# FinalProject_XinyiYu

Xinyi Yu

2025-08-19

1.

```
# 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   1.5.0
## v ggplot2   3.5.2     v tibble    3.3.0
## v lubridate 1.9.2     v tidyr     1.3.1
## v purrr     1.0.1
## -- 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
```

```
library(kableExtra)
```

```
## Warning in !is.null(rmarkdown::metadata$output) && rmarkdown::metadata$output
## %in% : 'length(x) = 2 > 1' in coercion to 'logical(1)'
```

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

```
library(pheatmap)
library(ggplot2)
```

1.

```
#load gene expression matrix
gene_data <- read.csv(
  "/Users/yuxinyi/Dartmouth/Data Science/QBS103_GSE157103_genes.csv",
  row.names = 1, check.names = FALSE
)
#load metadata
```

```r
meta_raw  <- read.csv(
  "/Users/yuxinyi/Dartmouth/Data Science/QBS103_GSE157103_series_matrix-1.csv",
  row.names = 1, check.names = FALSE
)

#check column names of metadata
names(meta_raw)
```

```
##  [1] "geo_accession"
##  [2] "status"
##  [3] "!Sample_submission_date"
##  [4] "last_update_date"
##  [5] "type"
##  [6] "channel_count"
##  [7] "source_name_ch1"
##  [8] "organism_ch1"
##  [9] "disease_status"
## [10] "age"
## [11] "sex"
## [12] "icu_status"
## [13] "apacheii"
## [14] "charlson_score"
## [15] "mechanical_ventilation"
## [16] "ventilator-free_days"
## [17] "hospital-free_days_post_45_day_followup"
## [18] "ferritin(ng/ml)"
## [19] "crp(mg/l)"
## [20] "ddimer(mg/l_feu)"
## [21] "procalcitonin(ng/ml):"
## [22] "lactate(mmol/l)"
## [23] "fibrinogen"
## [24] "sofa"
```

2. Data Cleaning

```r
#Select and clean relevant columns
meta_sel <- meta_raw %>%
  rownames_to_column("SampleID") %>%# move sample IDs into a new column
  transmute( # create a new, cleaned metadata table
    SampleID,
    #convert each column to character, replace "unknown" with NA
    age  = na_if(trimws(as.character(.data[["age"]])), "unknown"),
    hospital_free = na_if(trimws(as.character(.data[["hospital-free_days_post_45_day_followup"]])), "unl
    ferritin = na_if(trimws(as.character(.data[["ferritin(ng/ml)"]])), "unknown"),
    sex   = na_if(trimws(as.character(.data[["sex"]])), "unknown"),
    disease_status = na_if(trimws(as.character(.data[["disease_status"]])), "unknown"),
    icu_status     = na_if(trimws(as.character(.data[["icu_status"]])), "unknown")
  ) %>%
  mutate(
    #suppress warnings
    age = suppressWarnings(as.numeric(age)),
    hospital_free = suppressWarnings(as.numeric(hospital_free)),
    ferritin = suppressWarnings(as.numeric(ferritin)),
    # Convert categorical variables into factors
    sex = factor(sex),
    disease_status = factor(disease_status),
    icu_status = factor(icu_status)
  )

summary(meta_sel)
```

```
##    SampleID               age         hospital_free      ferritin     
##  Length:125         Min.   :21.00   Min.   : 0.00   Min.   :  14.0  
##  Class :character   1st Qu.:50.25   1st Qu.: 0.00   1st Qu.: 222.0  
##  Mode  :character   Median :62.00   Median :29.00   Median : 573.0  
##                     Mean   :61.06   Mean   :24.14   Mean   : 833.5  
##                     3rd Qu.:73.75   3rd Qu.:39.00   3rd Qu.:1091.5  
##                     Max.   :88.00   Max.   :44.00   Max.   :5971.0  
##                     NA's   :3                       NA's   :15      
##      sex                          disease_status icu_status
##  female:51   disease state: COVID-19    :100    no :60    
##  male  :74   disease state: non-COVID-19: 25    yes:65    
##                                                           
##                                                           
##                                                           
##                                                           
## 
```

```r
#build gene-level dataframe used by all plots
get_gene_df <- function(gene_symbol) {
  stopifnot(gene_symbol %in% rownames(gene_data))
  expr_vec <- as.numeric(gene_data[gene_symbol, ])
  tibble(
    SampleID = colnames(gene_data),
    expr     = expr_vec
  ) %>%
    left_join(meta_sel, by = "SampleID")
}

# choose AAAS as main gene and build df_main
gene_main <- "AAAS"
df_main   <- get_gene_df(gene_main)

#sanity check
dplyr::glimpse(df_main)
```

```
## Rows: 126
## Columns: 8
## $ SampleID       <chr> "COVID_01_39y_male_NonICU", "COVID_02_63y_male_NonICU",~
## $ expr           <dbl> 18.92, 18.68, 13.85, 22.11, 8.45, 19.60, 28.59, 10.50, ~
## $ age            <dbl> 39, 63, 33, 49, 49, NA, 38, 78, 64, 62, 52, 50, 37, 55,~
## $ hospital_free  <dbl> 0, 39, 18, 39, 27, 36, 42, 0, 0, 0, 37, 22, 39, 20, 0, ~
## $ ferritin       <dbl> 946, 1060, 1335, 583, 800, 563, 366, 1103, 680, 1746, 4~
## $ sex            <fct> male, male, male, male, male, male, female, male, femal~
## $ disease_status <fct> disease state: COVID-19, disease state: COVID-19, disea~
## $ icu_status     <fct> no, no, no, no, no, no, no, yes, yes, yes, no, yes, no,~
```

```r
summary(df_main$expr)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##    5.17   11.48   15.57   16.24   19.79   29.46 
```

