

R final submission

Ruiqi Li

2025-08-18

Final Project

Before class on 8/21, you should submit (1) your compiled PDF from LaTeX, (2) your corresponding .tex file, (3) a knitted PDF of the R markdown file with your code for the final submission, and (4) a link to your now public facing GitHub repository.

```
#Final subission
#Load gene expression data
gene_data <- read.csv(file = "QBS103_GSE157103_genes.csv",row.names=1)

#Load metadata for participants
series <- read.csv(file = "QBS103_GSE157103_series_matrix-1.csv")
str(series)
```

```
## 'data.frame': 126 obs. of 25 variables:
## $ participant_id : chr "COVID_01_39y_male_NonICU" "COVID_02_63y_male_NonICU" ...
## $ geo_accession : chr "GSM4753021" "GSM4753022" "GSM4753023" "GSM4753024" ...
## $ status : chr "Public on Aug 29 2020" "Public on Aug 29 2020" "Public on Aug 29 2020" ...
## $ X.Sample_submission_date : chr "Aug 28 2020" "Aug 28 2020" "Aug 28 2020" "Aug 28 2020" ...
## $ last_update_date : chr "Aug 29 2020" "Aug 29 2020" "Aug 29 2020" "Aug 29 2020" ...
## $ type : chr "SRA" "SRA" "SRA" "SRA" ...
## $ channel_count : int 1 1 1 1 1 1 1 1 1 1 ...
## $ source_name_ch1 : chr "Leukocytes from whole blood" "Leukocytes from whole blood" ...
## $ organism_ch1 : chr "Homo sapiens" "Homo sapiens" "Homo sapiens" "Homo sapiens" ...
## $ disease_status : chr "disease state: COVID-19" "disease state: COVID-19" "disease state: COVID-19" ...
## $ age : chr "39" "63" "33" "49" ...
## $ sex : chr " male" " male" " male" " male" ...
## $ icu_status : chr " no" " no" " no" " no" ...
## $ apacheii : chr "15" " unknown" " unknown" " unknown" ...
## $ charlson_score : int 0 2 2 1 1 1 7 2 1 ...
## $ mechanical_ventilation : chr " yes" " no" " no" " no" ...
## $ ventilator.free_days : int 0 28 28 28 23 28 28 0 0 2 ...
## $ hospital.free_days_post_45_day_followup : int 0 39 18 39 27 36 42 0 0 0 ...
## $ ferritin.ng.ml. : chr "946" "1060" "1335" "583" ...
## $ crp.mg.l. : chr "73.1" " unknown" "53.2" "251.1" ...
## $ ddimer.mg.l_feu. : chr "1.3" "1.03" "1.48" "1.32" ...
## $ procalcitonin.ng.ml.. : chr "36" "0.37" "0.07" "0.98" ...
## $ lactate.mmol.l. : chr "0.9" " unknown" " unknown" "0.87" ...
## $ fibrinogen : chr "513" "unknown" "513" "949" ...
## $ sofa : chr "8" " unknown" " unknown" " unknown" ...
```

```
dim(series)
```

```
## [1] 126 25
```

