R_final_V3_FINAL_SUBMISSION

2025-08-16

import

```
library(ggplot2) #import lib and data
genes <- read.csv("QBS103_GSE157103_genes.csv", row.names = 1)
metadata <- read.csv("QBS103_GSE157103_series_matrix-1.csv")
```

redefining the plotting function from v2 to generate publication ready plots

```
plot_all_figures <- function(metadata, gene_name, cont_var, cat_var1, cat_var2) {

  temp <- metadata  # make a copy
  temp[[gene_name]] <- as.numeric(genes[gene_name, ])

  # remove row wehre age is : and convert >89 to 90
  #trim whitespace to be able to find :
  #from  https://www.rdocumentation.org/packages/base/versions/3.6.2/topics/trimws
  metadata <- metadata[trimws(metadata[[cont_var]]) != ":", ]
  temp[[cont_var]][temp[[cont_var]] == ">89"] <- "90"

  # convert age to numeric
  temp[[cont_var]] <- as.numeric(temp[[cont_var]])

  # filter out rows with missing values
  metadata_clean <- temp[ !(is.na(temp[[cont_var]]) | is.na(temp[[gene_name]])), ]
  metadata_clean <- metadata_clean[metadata_clean[[cat_var1]] != " unknown", ]


  #histogram for gene expression
  print(
    ggplot(metadata_clean, aes_string(x = gene_name)) +
      geom_histogram(bins = 50, fill = "pink", color = "black") +
      scale_x_continuous(breaks = seq(0, 190, by = 10)) +
      scale_y_continuous(breaks = seq(0, 10, by = 1)) +
      labs(title = paste("Histogram of", gene_name, "Expression"),
           x = paste(gene_name, "Expression Level"),
           y = "Frequency")
  )

  #scatterplot
  print(
    ggplot(metadata_clean, aes_string(x = cont_var, y = gene_name)) +
      scale_x_continuous(breaks = seq(10, 100, by = 10)) +
      scale_y_continuous(breaks = seq(0, 200, by = 10)) +
      geom_point(size = 2, colour = "pink", alpha = 0.8) +
```

```
      labs(title = paste(gene_name, "Expression vs. Age"),
           x = "Age (years)",
           y = paste(gene_name, "Expression Level"))
  )

  #boxplot
  print(
    ggplot(metadata_clean, aes_string(x = cat_var2, y = gene_name, fill = cat_var1)) +
      geom_boxplot() +
      scale_fill_manual(values = c(" female" = "pink", " male" = "deeppink")) +
      labs(title = paste(gene_name, "Expression by ICU Status and Sex" ),
           x = "ICU Status",
           y = paste(gene_name, "Expression Level"),
           fill = "Sex") +
      scale_y_continuous(breaks = seq(0, 200, by = 20))
  )
}
```

Generate summary statistics table

```
library(dplyr)
```

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

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

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

```
library(tableone)

#extract my genes row from the gene df and add to metadata
metadata$ABHD5 <- as.numeric(genes["ABHD5", ])
metadata$ventilator.free_days <- as.numeric(metadata$ventilator.free_days)
metadata$ferritin.ng.ml. <- as.numeric(metadata$ferritin.ng.ml.)
```

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

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

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

```
#clean
metadata_clean <- metadata %>%
  filter(!is.na(age) & !is.na(ferritin.ng.ml.) & !is.na(ventilator.free_days) &
         !is.na(sex) & !is.na(icu_status))
```