3. Generate a table formatted in LaTeX of summary statistics for all the covariates you looked at and 2 additional continuous (3 total) and 1 additional categorical variable (3 total). (5 pts) Stratifying by one of your categorical variables Tables should report n (%) for categorical variables Tables should report mean (sd) or median [IQR] for continuous variables

```r
#build Table 1
strata    <- "disease_status" # column used to stratify the table
cont_vars <- c("age", "hospital_free", "ferritin")# continuous covariates
cat_vars  <- c("sex", "icu_status") # categorical covariates

# continuous block
cont_block <- meta_sel %>%
  tidyr::pivot_longer(dplyr::all_of(cont_vars), names_to = "Variable", values_to = "value") %>%
  dplyr::group_by(Variable, .data[[strata]]) %>%
  dplyr::summarise(
    stat = sprintf("%.2f (%.2f)", mean(value, na.rm = TRUE), sd(value, na.rm = TRUE)),
    .groups = "drop"
  ) %>%
  tidyr::pivot_wider(names_from = dplyr::all_of(strata), values_from = stat) %>%
  # add an Overall column
  dplyr::left_join(
    meta_sel %>%
      tidyr::pivot_longer(dplyr::all_of(cont_vars), names_to = "Variable", values_to = "value") %>%
      dplyr::group_by(Variable) %>%
      dplyr::summarise(
        Overall = sprintf("%.2f (%.2f)", mean(value, na.rm = TRUE), sd(value, na.rm = TRUE)),
        .groups = "drop"
      ),
    by = "Variable"
  ) %>%
  dplyr::arrange(match(Variable, cont_vars))

# Categorical block
cat_by_strata <- dplyr::bind_rows(lapply(cat_vars, function(v) {
  #count each level within each stratum, and compute percentages column-wise
  df_levels <- meta_sel %>%
    dplyr::filter(!is.na(.data[[v]]), !is.na(.data[[strata]])) %>%
    dplyr::count(.data[[v]], .data[[strata]], name = "n") %>%
    dplyr::group_by(.data[[strata]]) %>%
    dplyr::mutate(p = round(100 * n / sum(n), 1)) %>%
    dplyr::ungroup() %>%
    dplyr::mutate(
      Variable = as.character(.data[[v]]),
      stat = sprintf("%d (%.1f%%)", n, p)
    ) %>%
    dplyr::select(Variable, .data[[strata]], stat) %>%
    tidyr::pivot_wider(
      names_from  = dplyr::all_of(strata),
      values_from = stat
    )
  header <- tibble::tibble(Variable = paste0("**", v, "**"))
  dplyr::bind_rows(header, df_levels)
}))
```

```
## Warning: Use of .data in tidyselect expressions was deprecated in tidyselect 1.2.0.
## i Please use `all_of(var)` (or `any_of(var)`) instead of `.data[[var]]`
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```

