Final Assignment

2025-08-11

Loading libraries

```r
library(ggplot2)
```

```r
# Loading gene expression data.

gene_expression <- read.csv2(paste0("C:/Users/Guest1/Dropbox/",
                                    "VuiFinal/QBS103_GSE157103_genes.csv"),sep = ",")

# Adding X as row.names of the dataframe.

rownames(gene_expression) <- gene_expression$X

# Removing first column.

gene_expression <- gene_expression[,-c(1)]

# Making gene expression values numeric.

gene_expression_num <- data.frame(apply(gene_expression,2,as.numeric))

rownames(gene_expression_num) <- rownames(gene_expression)

gene_expression <- gene_expression_num

dim(gene_expression)
```

```
## [1] 100 126
```

```r
# Transposing gene expression dataframe.

gene_expression_t <- as.data.frame(t(gene_expression))

# Assigning column and row names.

colnames(gene_expression_t) <- rownames(gene_expression)

rownames(gene_expression_t) <- colnames(gene_expression)

# Reading covariates data.

covariates_data <- read.csv2(paste0("C:/Users/Guest1/Dropbox/",

          "VuiFinal/QBS103_GSE157103_series_matrix-1.csv"),sep = ",")

# Selecting covariates
# Selecting three continuous and three categorical covariates.

continuous_cov <- c("hospital.free_days_post_45_day_followup","ferritin.ng.ml.",
                    "fibrinogen")

# Selecting three categorical covariates.

categorical_cov <- c("sex","mechanical_ventilation","disease_status")

# Subseting original covariates dataframe.

covariates_data_filt <- covariates_data[,c(continuous_cov,categorical_cov)]
```

Selecting genes.

To apply a criteria to select genes for this assignment, in this section I looked for the genes that presented the strongest associations with the selected categorical and continuous variables. For continuous variables perason´s correlations were computed and for categorigal t-tests were carried out.

```r
# Selecting genes with top accocititon to our covariates.

# Computing correlations with continuos covariates

# Getting the top associated genes for hospital free days post 45 days follow up

correlations <- c()

for (i  in 1:nrow(gene_expression)){

  cor_results <- cor.test(as.numeric(gene_expression[i,]),
                          as.numeric(covariates_data_filt[,1]), method = "pearson")

  correlations <- c(correlations,cor_results$estimate)
}


rownames(gene_expression)[which.max(correlations)]
```

```
## [1] "ABCD4"
```

```r
max(correlations,na.rm = TRUE)
```

```
## [1] 0.5248628
```

```r
rownames(gene_expression)[which.min(correlations)]
```

```
## [1] "ABCB6"
```

```r
min(correlations,na.rm = TRUE)
```

```
## [1] -0.3123125
```

```r
# ABCD4 top pos corr
# ABCB6 topneg corr

# Getting the top associated genes for ferritin
```

```
correlations <- c()

for (i  in 1:nrow(gene_expression)){

  cor_results <- cor.test(as.numeric(gene_expression[i,]),
                          as.numeric(covariates_data_filt[,2]), method = "pearson")

  correlations <- c(correlations,cor_results$estimate)
}


rownames(gene_expression)[which.max(correlations)]
```

```
## [1] "A4GALT"
```

```
max(correlations,na.rm = TRUE)
```

```
## [1] 0.2989352
```

```
rownames(gene_expression)[which.min(correlations)]
```

```
## [1] "AAMP"
```

```
