Project_Final

Luwei Zeng

2024-08-29

Contents

Generate a table formatted in LaTeX of summary statistics for 3 continuous variables and 3 categorical variables. 3 continuous variables: ferritin.ng.ml., crp.mg.l., fibrinogen, 3 categorial variables: sex, icu_status, mechanical_ventilation, stratifying by the categorical variable of icu_status.

The summary table reports n (%) for categorical variables and report mean (sd) or median [IQR] for continuous variables.

```
library(tidyverse)
library(knitr) # for base kable function
```

```
#Load data
setwd("/Users/hekaiwei/Desktop/R class/project")
gene <- read.csv(file = "QBS103_GSE157103_genes.csv",row.names=1)
metadata <- read.csv(file = "QBS103_GSE157103_series_matrix.csv", row.names = 1)

# Select relevant columns
selected_data <- metadata %>%
  select(ferritin.ng.ml., crp.mg.l., fibrinogen, sex, icu_status, mechanical_ventilation)

selected_data$ferritin.ng.ml. <- as.numeric(selected_data$ferritin.ng.ml.)
selected_data$crp.mg.l. <- as.numeric(selected_data$crp.mg.l.)
selected_data$fibrinogen <- as.numeric(selected_data$fibrinogen)
```

```
# Function for continuous variables
contSummary <- function(x, normal = TRUE) {

  #if normal, calculate mean and sd
  if (normal==T) {
    myMean <- round(mean(x, na.rm = TRUE),0)
    mySD <- round(sd(x, na.rm = TRUE), 0)
    paste0(myMean, " (", mySD, ")")
  }

  #if not normal, calculate median and IQR
  else {
    myMedian <- round(median(x, na.rm = TRUE), 0)
    myIQR1 <- round(quantile(x, 1/4, na.rm = TRUE), 0)
    myIQR2 <- round(quantile(x, 3/4, na.rm = TRUE), 0)
    paste0(myMedian, " [", myIQR1, ", ", myIQR2, "]")
```

```
  }
}


# Function for categorical variables
catSummary <- function(x) {
  table_x <- table(x) #count the frequency of each category in the categorical variable x
  prop_x <- prop.table(table_x) * 100 #calculate the proportion of for each category.

  # Create a summary string that combines the category names, counts, and percentages.
  # Each category's information is separated by a semicolon.
  summary <- paste0(names(table_x), ": ", table_x, " (", round(prop_x, 0), "%)", collapse = "; ")

  return(summary)
}


# Perform Shapiro-Wilk normality tests to check if these three continuous variables are normal
normal_ferritin <- shapiro.test(selected_data$ferritin.ng.ml.)$p.value > 0.05
normal_crp <- shapiro.test(selected_data$crp.mg.l.)$p.value > 0.05
normal_fibrinogen <- shapiro.test(selected_data$fibrinogen)$p.value > 0.05


# Generate the summary table and stratified by 'icu_status'
summary_table <- selected_data %>%
  group_by(icu_status) %>%
  summarise(
    Ferritin = contSummary(ferritin.ng.ml., normal = normal_ferritin),
    CRP = contSummary(crp.mg.l., normal = normal_crp),
    Fibrinogen = contSummary(fibrinogen, normal = normal_fibrinogen),
    Sex = catSummary(sex),
    Mechanical_Ventilation = catSummary(mechanical_ventilation)
  )

summary_table
```

```
## # A tibble: 2 x 6
##   icu_status Ferritin        CRP         Fibrinogen Sex   Mechanical_Ventilation
##   <chr>      <chr>           <chr>       <chr>      <chr> <chr>
## 1 " no"      401 [131, 870]  109 [38, 1~ 501 [399,~ " fe~ " no: 55 (92%);  yes:~
## 2 " yes"     685 [325, 1212] 136 [52, 2~ 489 [317,~ " fe~ " no: 20 (30%);  yes:~
```

```
tab <- kable(summary_table,
             caption = 'Summary Table',
             format = 'latex',
             booktabs = T,
             col.names = c("ICU Status", "Ferritin", "CRP", "Fibrinogen",
                           "Sex", "Mechanical Ventilation"),
             align = 'l')
tab
```