```r
#Combine continuous + categorical
# decide the order of stratum columns
if (is.factor(meta_sel[[strata]])) {
  strata_levels <- levels(meta_sel[[strata]])
} else {
  strata_levels <- sort(unique(meta_sel[[strata]]))
}

#stack continuous and categorical blocks
table1_df_tmp <- dplyr::bind_rows(
  cont_block,
  cat_by_strata
)

# ensure there is an Overall column
if (!("Overall" %in% names(table1_df_tmp))) {
  table1_df_tmp$Overall <- NA_character_
}

table1_df <- table1_df_tmp %>%
  dplyr::select(c("Variable", strata_levels, "Overall"))
```

```
## Warning: Using an external vector in selections was deprecated in tidyselect 1.1.0.
## i Please use `all_of()` or `any_of()` instead.
##   # Was:
##   data %>% select(strata_levels)
##
##   # Now:
##   data %>% select(all_of(strata_levels))
##
## See <https://tidyselect.r-lib.org/reference/faq-external-vector.html>.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```

```r
#indent categorical levels
indent_rows <- which(!grepl("^\\*\\*", table1_df$Variable))
table1_df$Variable <- gsub("\\*\\*", "", table1_df$Variable)
table1_df[is.na(table1_df)] <- "-"
```

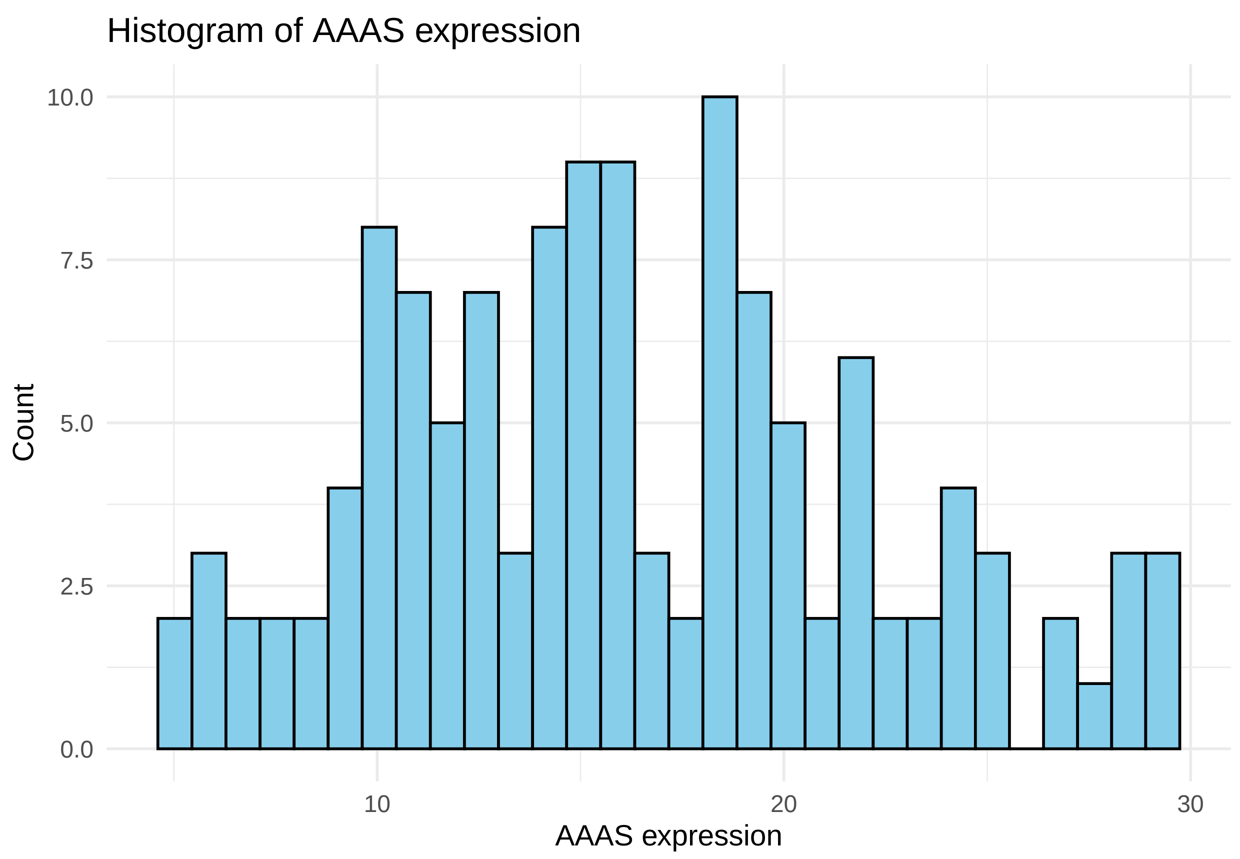
Table 1: Table 1. Summary statistics by disease status