min(correlations,na.rm = TRUE)
```

```
## [1] -0.2340216
```

```
# A4GALT top pos corr
# AAMP topneg corr

# Getting the top associated genes for fibrinogen

correlations <- c()

for (i  in 1:nrow(gene_expression)){

  cor_results <- cor.test(as.numeric(gene_expression[i,]),
                          as.numeric(covariates_data_filt[,3]), method = "pearson")

  correlations <- c(correlations,cor_results$estimate)
}


rownames(gene_expression)[which.max(correlations)]
```

```
## [1] "ABCA13"
```

```
max(correlations,na.rm = TRUE)
```

```
## [1] 0.3453429
```

```
rownames(gene_expression)[which.min(correlations)]
```

```
## [1] "ABCG2"
```

```
min(correlations,na.rm = TRUE)
```

```
## [1] -0.173614
```

```r
# ABCA13 top pos corr
# ABCG2 topneg corr


# Selecting top associated genes for categorical covariates.

# Removing white spaces in the variable values.

covariates_data_filt$sex <- trimws(covariates_data_filt$sex, which = "left")

# Replacing unknown by NA.

covariates_data_filt$sex[covariates_data_filt$sex == "unknown"] <- NA

table(covariates_data_filt$sex)
```

```
##
## female   male
##     51     74
```

```r
t_tests <- c()

for (i  in 1:nrow(gene_expression)){

  test_results <- t.test(as.numeric(gene_expression[i,]) ~ covariates_data_filt[,4])

  t_tests <- c(t_tests,test_results$statistic)
}


rownames(gene_expression)[which.max(t_tests)]
```

```
## [1] "ABHD16B"
```

```r
max(t_tests,na.rm = TRUE)
```

```
## [1] 2.585473
```

```r
rownames(gene_expression)[which.min(t_tests)]
```

```
## [1] "ABHD5"
```

```r
min(t_tests,na.rm = TRUE)
```

```
## [1] -2.425871
```

```r
# ABHD16B top pos corr
# ABHD5 topneg corr


# Selecting top associated genes for categorical covariates.

# Genes associated with mechanical ventilation.

table(covariates_data_filt$mechanical_ventilation)
```

```
##
##   no  yes
##   75   51
```

```r
t_tests <- c()

for (i  in 1:nrow(gene_expression)){

  test_results <- t.test(as.numeric(gene_expression[i,]) ~ covariates_data_filt[,5])

  t_tests <- c(t_tests,test_results$statistic)
}


rownames(gene_expression)[which.max(t_tests)]
```

```
## [1] "ABCD4"
```

```r
max(t_tests,na.rm = TRUE)
```

```
## [1] 8.703651
```

```r
rownames(gene_expression)[which.min(t_tests)]
```

```
## [1] "ABHD5"
```

```r
min(t_tests,na.rm = TRUE)
```

```
## [1] -5.494153
```

```r
# ABCD4 top pos t-test statistic
# ABHD5 top neg t-test statistic


# Selecting top associated genes for categorical covariates.

# Genes associated with disease status

table(covariates_data_filt$disease_status)
```

```
##
##     disease state: COVID-19 disease state: non-COVID-19
##                         100                          26
```

```r
t_tests <- c()

for (i  in 1:nrow(gene_expression)){

  test_results <- t.test(as.numeric(gene_expression[i,]) ~ covariates_data_filt[,6])

  t_tests <- c(t_tests,test_results$statistic)
}


rownames(gene_expression)[which.max(t_tests)]
```

```
## [1] "ABCB6"
```

```r
max(t_tests,na.rm = TRUE)
```

```
## [1] 9.148363
```

```r
rownames(gene_expression)[which.min(t_tests)]
```

```
## [1] "ABHD17A"
```

```r
min(t_tests,na.rm = TRUE)
```

```
## [1] -3.666531
```

```r
# ABCB6 top pos t-test statistic
# ABHD17A top neg t-test statistic
```