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

```
only_num <- function(x) {
  is_num <- grepl("^[:space:]*[0-9]+(?:\\. [0-9]+)?[:space:]*$", x)
  out <- rep(NA_real_, length(x))
  out[is_num] <- as.numeric(trimws(x[is_num]))
  out
}

df <- series %>%
  mutate(
    # Remove any extra spaces from 'sex' variable
    sex = trimws(sex),
    # Recode ICU status to standardized labels
    icu_status = ifelse(tolower(trimws(icu_status)) == "yes", "ICU", "NonICU"),
    # Convert 'age' to numeric
    age = only_num(age),
    # Convert ferritin to numeric (ignore warnings if conversion fails)
    ferritin = only_num(ferritin.ng.ml.),
    # Convert CRP to numeric (ignore warnings)
    crp = only_num(crp.mg.l.),
    # Recode mechanical ventilation variable
    mech_vent = ifelse(trimws(mechanical_ventilation) == "yes", "Yes", "No")
  )

# Summarize categorical variables (sex and mechanical ventilation) by ICU status
df %>%
  group_by(icu_status) %>%
  summarise(
    n_sex_female = sum(sex == "female", na.rm=TRUE), # Count number of females
    n_sex_male = sum(sex == "male", na.rm=TRUE), # Count number of males
    n_mechvent_yes = sum(mech_vent == "Yes", na.rm=TRUE), # Count patients with mechanical ventilation
    n_mechvent_no = sum(mech_vent == "No", na.rm=TRUE), # Count patients without mechanical ventilation
    .groups="drop"
  )
```

```
## # A tibble: 2 x 5
```

```
##   icu_status n_sex_female n_sex_male n_mechvent_yes n_mechvent_no
##   <chr>      <int>      <int>      <int>      <int>
## 1 ICU          24         41         46         20
## 2 NonICU       27         33         5         55
```

```
# Summarize continuous variables (age, ferritin, CRP) by ICU status
df %>%
  group_by(icu_status) %>%
```

```

summarise(
  age_mean = mean(age, na.rm=TRUE), # Calculate mean age
  age_sd   = sd(age, na.rm=TRUE),   # Calculate standard deviation of age
  ferr_median = median(ferritin, na.rm=TRUE), # Median ferritin level
  ferr_IQR1 = quantile(ferritin, 0.25, na.rm=TRUE), # 25th percentile (Q1) of ferritin
  ferr_IQR3 = quantile(ferritin, 0.75, na.rm=TRUE), # 75th percentile (Q3) of ferritin
  crp_mean = mean(crp, na.rm=TRUE), # Mean CRP level
  crp_sd   = sd(crp, na.rm=TRUE)   # Standard deviation of CRP
)

```

```

## # A tibble: 2 x 8
##   icu_status age_mean age_sd ferr_median ferr_IQR1 ferr_IQR3 crp_mean crp_sd
##   <chr>      <dbl>  <dbl>      <dbl>      <dbl>      <dbl>      <dbl>  <dbl>
## 1 ICU          63.5   14.0         685         325        1212        150.   106.
## 2 NonICU       58.7   17.8         401         131         870        109.   94.4

```

Generate final a publication quality histogram, scatter plot, and boxplot from submission 1 (i.e. only for your first gene of interest) (5 pts)