| Variable | disease state: COVID-19 | disease state: non-COVID-19 | Overall |
|---|---|---|---|
| age | 60.84 (16.15) | 61.96 (15.36) | 61.06 (15.94) |
| hospital_free | 22.09 (16.62) | 32.36 (15.09) | 24.14 (16.79) |
| ferritin | 932.76 (1094.04) | 250.50 (238.21) | 833.52 (1042.80) |
| sex | - | - | - |
| female | 38 (38.0%) | 13 (52.0%) | - |
| male | 62 (62.0%) | 12 (48.0%) | - |
| icu_status | - | - | - |
| no | 50 (50.0%) | 10 (40.0%) | - |
| yes | 50 (50.0%) | 15 (60.0%) | - |

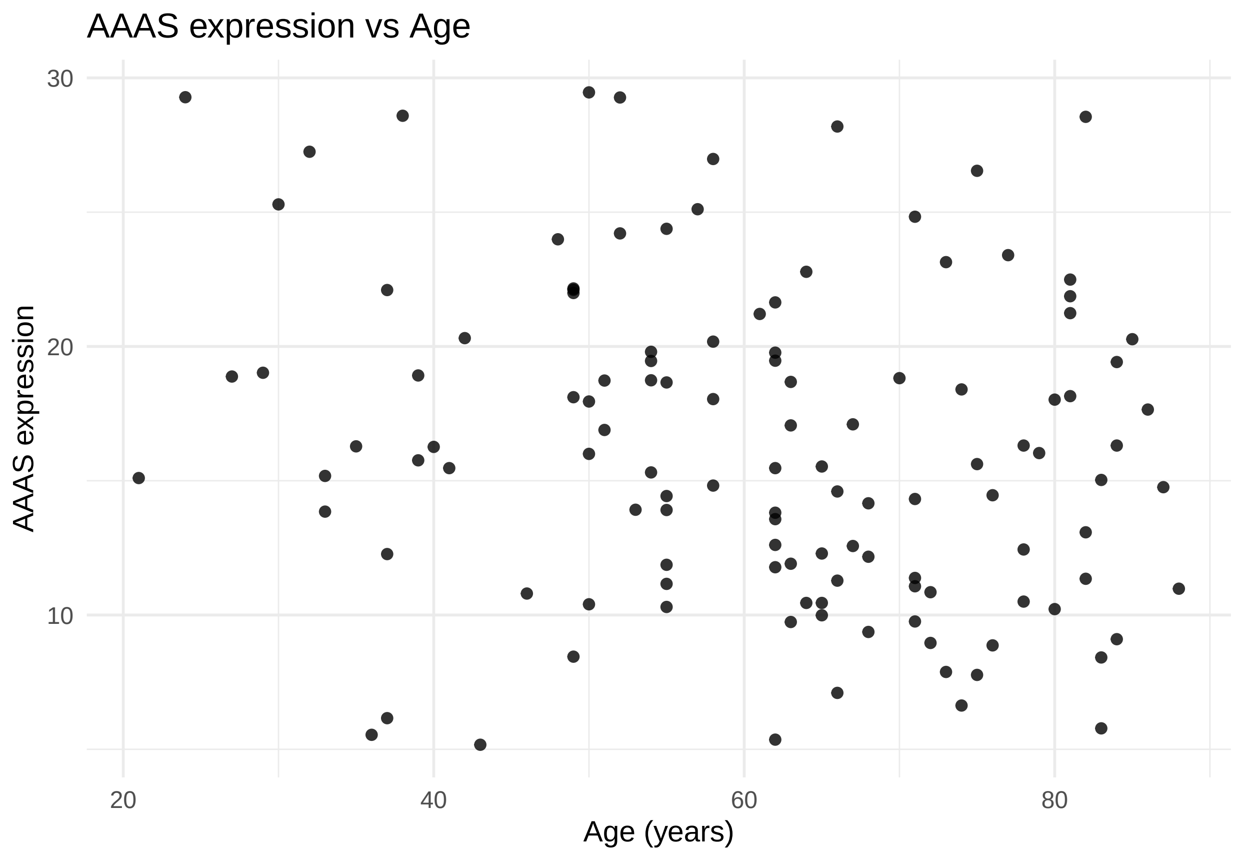
```r
#ender a clean preview
kable(table1_df, caption = "Table 1. Summary statistics by disease status") %>%
  add_indent(indent_rows) %>%
  kable_classic(full_width = FALSE)
```