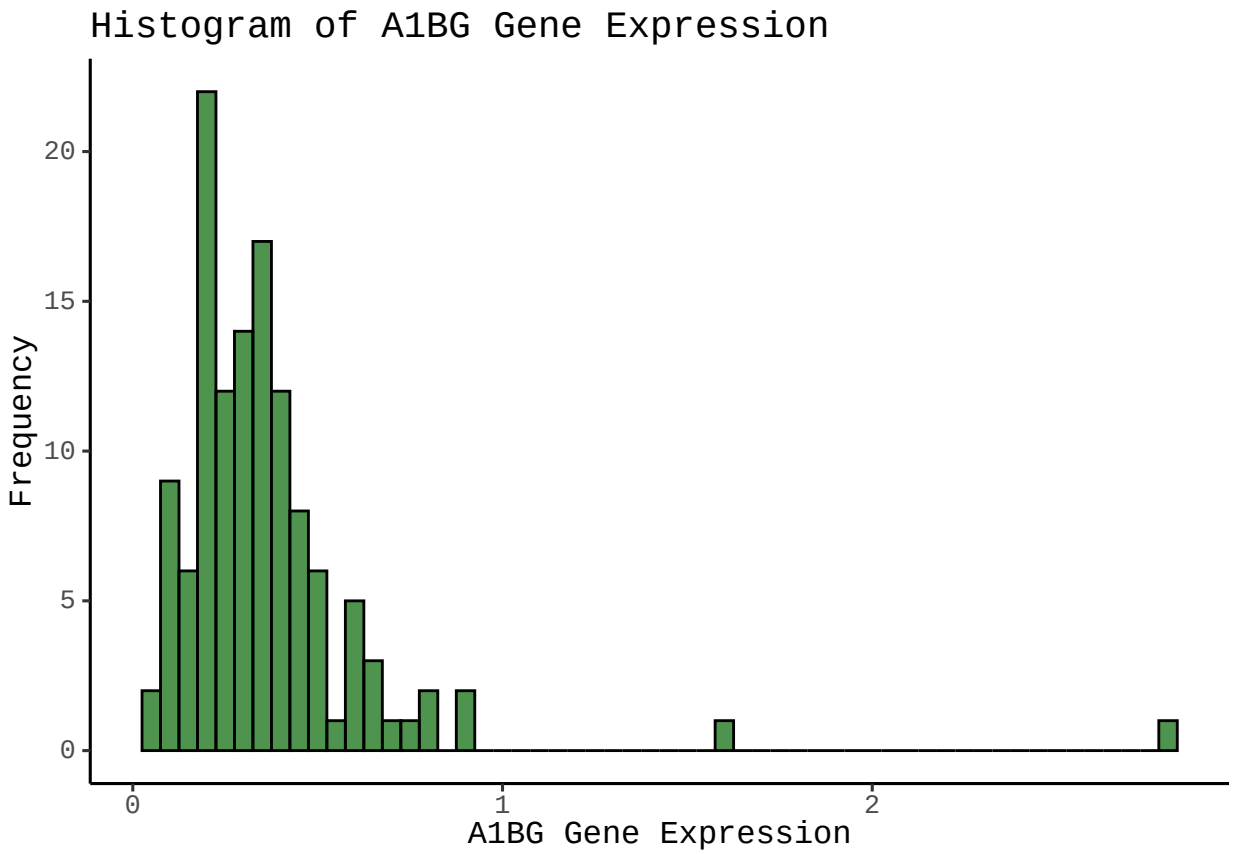
Table 1: Summary Table

ICU Status	Ferritin	CRP	Fibrinogen	Sex	Mec
no	401 [131, 870]	109 [38, 146]	501 [399, 636]	female: 27 (45%); male: 33 (55%)	no:
yes	685 [325, 1212]	136 [52, 233]	489 [317, 654]	female: 24 (36%); male: 41 (62%); unknown: 1 (2%)	no:

```
#convert to dataframe and switch column and row
A1BG_gene <- gene["A1BG", ]
A1BG_gene <- as.data.frame(t(A1BG_gene))
#head(A1BG_gene)

#combine A1BG gene expression and metadata
data_combined<-merge(A1BG_gene,metadata,by = "row.names")
colnames(data_combined)[1] <- "ID" #set the 1st column name as "ID"
#head(data_combined$icu_status)
```