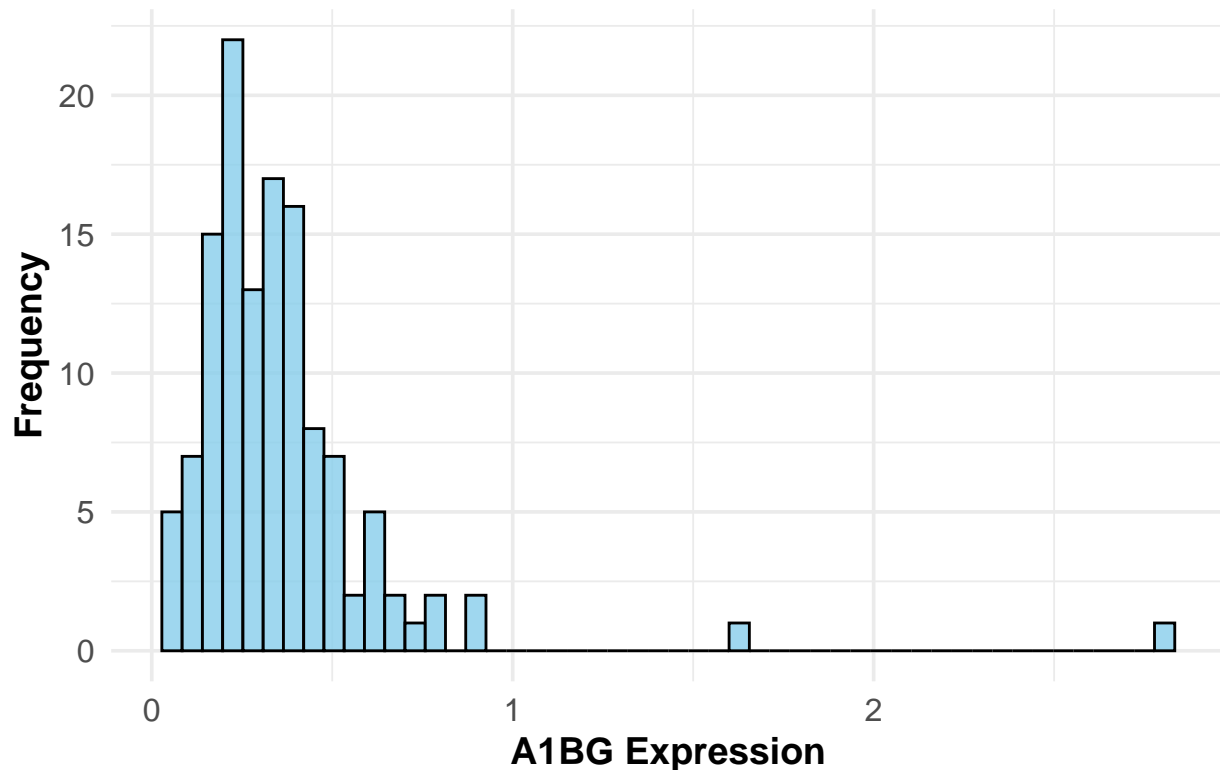
```

#1.
#Histogram of gene expression
#Select the first gene (A1BG) for analysis and convert to numeric vector
new_gene <- gene_data[1, ]

#Plot histogram for A1BG expression values across all participants
ggplot(data.frame(value = as.numeric(new_gene)), aes(x = value)) +
  geom_histogram(bins = 50, color = "black", fill = "skyblue", alpha = 0.8) +
  labs(
    title = "Histogram of A1BG Expression",
    x = "A1BG Expression",
    y = "Frequency"
  ) +
  theme_minimal(base_size = 14) +
  theme(
    plot.title = element_text(size = 18, face = "bold", hjust = 0.5),
    axis.title = element_text(size = 14, face = "bold"),
    axis.text = element_text(size = 12)
  )