4. Generate final a publication quality histogram, scatter plot, and boxplot from submission 1 (i.e. only for your first gene of interest)

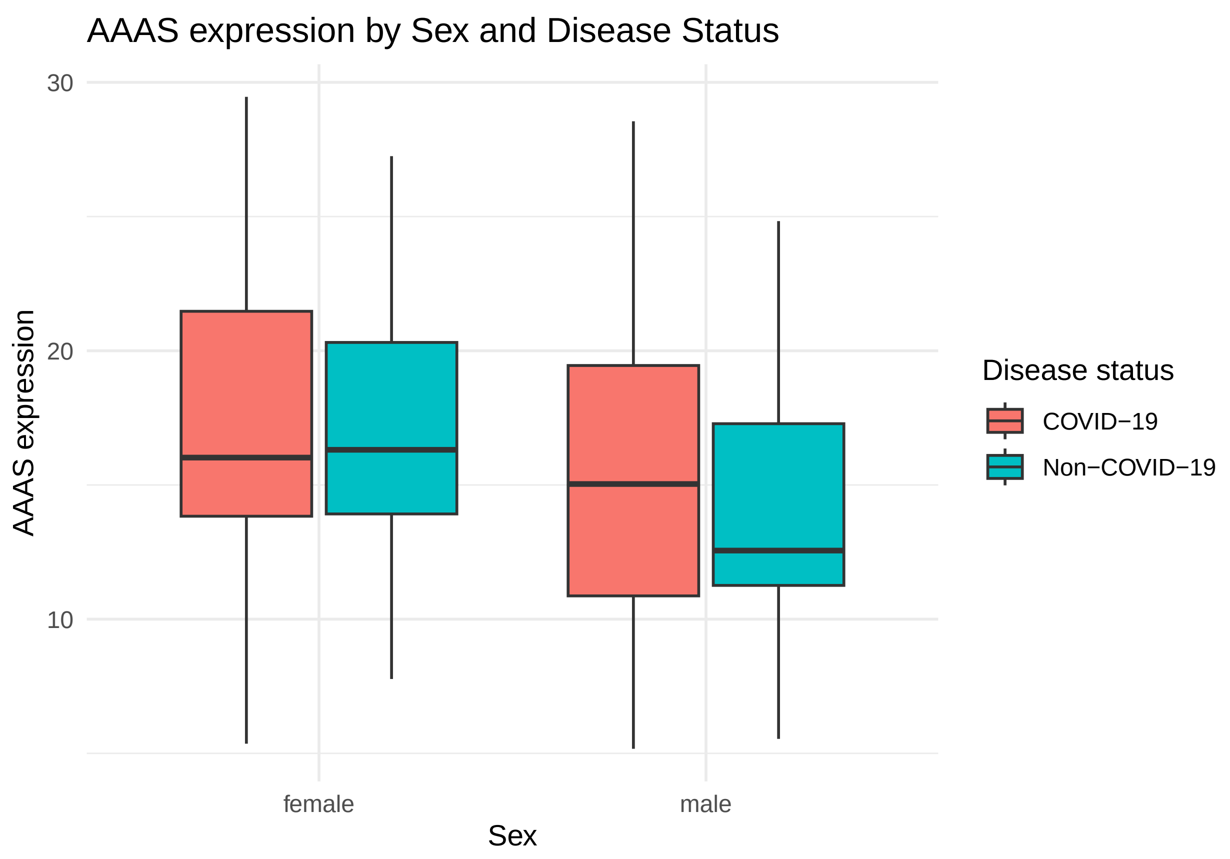
```r
#Prepare a clean plotting dataframe
df_plot <- df_main %>%
  mutate(
    disease_status = factor(
      disease_status,
      levels = c("disease state: COVID-19", "disease state: non-COVID-19"),
      labels = c("COVID-19", "Non-COVID-19")
    )
  )

# histogram)
ggplot(filter(df_plot, !is.na(expr)), aes(x = expr)) +
  geom_histogram(bins = 30, fill = "skyblue", color = "black") +
  labs(
    title = paste("Histogram of", gene_main, "expression"),
    x = paste(gene_main, "expression"), y = "Count"
  ) +
  theme_minimal()
```



```
# scatterplot
ggplot(filter(df_plot, !is.na(expr), !is.na(age)), aes(x = age, y = expr)) +
  geom_point(alpha = 0.8) +
  labs(
    title = paste(gene_main, "expression vs Age"),
    x = "Age (years)", y = paste(gene_main, "expression")
  ) +
  theme_minimal()
```