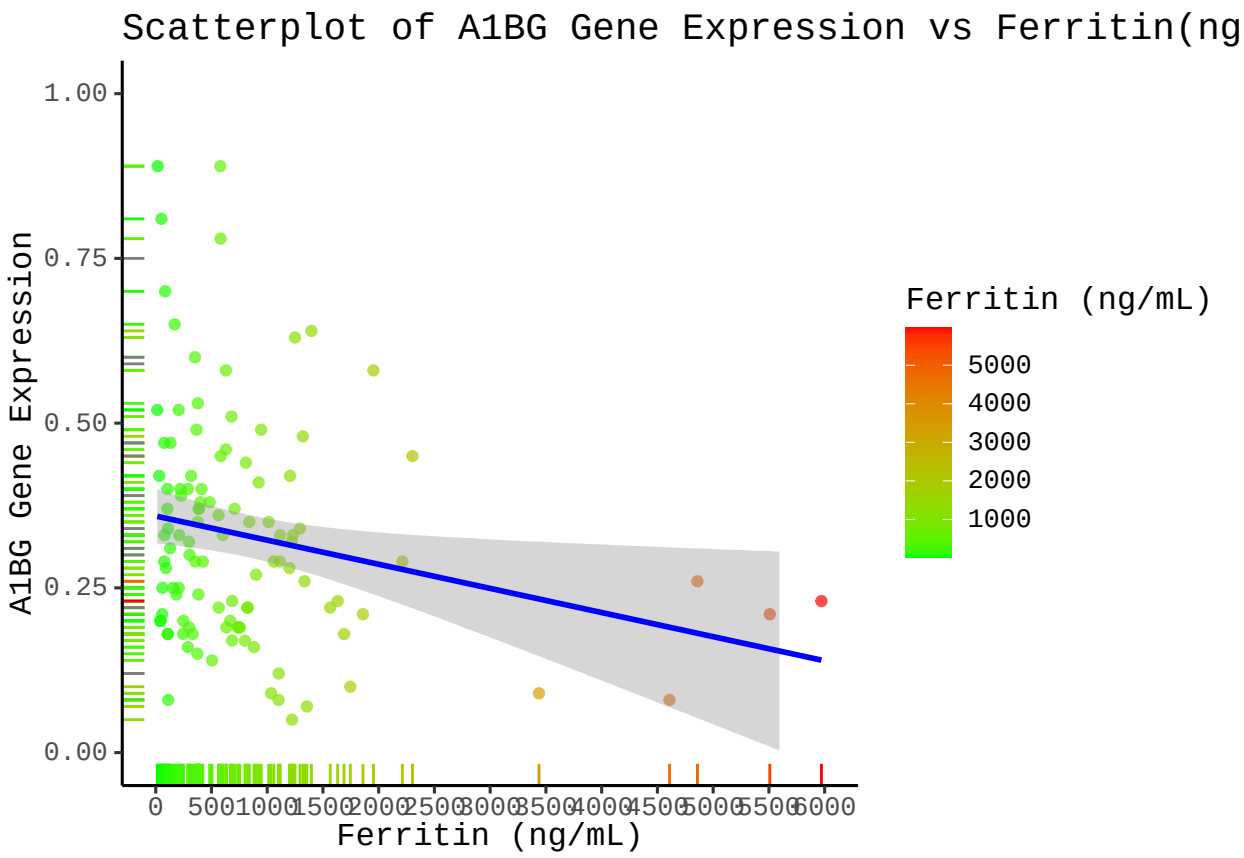
Histogram of A1BG Gene Expression:

```
#One gene:AIBG
#Define the overall plot with 'data_combined' as the data source and map 'A1BG' to the x-axis
ggplot(data_combined,aes(x = A1BG)) +
  #Add histogram data with specific bin width, fill color, and border color
  geom_histogram(binwidth=0.05,fill="darkgreen",color="black",alpha=0.7)+
  labs(title="Histogram of A1BG Gene Expression",
     x="A1BG Gene Expression",
     y="Frequency")+
  theme_classic(base_family = 'Courier',base_size = 12)
```