In this section I will format the column names of the filtered covariates dataframe.

```r
# Renaming columns of the covariate dataframe and claining data.

colnames(covariates_data_filt) <- c("HFD-45","Ferritin (ng/mL)",

          "Fibrinogen (mg/dL)","Sex","Mechanical Ventilation", "Disease Status")

# Replacing unknown values by NAs.

covariates_data_filt$`Fibrinogen (mg/dL)`[covariates_data_filt$`Fibrinogen (mg/dL)` == "unknown"] <- NA

covariates_data_filt$`Fibrinogen (mg/dL)` <- as.numeric(covariates_data_filt$`Fibrinogen (mg/dL)`)
```

```
## Warning: NAs introducidos por coerción
```

```r
covariates_data_filt$`Ferritin (ng/mL)`[covariates_data_filt$`Ferritin (ng/mL)` == "unknown"] <- NA

covariates_data_filt$`Ferritin (ng/mL)` <- as.numeric(covariates_data_filt$`Ferritin (ng/mL)`)
```

```
## Warning: NAs introducidos por coerción
```

I created a summary statistics table for our continuous and categorical data stratifying by Sex. To do that I first formated the covariates data. Then using the tableone palcage I created the tables that included means and standard deviations for continuous variables and percentages for categorical ones.

```r
library(tableone)

# Formatting the values.

covariates_data_filt$Sex[covariates_data_filt$Sex == "male"] <- "Male"

covariates_data_filt$Sex[covariates_data_filt$Sex == "female"] <- "Female"

covariates_data_filt$`Disease Status`[covariates_data_filt$`Disease Status` == "disease state: COVID-19

covariates_data_filt$`Disease Status`[covariates_data_filt$`Disease Status` == "disease state: non-COVI


# Using the package tableone to create the table.

tab <- CreateTableOne(vars = colnames(covariates_data_filt),
                      strata = "Sex",
                      data = covariates_data_filt,
                      factorVars = c("Sex","Mechanical Ventilation", "Disease Status"))
```

```
print(tab, showAllLevels = TRUE, quote = FALSE, noSpaces = TRUE, test = FALSE)
```

```
##                                 Stratified by Sex
##                                  level       Female           Male
##   n                                          51               74
##   HFD-45 (mean (SD))                         26.37 (16.34)    22.61 (17.02)
##   Ferritin (ng/mL) (mean (SD))               619.28 (1054.33) 993.35 (1013.05)
##   Fibrinogen (mg/dL) (mean (SD))             469.15 (163.26)  550.72 (219.68)
##   Sex (%)                        Female      51 (100.0)       0 (0.0)
##                                  Male        0 (0.0)          74 (100.0)
##   Mechanical Ventilation (%)     no          35 (68.6)        39 (52.7)
##                                  yes         16 (31.4)        35 (47.3)
##   Disease Status (%)             COVID-19    38 (74.5)        62 (83.8)
##                                  non-COVID-19 13 (25.5)        12 (16.2)
```

```
# Using the package xtable to create the summary statistics table in latex format.

library(xtable)

tab_latex <- print(tab, showAllLevels = TRUE, quote = FALSE, noSpaces = TRUE, test = FALSE, printToggle

print(xtable(tab_latex),type = "latex",)
```

```
## % latex table generated in R 4.5.1 by xtable 1.8-4 package
## % Wed Aug 20 04:13:34 2025
## \begin{table}[ht]
## \centering
## \begin{tabular}{rlll}
##   \hline
##  & level & Female & Male \\
##   \hline
## n &  & 51 & 74 \\
##   HFD.45..mean..SD.. &  & 26.37 (16.34) & 22.61 (17.02) \\
##   Ferritin..ng.mL...mean..SD.. &  & 619.28 (1054.33) & 993.35 (1013.05) \\
##   Fibrinogen..mg.dL...mean..SD.. &  & 469.15 (163.26) & 550.72 (219.68) \\
##   Sex.... & Female & 51 (100.0) & 0 (0.0) \\
##   X & Male & 0 (0.0) & 74 (100.0) \\
##   Mechanical.Ventilation.... & no & 35 (68.6) & 39 (52.7) \\
##   X.1 & yes & 16 (31.4) & 35 (47.3) \\
##   Disease.Status.... & COVID-19 & 38 (74.5) & 62 (83.8) \\
##   X.2 & non-COVID-19 & 13 (25.5) & 12 (16.2) \\
##    \hline
## \end{tabular}
## \end{table}
```

In this section I define the plotting functions

```
# Merging gene expression and covariate data.frames

selected_gene_tran <- data.frame(t(as.matrix(gene_expression)))

data <- data.frame(covariates_data_filt,selected_gene_tran,check.names = FALSE)
```

Generating plots for all selected covariates.