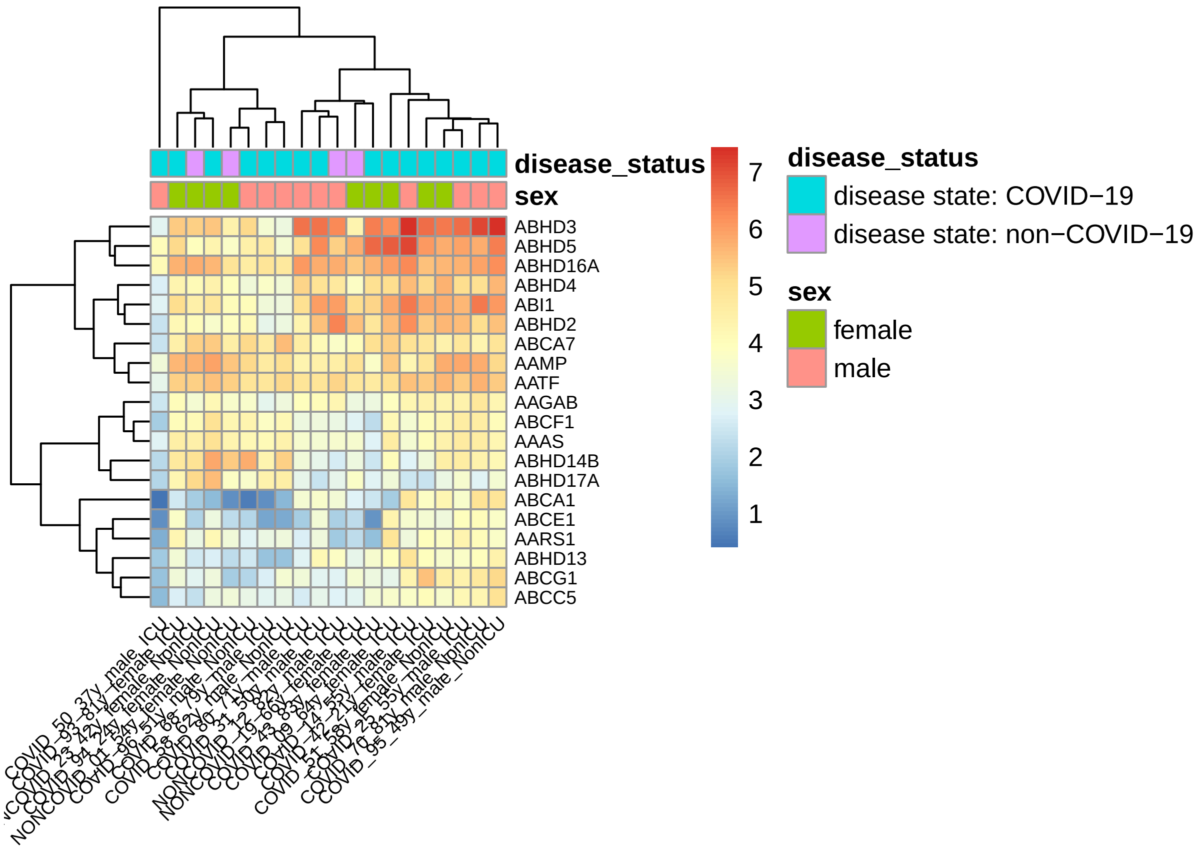
```
# boxplot)
ggplot(filter(df_plot, !is.na(expr), !is.na(sex), !is.na(disease_status)),
       aes(x = sex, y = expr, fill = disease_status)) +
  geom_boxplot(outlier.alpha = 0.6) +
  labs(
    title = paste(gene_main, "expression by Sex and Disease Status"),
    x = "Sex", y = paste(gene_main, "expression"), fill = "Disease status"
  ) +
  theme_minimal()
```