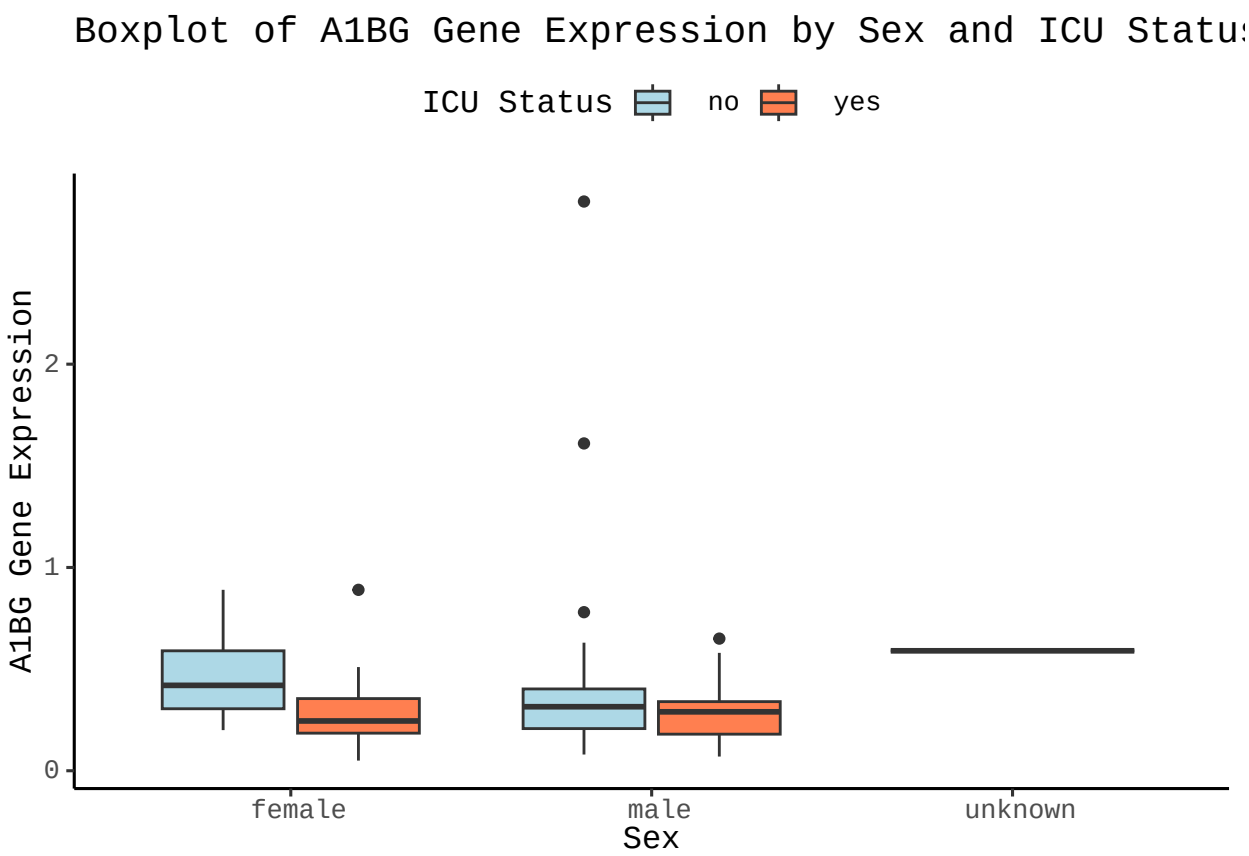
Scatterplot of A1BG Gene Expression vs Ferritin Level:

```
#One continuous covariate:ferritin(ng/ml)

#Ensure the column 'ferritin.ng.ml.' is of numeric type
data_combined$ferritin.ng.ml. <- as.numeric(data_combined$ferritin.ng.ml.)

#Define the overall plot with 'data_combined' as the data source and map 'ferritin.ng.ml.' to the x-axi
ggplot(data_combined, aes(x = ferritin.ng.ml., y = A1BG,color=ferritin.ng.ml.)) +
  geom_point(alpha=0.7) +
  geom_smooth(method = "lm", color = "blue")+ #add trendline
  geom_rug(sides="bl")+ #visualize the density of the data
  labs(title="Scatterplot of A1BG Gene Expression vs Ferritin(ng/ml)",
       x = "Ferritin (ng/mL)",
       y = "A1BG Gene Expression",
       color = "Ferritin (ng/mL)") +
  ylim(0,1)+#Define the scale for the x-axis, setting limits from 0 to 6000 and breaks at intervals of
  scale_x_continuous(limits = c(0, 6000), breaks = seq(0, 6000, by = 500)) +
  scale_color_gradient(low = "green", high = "red") +
  theme_classic(base_family = 'Courier',base_size = 12)
```