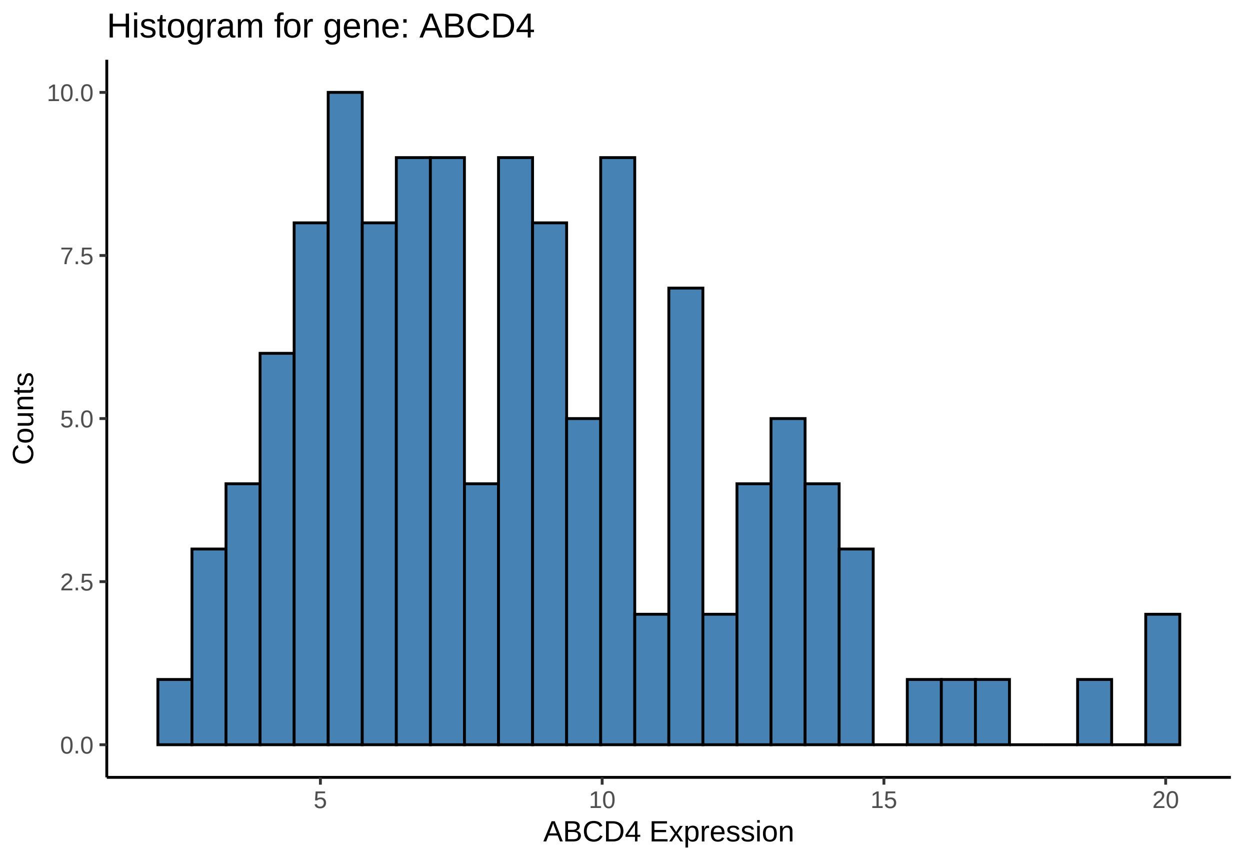
This section contains the code to generate the histograms, scatteplots and boxplots for all selected covariates and each selected gene.

```r
#  Dividing continous and categorical covariates into two data.frames

cont_covariate_list <- colnames(covariates_data_filt)[c(1,2,3)]

cat_covariate_list <-  colnames(covariates_data_filt)[c(4,5,6)]

# Writing function to create all plots.


function_vui <- function(d_f,list_of_genes,list_cont_cov,list_cat_cov){

  for (gene in list_of_genes){

    # Plotting histograms

    plot <- ggplot(d_f, aes(x = !!sym(gene))) + geom_histogram() +
    labs(title = paste("Histogram for gene: ", gene, sep ="")) + theme_minimal() +
    labs( y = "Counts")
    print(plot)

    # Plotting scatterplot.

    for (cont_cov in list_cont_cov){

      plot_2 <- ggplot(d_f, aes(y = !!sym(gene),
    x = !!sym(cont_cov))) +
    geom_point() +
    ggtitle(paste("Expression of ",
    gene," by Hospital Free Days", sep = "")) + theme_minimal()

    print(plot_2)

    }

    # Plotting boxplots.

    for (cat_cov in list_cat_cov){

        # Removing potential NA categories in the variable

        data_filt <- d_f[!is.na(d_f[,cat_cov]),]

        plot_3 <- ggplot(data_filt, aes(x = !!sym(cat_cov) ,
                                        y = !!sym(gene),fill = !!sym(cat_cov))) +
        geom_boxplot(na.rm = TRUE) + ggtitle(paste("Expression of ",
        gene," by ", cat_cov, sep = "")) + theme_minimal()

        print(plot_3)

      }
    }

}

library(ggplot2)

selected_genes <- c("ABCD4", "ABCB6", "A4GALT",
                    "AAMP", "ABCA13", "ABCG2", "ABHD16B", "ABHD5", "ABHD17A","A1BG")

# Generating plots for all the genes

function_vui(data,selected_genes,cont_covariate_list,cat_covariate_list)
```

Plots for the final assignment.

This section contains the function to generate the plots for the final submission of the final assignment.