```

Histogram of A1BG Expression



```
#2.
#Scatterplot of gene expression vs age
#Pivot gene expression row into long format for merging
gene_line1 <- new_gene %>%
  pivot_longer(cols = everything(),names_to = "participant_id",values_to = "A1BG_value")

# Merge with metadata by participant_id to obtain age and other covariates
new_df<-merge(series,gene_line1,by="participant_id")

#Prepare age levels for the x-axis: numeric ages in ascending order, then special categories
num_ages <- sort(as.numeric(unique(new_df$age)[!grepl("^0-9", unique(new_df$age))]))
special_ages <- unique(new_df$age)[grepl("^0-9", unique(new_df$age))]
age_levels <- c(as.character(num_ages), special_ages)

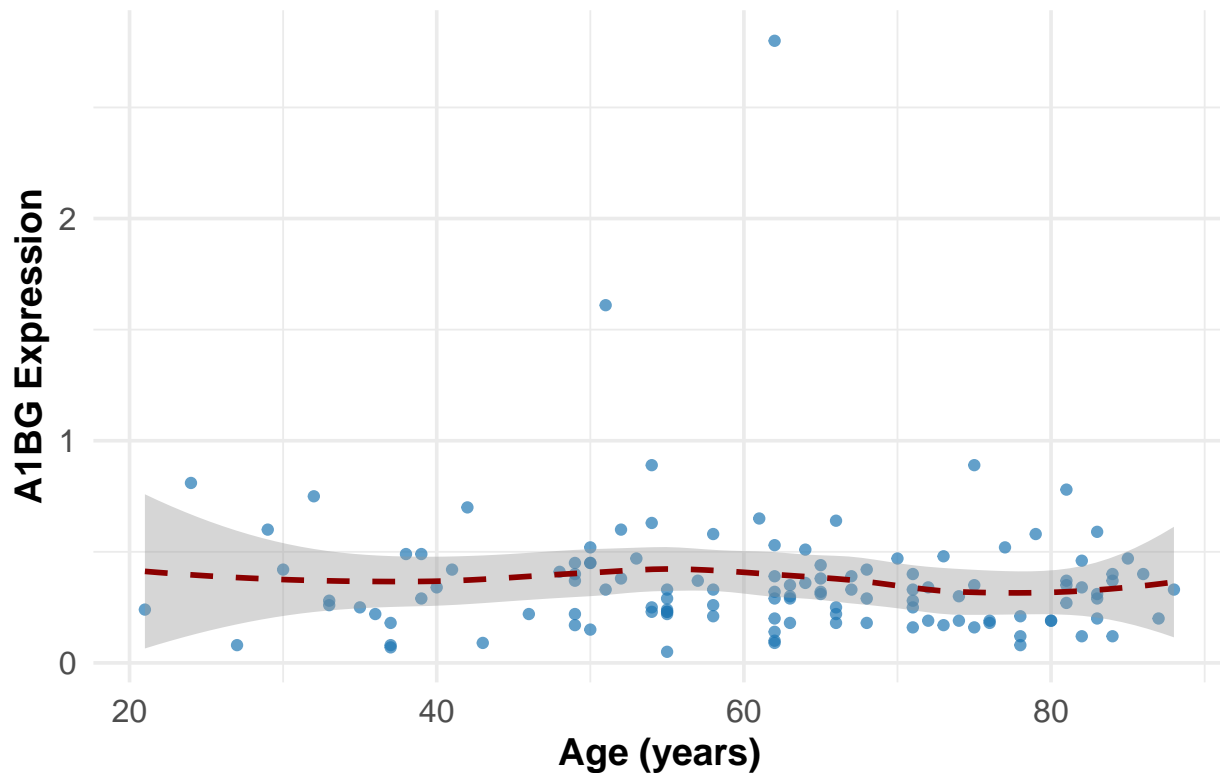
# Scatterplot: A1BG expression vs age (x-axis shows all ages in specified order)
new_df <- new_df %>%
  mutate(age_num = suppressWarnings(as.numeric(trimws(age)))) %>% # avoid coercion warnings
  filter(is.finite(age_num), is.finite(A1BG_value)) # drop NAs/Inf before ggplot

ggplot(new_df, aes(x = age_num, y = A1BG_value)) +
  geom_point(alpha = 0.7, color = "#1F77B4") +
  geom_smooth(method = "loess", se = TRUE, color = "darkred", linetype = "dashed") +
  labs(
    title = "Scatterplot of A1BG Expression versus Age",
    x = "Age (years)",
    y = "A1BG Expression"
```

```
) +
theme_minimal(base_size = 14) +
theme(
  plot.title = element_text(size = 18, face = "bold", hjust = 0.5),
  axis.title = element_text(size = 14, face = "bold"),
  axis.text = element_text(size = 12)
)
```