5. 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

```
#select top 20 most variable genes
var_by_gene <- apply(gene_data, 1, var, na.rm = TRUE)
top_genes <- names(sort(var_by_gene, decreasing = TRUE))[1:20]
#build expression matrix: rows = genes, columns = samples
expr_mat <- as.matrix(gene_data[top_genes, ])
expr_mat_log2 <- log2(expr_mat + 1)  # log2 transform to enhance contrast

#Randomly select sample columns
set.seed(123)
n_samples <- 20
cand_ids  <- intersect(colnames(gene_data), meta_sel$SampleID)
sample_ids <- sample(cand_ids, size = n_samples)

#Subset matrix with sampled samples, keep order consistent
expr_mat_log2 <- expr_mat_log2[, sample_ids, drop = FALSE]

#Build annotation
anno <- meta_sel %>%
  filter(SampleID %in% sample_ids) %>%
  select(SampleID, sex, disease_status) %>%
  column_to_rownames("SampleID") %>%
  .[colnames(expr_mat_log2), , drop = FALSE]
```

```r
#Plot heatmap
pheatmap(expr_mat_log2,
         annotation_col = anno,# tracking bars
         show_rownames = TRUE,
         show_colnames = TRUE,
         clustering_distance_rows = "euclidean",
         clustering_distance_cols = "euclidean",
         fontsize_row = 7,
         fontsize_col = 7,
         angle_col = 45
)
```