Boxplot of gene expression separated by Sex and ICU Status:

```
#Two categorical covariates sex, icu status
#Define the overall plot with 'data_combined' as the data source and map 'sex' to the x-axis, 'A1BG' to
ggplot(data_combined, aes(x = sex, y = A1BG, fill = icu_status)) +
  # Add boxplot
  geom_boxplot()+
  #Change labels for the title, x-axis, y-axis, and fill legend
  labs(title = "Boxplot of A1BG Gene Expression by Sex and ICU Status",
       x = "Sex",
       y = "A1BG Gene Expression",
       fill = "ICU Status") +
  scale_fill_manual(values = c(" no" = "lightblue", " yes" = "coral")) +
  #Define the theme as classic with 'Courier' font and base font size of 10
  theme_classic(base_family = 'Courier',base_size = 12)+
  # Customize the theme to position the legend at the top of the plot
  theme(legend.position = 'top')
```



Heatmap of across 10 Gense Stratified by ICU Status and Sex:

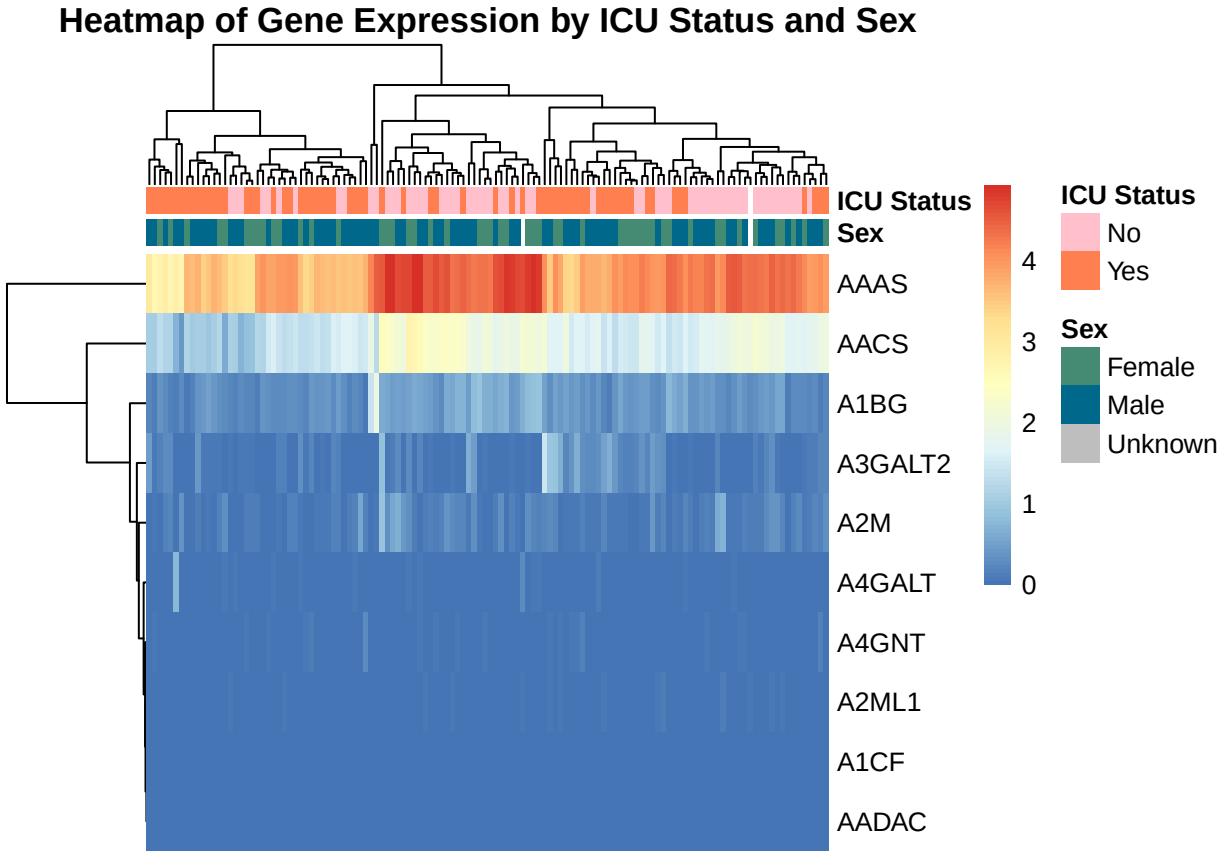
```
library(pheatmap)
```

```
selected_genes_hm<-c("A1BG","A1CF","A2M","A2ML1","A3GALT2","A4GALT","A4GNT","AAAS","AACS","AADAC") #sel
gene_data_hm<-gene[selected_genes_hm,]
gene_data_hm <- as.data.frame(t(gene_data_hm)) #convert to dataframe and switch column and row

#combine selected genes expression and metadata
data_combined_hm<-merge(gene_data_hm,metadata,by = "row.names")
colnames(data_combined_hm)[1] <- "ID" #set the 1st column name as "ID"
```

```
# Log2-normalize the gene expression data
#log2_gene_data_hm <- log2(gene_data_hm + 1)  # Adding 1 to avoid log2(0)

log2_gene_data_hm <- log2(t(gene_data_hm + 1))
log2_gene_data_hm <- as.data.frame(log2_gene_data_hm)

# Define covariate for tracking bar (sex and icu_status) for rows
annotationData <- data_combined_hm %>%
  select(ID, sex, icu_status) %>%
  column_to_rownames("ID")


# Rename columns to 'Sex' and 'ICU Status'
```

```r
annotationData <- annotationData %>%