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

Scatterplot of A1BG Expression versus Age

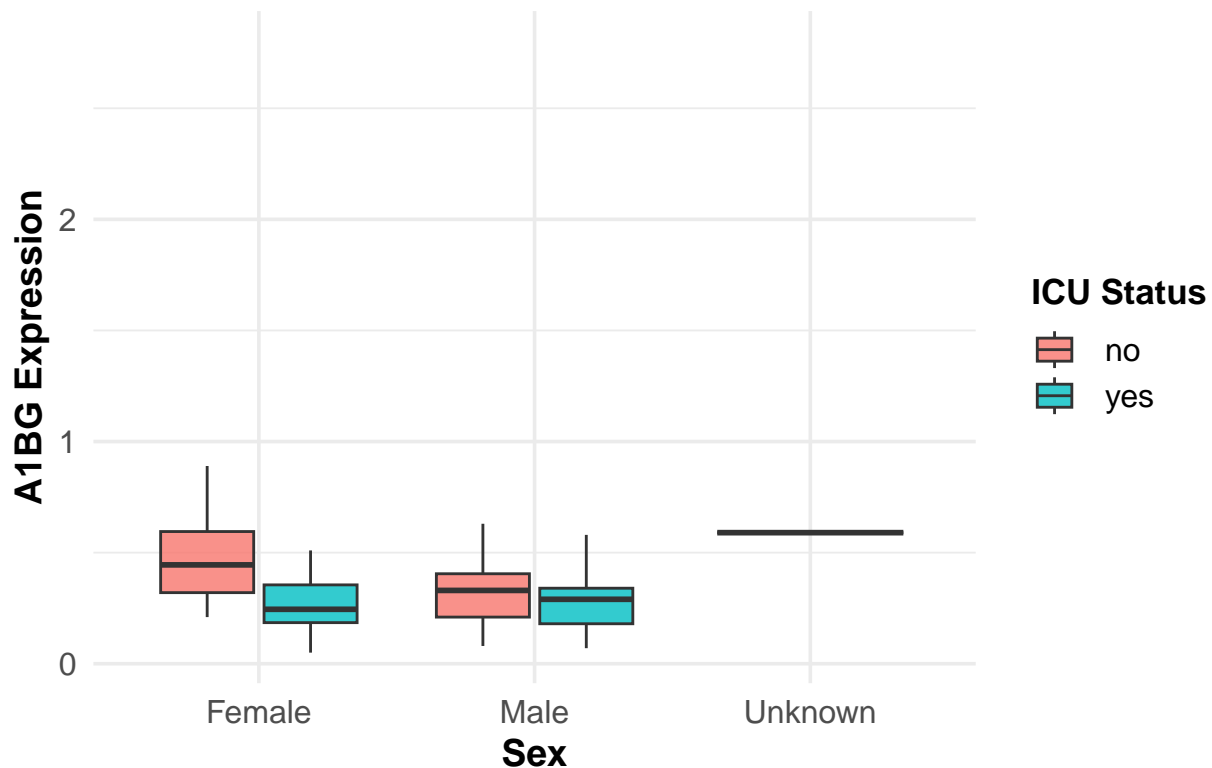


```
#3.
#Boxplot of gene expression by sex and ICU status
#sex: categorical variable; icu_status: categorical variable
new_df <- new_df %>%
  mutate(
    sex = tolower(trimws(sex)),
    sex = recode(sex,
      "female" = "Female",
      "male" = "Male",
      "unknown" = "Unknown"),
    sex = factor(sex, levels = c("Female", "Male", "Unknown"))
  )

ggplot(new_df, aes(x = sex, y = A1BG_value, fill = icu_status)) +
  geom_boxplot(alpha = 0.8, outlier.shape = NA) +
```

```
labs(
  title = "A1BG Expression by Sex and ICU Status",
  x = "Sex",
  y = "A1BG Expression",
  fill = "ICU Status"
) +
theme_minimal(base_size = 14) +
theme(
  plot.title = element_text(size = 18, face = "bold", hjust = 0.5),
  axis.title = element_text(size = 14, face = "bold"),
  axis.text = element_text(size = 12),
  legend.title = element_text(size = 13, face = "bold"),
  legend.text = element_text(size = 12)
)
```