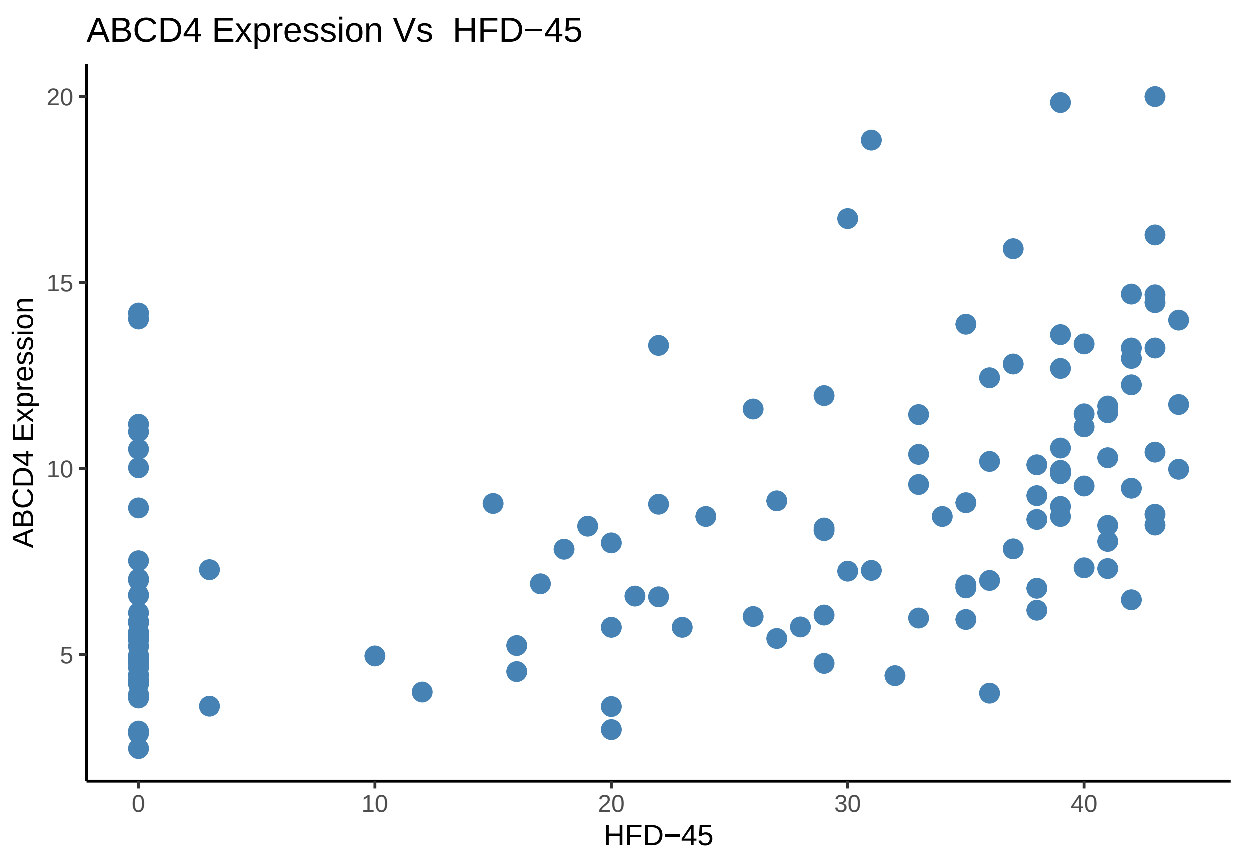
```r
# Creating function to plot histogram.

function_histogram_vui <- function(d_f,gene_name){

    plot <- ggplot(d_f, aes(x = !!sym(gene_name))) +

      geom_histogram(bins = 30, fill = "steelblue",

    color = "black") +

    labs(title = paste("Histogram for gene: ", gene_name, sep =""),

         y = "Counts", x = paste0(gene_name, " Expression") ) +

    theme_classic()

    print(plot)

    return(plot)
}

# Running the function

plot_hist <- function_histogram_vui(data,"ABCD4")
```



```
# Saving plot in png format using ggsave
ggsave("C:/Users/Guest1/Dropbox/VuiFinal/Plot_ABCD4_Hist.png", plot = plot_hist, width = 6, height = 4,
```

Here the function that creates the final scatter plot is created and executed.