  rename(Sex = sex, 'ICU Status' = icu_status)
```

```r
# Convert factors to factor levels
annotationData$Sex <- factor(annotationData$Sex, levels = c(" female", " male"," NA"), labels = c("Fema
annotationData$'ICU Status' <- factor(annotationData$'ICU Status', levels = c(" no", " yes"), labels = 

# Define colors for the annotation tracks
annotationColors <- list(
  `Sex` = c('Female' = 'aquamarine4', 'Male' = 'deepskyblue4','Unknown' = 'gray'),
  `ICU Status` = c('No' = 'pink', 'Yes' = 'coral')
)



# Generate heatmap
pheatmap(log2_gene_data_hm,
         clustering_distance_cols = 'euclidean',
         clustering_distance_rows = 'euclidean',
         annotation_col= annotationData,
         annotation_colors = annotationColors,
         show_colnames =FALSE, #Hide column names
         main = "Heatmap of Gene Expression by ICU Status and Sex")
```



Ridge Plot of Ferritin Levels by ICU Status:

```r
# Load ggridges
library(ggridges)

# Convert necessary columns to numeric and factor
data_combined$ferritin.ng.ml. <- as.numeric((data_combined$ferritin.ng.ml.))
data_combined$icu_status <- factor(data_combined$icu_status, levels = c(" no", " yes"), labels = c("No"
```

```r
# Generate a ridge plot for Ferritin levels by ICU Status
ggplot(data_combined, aes(x = ferritin.ng.ml., y = icu_status, fill = icu_status)) +
  geom_density_ridges(alpha = 0.8, scale = 1) +  # Add ridge plot with transparency and scale adjustmen
  scale_fill_manual(values = c("No" = "lightblue", "Yes" = "coral")) +  # Customize fill colors
  labs(title = "Distribution of Ferritin Levels by ICU Status",
       x = "Ferritin (ng/mL)",
       y = "ICU Status",
       fill = "ICU Status") +
  scale_x_continuous(limits = c(0, 6000), breaks = seq(0, 6000, by = 500)) +
  theme_classic(base_family = 'Courier',base_size = 12)+
  # Customize the theme to position the legend at the top of the plot
  theme(legend.position = 'top')
```