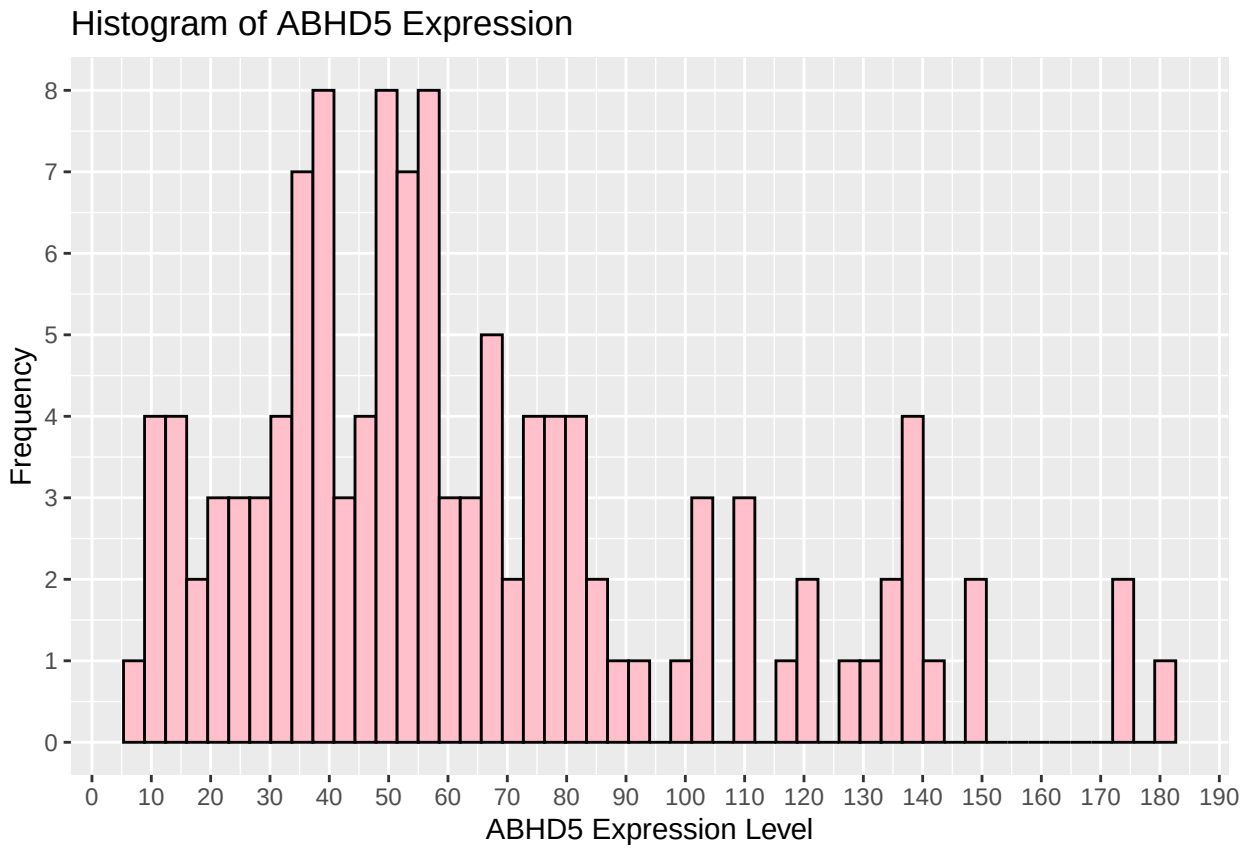
```
contVars <- c("age", "ventilator.free_days", "ferritin.ng.ml.")
catVars <- c("sex", "mechanical_ventilation", "icu_status")

vars <- c(contVars, catVars)

table1 <- CreateTableOne(
  vars = vars,
  strata = "icu_status",        # stratify by icu status
  data = metadata_clean,
  factorVars = catVars
)


print(table1,
      nonnormal = c("ferritin.ng.ml."),
      quote = FALSE,
      noSpaces = TRUE,
      test = TRUE)
```