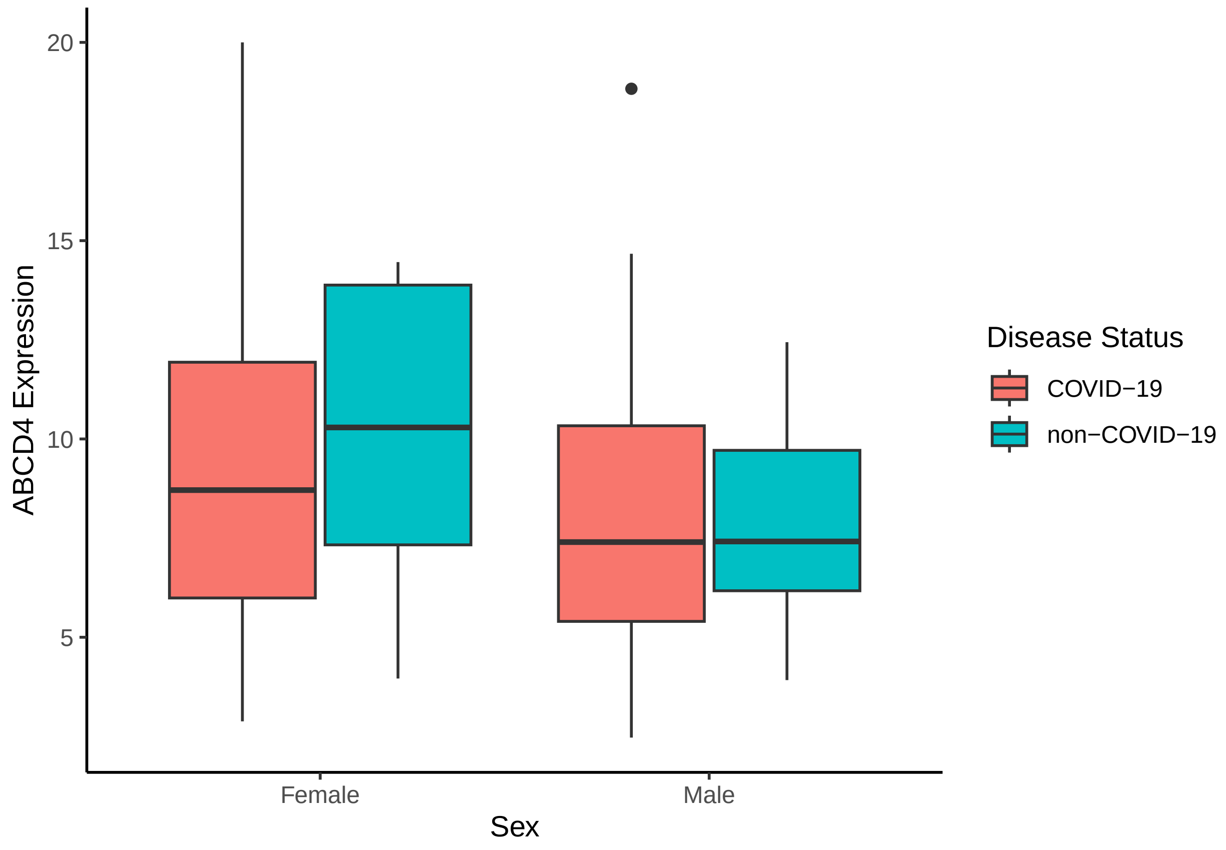
```
function_scatter_vui <- function(d_f, gene_name, cont_cov) {

  plot <- ggplot(d_f, aes(y = !!sym(gene_name),

                          x = !!sym(cont_cov))) +
    geom_point(color = "steelblue", size = 3) +

    labs(title = paste(gene_name, "Expression Vs ",

                       cont_cov),y = paste0(gene_name, " Expression")) +

    theme_classic()

  print(plot)

  return(plot)
}


# Running the function

dataplot_scat <- function_scatter_vui(data,"ABCD4","HFD-45")
```



```r
ggsave("C:/Users/Guest1/Dropbox/VuiFinal/Plot_ABCD4_Scatter.png",
       plot = dataplot_scat, width = 6, height = 4, dpi = 300)
```

Here the function that stratifies the boxplot by two different covariates is defined.

```r
function_box <- function(d_f,gene_name,cat_cov_1,cat_cov_2){

        # Filtering out data with missing values.

        data_filt <- d_f[!(is.na(d_f[,cat_cov_1]) | is.na(d_f[,cat_cov_2])),]

        plot <- ggplot(data_filt, aes(x = !!sym(cat_cov_1), y = !!sym(gene_name), fill = !!sym(cat_cov_

       geom_boxplot(position = position_dodge(width = 0.8)) +

       labs(x = cat_cov_1, y = paste0(gene_name, " Expression"), fill = cat_cov_2) +

       theme_classic()

       print(plot)

       return(plot)
}

dataplot_box <- function_box(data,"ABCD4","Sex","Disease Status")
```