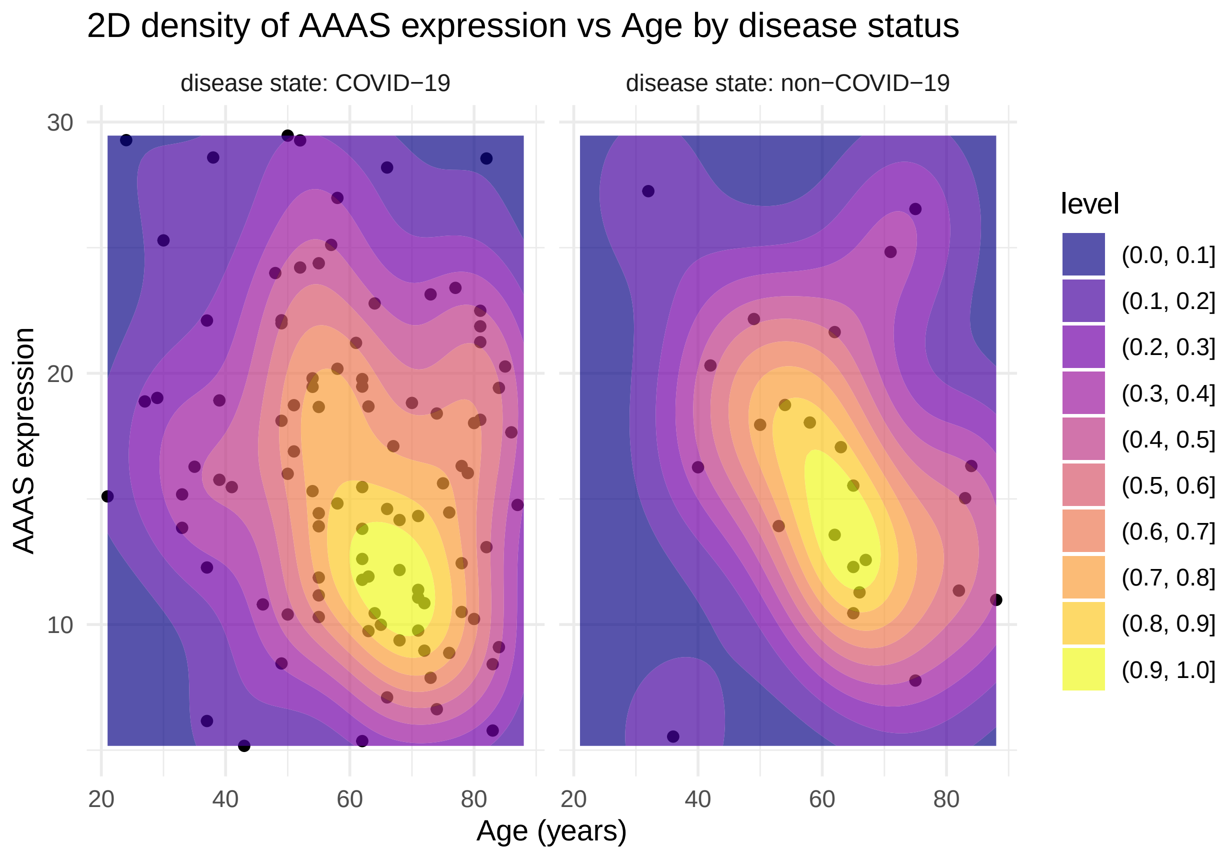


6. Going through the documentation for ggplot2, generate a plot type that we did not previously discuss in class that describes your data in a new and unique way (5 pts)

```r
df_main %>%
  filter(!is.na(disease_status)) %>% # remove rows with missing disease_status
  ggplot(aes(x = age, y = expr)) +
  geom_point(alpha = 1.0, size = 1.5, color = "black") + # plot raw points
  stat_density_2d_filled(alpha = 0.7, contour_var = "ndensity") +
  facet_wrap(~ disease_status) +
  labs(
    title = paste("2D density of", gene_main, "expression vs Age by disease status"),
    x = "Age (years)", # label x-axis
    y = paste(gene_main, "expression")# label y-axis
  ) +
  scale_fill_viridis_d(option = "plasma") +
  theme_minimal()
```

```
## Warning: Removed 3 rows containing non-finite outside the scale range
## (`stat_density2d_filled()`).
```

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


2D density of AAAS expression vs Age by disease status
disease state: COVID−19
disease state: non−COVID−19
AAAS expression
Age (years)
level
(0.0, 0.1]
(0.1, 0.2]
(0.2, 0.3]
(0.3, 0.4]
(0.4, 0.5]
(0.5, 0.6]
(0.6, 0.7]
(0.7, 0.8]
(0.8, 0.9]
(0.9, 1.0]