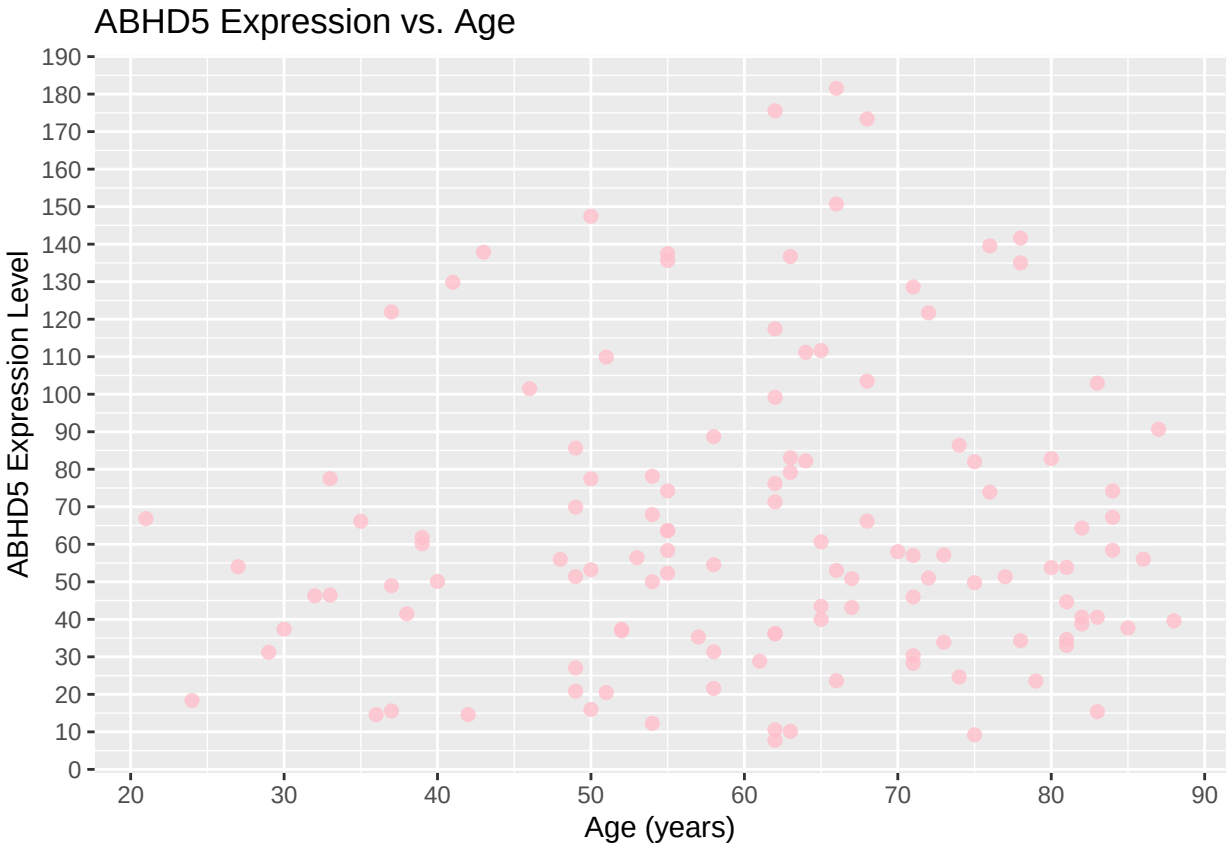
```
##                                    Stratified by icu_status
##                                     no
##   n                                 48
##   age (mean (SD))                   58.96 (18.00)
##   ventilator.free_days (mean (SD))  26.73 (5.68)
##   ferritin.ng.ml. (median [IQR])    406.00 [187.75, 905.75]
##   sex = male (%)                    24 (50.0)
##   mechanical_ventilation = yes (%) 3 (6.2)
##   icu_status = yes (%)              0 (0.0)
##                                    Stratified by icu_status
##                                     yes                     p      test
##   n                                 59
##   age (mean (SD))                   64.05 (13.38)           0.096
##   ventilator.free_days (mean (SD))  14.22 (11.82)           <0.001
##   ferritin.ng.ml. (median [IQR])    685.00 [325.00, 1212.00] 0.066  nonnorm
##   sex = male (%)                    37 (62.7)               0.261
##   mechanical_ventilation = yes (%) 43 (72.9)               <0.001
##   icu_status = yes (%)              59 (100.0)              <0.001
```

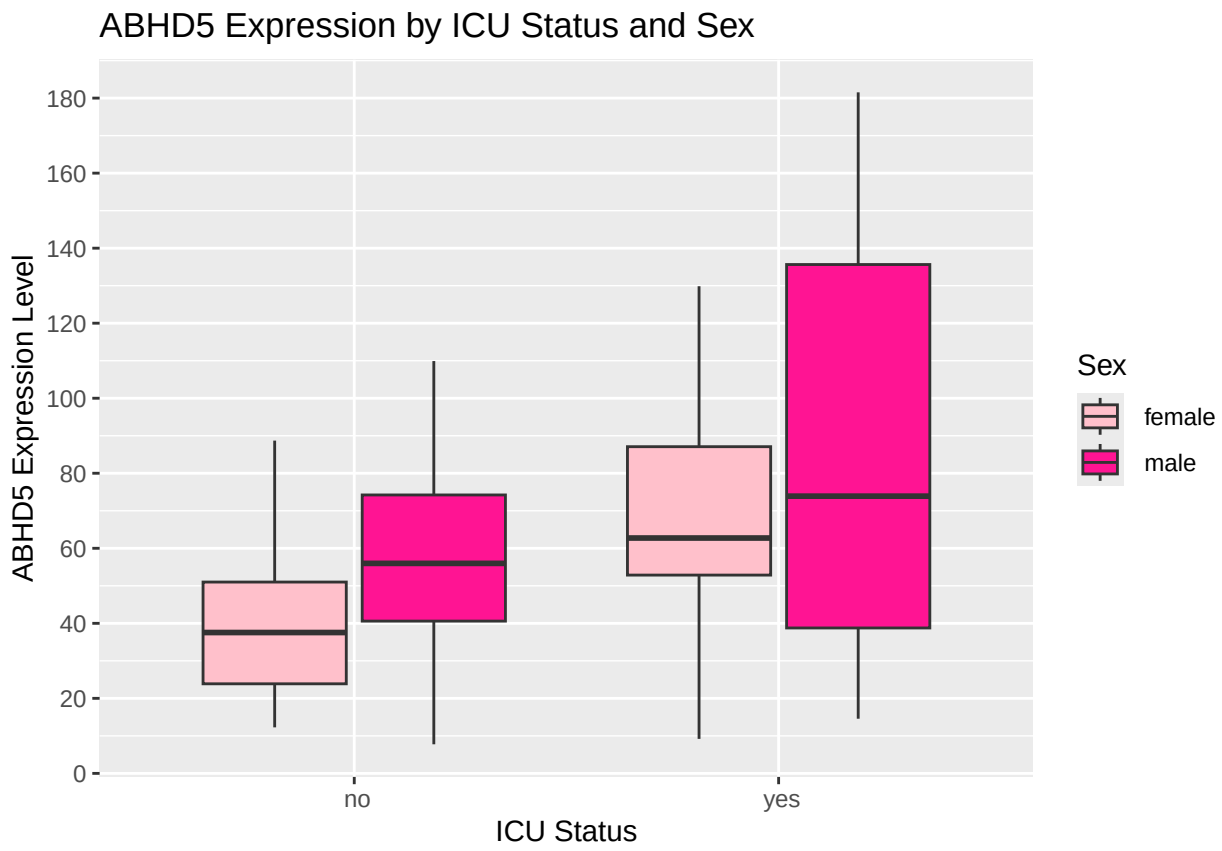
plot my gene from the first assignment: ABDH5

```
genes_to_plot <- c("ABHD5")
#plot
for (gene in genes_to_plot) {
  plot_all_figures(metadata, gene_name = gene, cont_var = "age", cat_var1 = "sex"
                   , cat_var2 = "icu_status")
}
```

```
## Warning: `aes_string()` was deprecated in ggplot2 3.0.0.
## i Please use tidy evaluation idioms with `aes()`.
## i See also `vignette("ggplot2-in-packages")` for more information.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```


Histogram of ABHD5 Expression
Frequency
ABHD5 Expression Level
0
1
2
3
4
5
6
7
8
0
10
20
30
40
50
60
70
80
90
100
110
120
130
140
150
160
170
180
190


ABHD5 Expression vs. Age
ABHD5 Expression Level
Age (years)
0
10
20
30
40
50
60
70
80
90
100
110
120
130
140
150
160
170
180
190
20
30
40
50
60
70
80
90



generating a heatmap