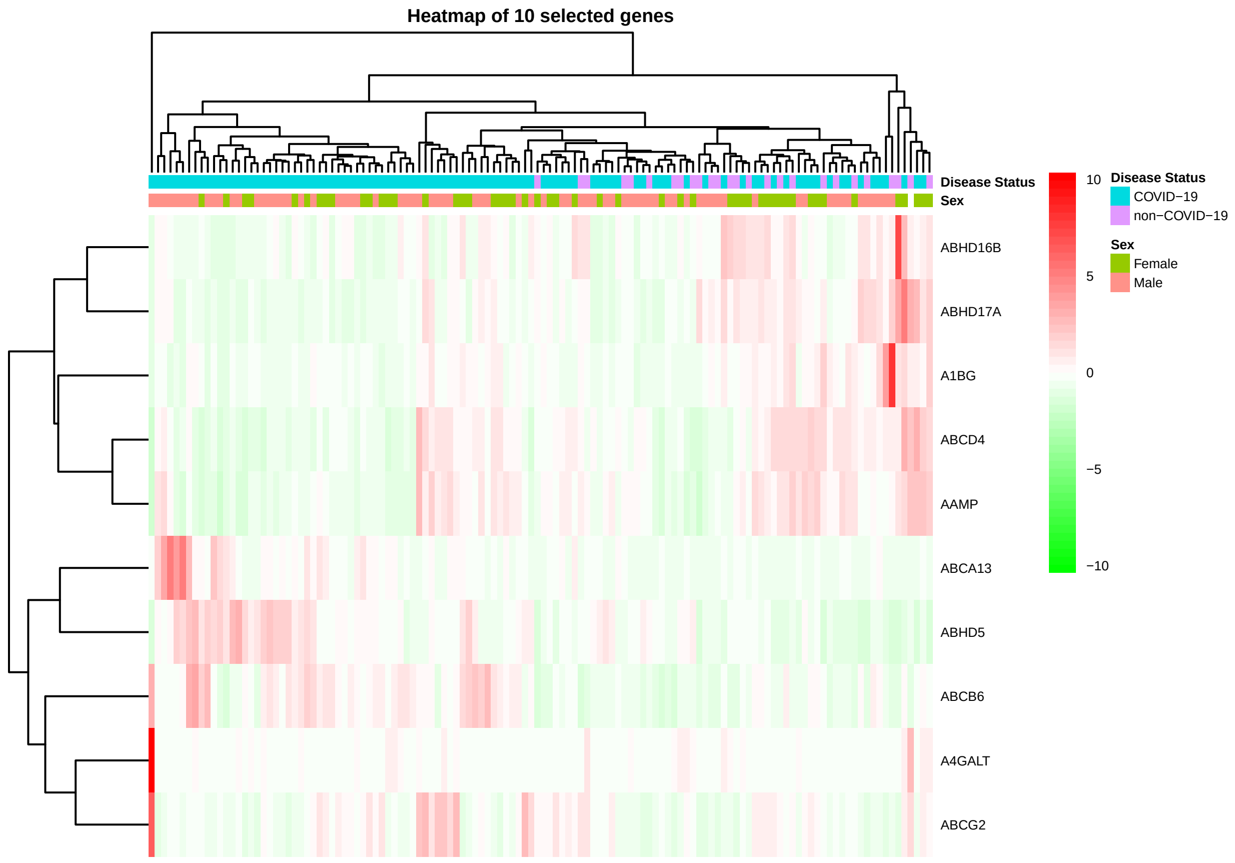


```
ggsave("C:/Users/Guest1/Dropbox/VuiFinal/Plot_ABCD4_Box.png",

       plot = dataplot_box, width = 6, height = 4, dpi = 300)
```

In this section we will create a heatmap with annotations. We selected the 9 genes that presented the top positive and negative associations with our selected covariates.

```
genes_selected_for_heatmap <- c("ABCD4", "ABCB6", "A4GALT", "AAMP", "ABCA13",
                                "ABCG2", "ABHD16B", "ABHD5", "ABHD17A","A1BG")


expression_selected <- data[,genes_selected_for_heatmap]

rownames(expression_selected) <- paste("Sample",seq(1,nrow(expression_selected)))

covariates_data <- data[,c(4,6)]

rownames(covariates_data) <- paste("Sample",seq(1,nrow(expression_selected)))

library(pheatmap)

# Sving Heatmap.

pheatmap(t(expression_selected),annotation_col = covariates_data, scale = "row",
         color = colorRampPalette(c("green", "white", "red"))(50),fontsize = 5, filename = "C:/Users/Gu
         show_colnames = FALSE, main = "Heatmap of 10 selected genes")
```

```r
# Plotting heatmap.

pheatmap(t(expression_selected),annotation_col = covariates_data, scale = "row",
         color = colorRampPalette(c("green", "white", "red"))(50),fontsize = 5,
         show_colnames = FALSE, main = "Heatmap of 10 selected genes")
```



Principal component analysis.

For the new plot we are going to generate a principal component analysis of the data.