A1BG Expression by Sex and ICU Status



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

```
# Convert gene expression data into a numeric matrix
mat <- as.matrix(gene_data)

# Find overlapping participant IDs between gene expression and metadata
common <- intersect(colnames(mat), series$participant_id)

# Keep only common participants in the gene expression matrix
mat <- mat[, common, drop = FALSE]
```

```

# Subset metadata for the same participants and align the order
series_sub <- series %>% filter(participant_id %in% common) %>%
  arrange(match(participant_id, colnames(mat)))

# Log2-transform the expression values to stabilize variance
mat_log <- log2(mat + 1)

# Select the top 20 most variable genes across samples
topN <- 16
top_idx <- order(apply(mat_log, 1, var, na.rm = TRUE), decreasing = TRUE)[1:topN]
sub <- mat_log[top_idx, , drop = FALSE]

# Clean up metadata variables: sex and ICU status
sex_clean <- trimws(series_sub$sex)
icu_raw <- trimws(series_sub$icu_status)

# Convert ICU status to binary factor
icu_clean <- ifelse(tolower(icu_raw) == "yes", "ICU", "NonICU")

# Convert sex into male/female categories
sex_clean <- ifelse(grepl("^f", tolower(sex_clean)), "Female", "Male")

# Prepare annotation dataframe for heatmap
annotationData <- data.frame(
  Sex = factor(sex_clean, levels = c("Female", "Male")),
  ICU = factor(icu_clean, levels = c("ICU", "NonICU")),
  row.names = series_sub$participant_id
)

# Define color scheme for annotations
annotationColors <- list(
  Sex = c(Female = "#E377C2", Male = "#1F77B4"),
  ICU = c(ICU = "#D62728", NonICU = "#2CA02C")
)

# Randomly select 25 samples for visualization
set.seed(100)
subset_cols <- sample(colnames(sub), 30)
sub25 <- sub[, subset_cols, drop = FALSE]
ann25 <- annotationData[subset_cols, , drop = FALSE]

# Shorten column labels for readability (keep ID, sex, and ICU status)
short_labels <- ifelse(
  grepl("^COVID", colnames(sub25)),
  sub("^COVID_(\\d+)_(\\d+y_([_]+)_(ICU|NonICU))$",
    "\\1_\\2_\\3", colnames(sub25)),
  sub("^NONCOVID_(\\d+)_(\\d+y_([_]+)_(ICU|NonICU))$",
    "N\\1_\\2_\\3", colnames(sub25))
)

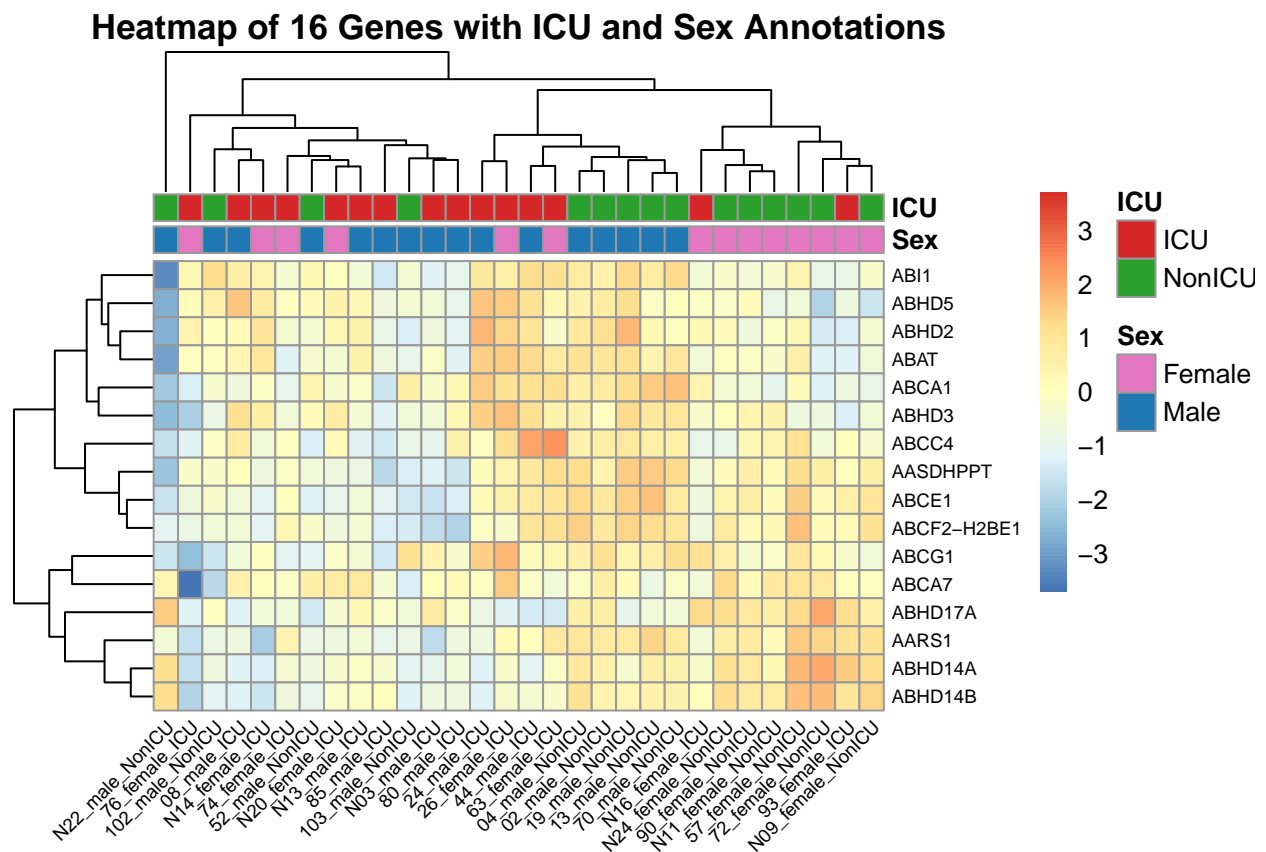
# Draw heatmap of top variable genes with sample annotations
pheatmap(
  sub25,