```r
library(pheatmap)

# 10 genes
gene_list <- c("ABHD17C", "ABHD18", "ABHD2" , "ABHD3", "ABHD4",
  "ABHD5", "ABHD6", "ABHD8", "ABI1", "ABI2")

# expression matrix
expr_mat <- as.matrix(apply(genes[gene_list, , drop = FALSE], 2, as.numeric))
#extract the 10 genes i picked
rownames(expr_mat) <- gene_list #labels with gene names

# tracking bars
bars <- data.frame(
  sex = factor(trimws(as.character(metadata$sex))),
  icu_status = factor(trimws(as.character(metadata$icu_status)))
)

#change names of labels
colnames(bars) <- c("ICU Status", "Sex")

#how to use matrix()
#https://www.datamentor.io/r-programming/matrix#google_vignette

#make tracking bars corresponding to icu status and sex
```

```r
rownames(bars) <- colnames(expr_mat)    # making sure things align

#how to use pheatmap
#https://biostatsquid.com/step-by-step-heatmap-tutorial-with-pheatmap/
# heatmap
pheatmap(expr_mat,#data from expression matrix
         scale = "row",
         main = paste( "Gene Expression Heatmap Stratified by
                              ICU Status and Sex" ),
         annotation_col = bars, #add tracking bars
         cluster_rows = TRUE,clustering_distance_rows = "euclidean", #cluster
         cluster_cols = TRUE, #cluster
         show_colnames = FALSE) #hide sample names
```

generating a novel plot

```r
library(ggplot2)
library(ggridges)
df <- data.frame( #make df
  expr = as.numeric(genes["ABHD5", ]), #get gene row
  icu_status = factor(trimws(metadata$icu_status), levels = #get icu status
                        c("no","yes")),
  sex = factor(trimws(metadata$sex), levels = c("female","male"))
  #get sex
)
df_clean <- subset(df, !is.na(expr) & sex != "unknown" & !is.na(icu_status))
```

```r
#plot
#ridgeline plot = stacked density plot
ggplot(df_clean, aes(x = expr, y = icu_status, fill = icu_status)) +
  geom_density_ridges(alpha = 0.7) +
  facet_wrap(~ sex) +
  scale_fill_manual(values = c("no" = "pink", "yes" = "hotpink"),
                    name = "Icu Status") +

  labs(
    x = "ABHD5 Expression",
    y = "ICU Status",
    title = "Ridgeline plot of ABHD5 expression by ICU status"
  ) +
  theme_minimal()
```

```
## Picking joint bandwidth of 11.1
```

```
## Picking joint bandwidth of 16.4
```


```r
#analysis
#In both sexes patients with ICU "yes" status show greater variability in
#ABHD5 expression compared to ICU "no."
```