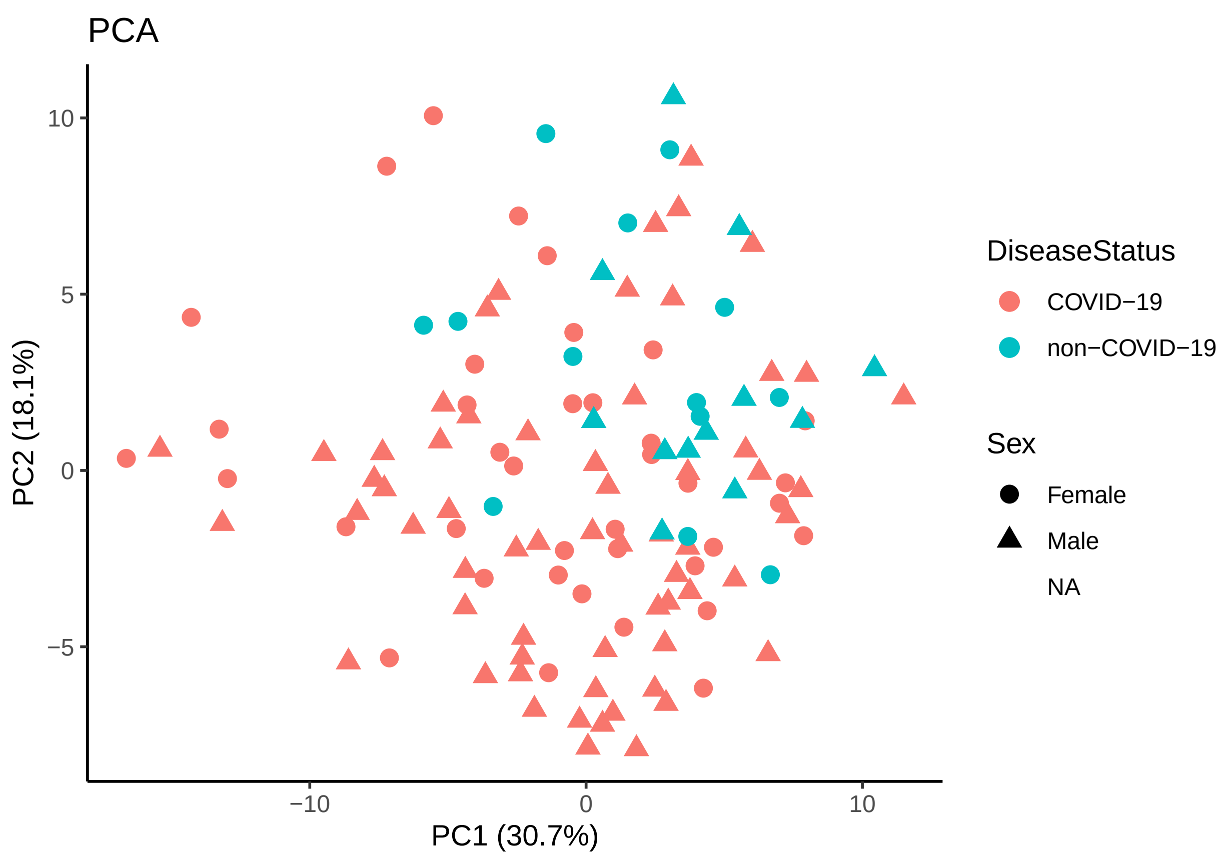
```r
# Getting gene expression.

gene_expression <- data[,seq(7,ncol(data))]

gene_expression <- gene_expression[, colSums(gene_expression != 0) > 0]

# Getting covariates.

covariates_data <- data[,seq(1,6)]

# Computing principal components of gene expression data using prcomp.

pca_result <- prcomp(gene_expression, center = TRUE, scale. = TRUE)

pca_scores <- as.data.frame(pca_result$x)
```

```
pca_scores$Sex <- covariates_data$Sex

pca_scores$DiseaseStatus <- covariates_data$`Disease Status`


library(ggplot2)

# Plotting PCA results using ggplot2.

pca_plot <- ggplot(pca_scores, aes(x = PC1, y = PC2,

                                  color = DiseaseStatus, shape = Sex )) +
  geom_point(size = 3) +

  theme_classic() +

  labs(title = "PCA",

   x = paste0("PC1 (", round(summary(pca_result)$importance[2,1]*100, 1), "%)"),

   y = paste0("PC2 (", round(summary(pca_result)$importance[2,2]*100, 1), "%)"))


ggsave("C:/Users/Guest1/Dropbox/VuiFinal/PlotPCA.png", plot = pca_plot, width = 6, height = 4, dpi = 30
```

```
## Warning: Removed 1 row containing missing values or values outside the scale range
## (`geom_point()`).
```

```
pca_plot
```

```
## Warning: Removed 1 row containing missing values or values outside the scale range
## (`geom_point()`).
```


PCA
PC2 (18.1%)
PC1 (30.7%)
10
5
0
−5
−10
0
10
DiseaseStatus
COVID−19
non−COVID−19
Sex
Female
Male
NA