```

```

scale = "row",                                # scale genes across samples
clustering_distance_cols = "euclidean",        # distance metric for columns
clustering_distance_rows = "euclidean",        # distance metric for rows
clustering_method = "complete",               # hierarchical clustering method
annotation_col = ann25,                       # add sample metadata as annotation
annotation_colors = annotationColors,          # use predefined colors
show_colnames = TRUE,                         # display sample labels
labels_col = short_labels,                    # use shortened labels
fontsize_col = 7,                             # font size for column labels
angle_col = 45,                              # rotate labels for clarity
fontsize_row = 7,                             # font size for row labels
main = "Heatmap of 16 Genes with ICU and Sex Annotations"
)

```



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)

```

# Select top N genes with highest variance across samples
topN <- 50
top_idx <- order(apply(mat_log, 1, var, na.rm = TRUE), decreasing = TRUE)[1:topN]
mat_top <- mat_log[top_idx, , drop = FALSE]

# PCA on samples (transpose), centered & scaled
pca <- prcomp(t(mat_top), center = TRUE, scale. = TRUE)

# First two PCs to data frame

```



```

pca_df <- as.data.frame(pca$x[, 1:2])
pca_df$participant_id <- rownames(pca_df)

# Clean metadata; standardize labels
meta <- series_sub %>%
  mutate(
    sex = tolower(trimws(sex)),
    sex = ifelse(sex %in% c("female", "male"), sex, "unknown"),
    icu_status = ifelse(tolower(trimws(icu_status)) == "yes", "ICU", "NonICU")
  ) %>%
  select(participant_id, sex, icu_status)

# Join and coerce to factors (good for legends/shapes)
pca_df <- dplyr::left_join(pca_df, meta, by = "participant_id") %>%
  mutate(
    sex = recode(sex, "female" = "Female", "male" = "Male", "unknown" = "Unknown"),
    sex = factor(sex, levels = c("Female", "Male", "Unknown")),
    icu_status = factor(icu_status, levels = c("ICU", "NonICU"))
  )

# Variance explained for axis labels
var_exp <- round(100 * (pca$sdev^2 / sum(pca$sdev^2))) [1:2]

# Plot: map color globally, shape only on points (no warning from stat_ellipse)
ggplot(pca_df, aes(PC1, PC2, color = icu_status)) +
  geom_point(aes(shape = sex), size = 2.5, alpha = 0.9) +
  stat_ellipse(aes(group = icu_status), linetype = 2) +
  labs(
    title = "PCA of Top-Variance Genes",
    x = paste0("PC1 (", var_exp[1], "%)"),
    y = paste0("PC2 (", var_exp[2], "%)"),
    color = "ICU", shape = "Sex"
  ) +
  theme_minimal(base_size = 14) +
  theme(
    plot.title = element_text(size = 18, face = "bold", hjust = 0.5),
    axis.title = element_text(size = 14, face = "bold"),
    axis.text = element_text(size = 12),
    legend.title = element_text(size = 13, face = "bold"),
    legend.text = element_text(size = 12)
  )

```

