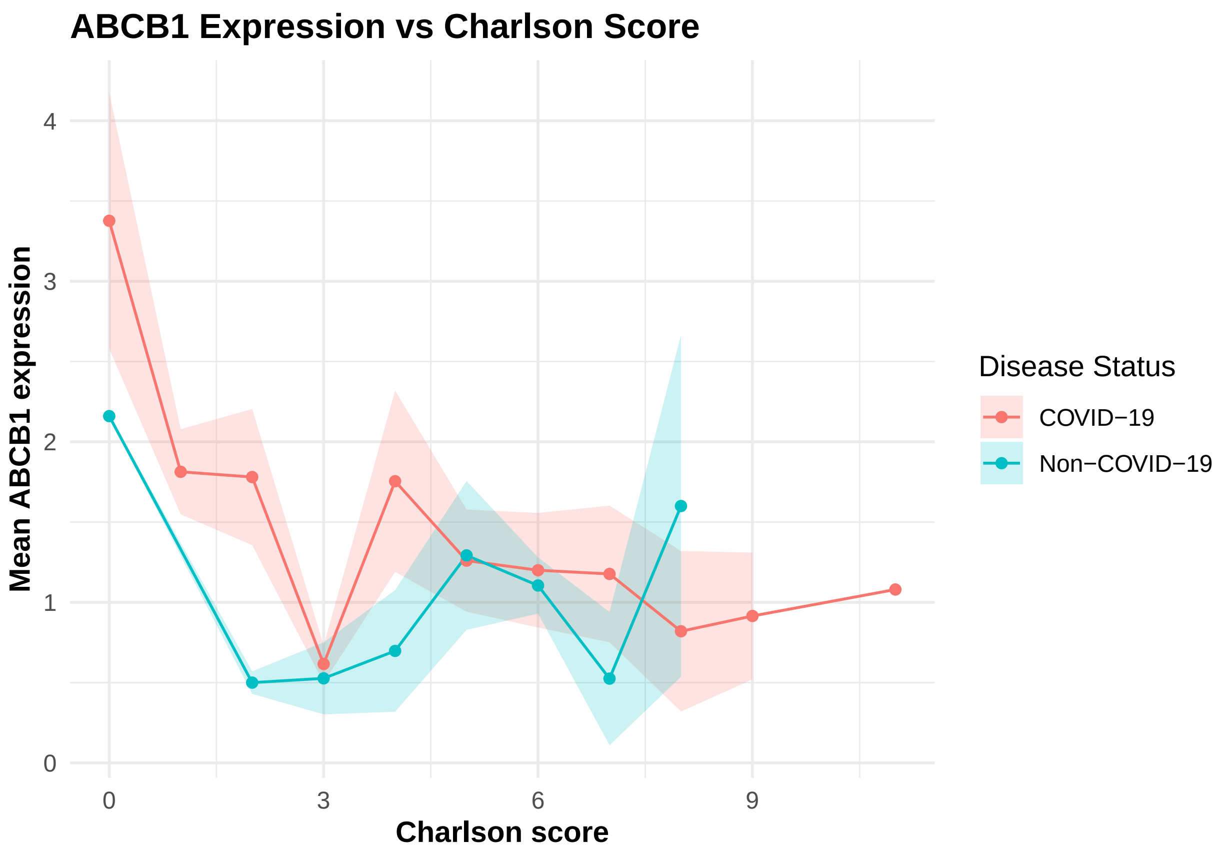


```r
pheatmap(
  matri10,
  scale = "row",
  annotation_col = annotation,
  clustering_distance_rows = "euclidean",
  clustering_distance_cols = "euclidean",
  clustering_method = "complete",
  show_rownames = TRUE,
  show_colnames = FALSE,
  main = "Heatmap of Top 10 Variable Genes",
  filename = "Heatmap of Top 10 Variable Genes.pdf",
  width = 7, height = 6
)
```

```r
sum_df <- data_merged %>%
  group_by(disease_status, charlson_score) %>%
  summarise(
    mean_expr = mean(expression, na.rm = TRUE),
    se = sd(expression, na.rm = TRUE) / sqrt(n()),
  ) %>%
  mutate(
    ymin = mean_expr - se,
    ymax = mean_expr + se
  )
```

```
## ‘summarise()‘ has grouped output by ’disease_status’. You can override using
## the ‘.groups‘ argument.
```

```r
ggplot(sum_df, aes(x = charlson_score, y = mean_expr,
                   color = disease_status, fill = disease_status)) +
  geom_ribbon(aes(ymin = ymin, ymax = ymax), alpha = 0.2, color = NA) +
  geom_line() +
  geom_point() +
  labs(
    title = "ABCB1 Expression vs Charlson Score",
    x = "Charlson score",
    y = "Mean ABCB1 expression",
    color = "Disease Status",
    fill  = "Disease Status"
  ) +
  fab_theme
```


ABCB1 Expression vs Charlson Score
Mean ABCB1 expression
Charlson score
0
1
2
3
4
0
3
6
9
Disease Status
COVID−19
Non−COVID−19