

project1_final_checkpoint

2024-08-19

Import data

```
genes <- read.csv('/Users/isabellabaldacci/Desktop/QBS103/Project/QBS103_GSE157103_genes.csv',  
                 sep = ',', header = TRUE)  
metadata <- read.csv('/Users/isabellabaldacci/Desktop/QBS103/Project/QBS103_GSE157103_series_matrix.csv',  
                    header = TRUE, stringsAsFactors = TRUE)
```

```
#'procalcitonin.ng.ml..' other continuous  
# 'ferritin.ng.ml' other continuous  
# sex other categorical
```

Clean up metadata

```
## cleaning up metadata  
# metadata were removed if they were not helpful or if they had too many unknowns  
metadata_clean <- dplyr::select(metadata,  
                                -c("X.Sample_submission_date", "channel_count",  
                                   "status", "last_update_date", "type",  
                                   "channel_count", "source_name_ch1",  
                                   "organism_ch1", "apacheii",  
                                   "ddimer.mg.l_feu.",  
                                   "lactate.mmol.l.", "fibrinogen", "sofa"))
```

Transpose Genes dataframe

```
#transposing the genes x samples matrix to samples x genes  
rownames(genes) <- genes$X #assigning row names  
genes_t <- as.data.frame(t(dplyr::select(genes, -c('X')))) #transpose as a dataframe  
genes_t$participant_id <- rownames(genes_t) #assigning participant id column for later joining  
#which(genes_t$participant_id == "COVID_06_.y_male_NonICU") #finding column with mismatch to metadata  
genes_t$participant_id[6] <- "COVID_06_.y_male_NonICU" #reassigning value to match metadata
```

```
#combining the samples x genes matrix to the metadata  
all_data <- dplyr::inner_join(genes_t, metadata_clean, by = 'participant_id')  
  
#finding all values that would cause issues later  
findNAs <- function(value) {
```

```

new_val <- ifelse(((value == 'unknown') |
                  (value == ' unknown') |
                  (value == ' :') |
                  (value == " >89")),
                 NA,
                 value)

return(new_val)
}

clean_data <- as.data.frame(apply(all_data, 2, FUN = findNAs))
head(clean_data)

```

```

##      A1BG A1CF  A2M A2ML1 A3GALT2 A4GALT A4GNT  AAAS AACS AADAC AADACL2 AADACL3
## 1 0.49 0.00 0.21  0.04      0.07  0.00  0.03 18.92 4.07      0    0.00    0.00
## 2 0.29 0.00 0.14  0.00      0.00  0.00  0.05 18.68 3.00      0    0.00    0.06
## 3 0.26 0.00 0.03  0.02      0.00  0.00  0.07 13.85 1.83      0    0.00    0.00
## 4 0.45 0.01 0.09  0.07      0.00  0.00  0.00 22.11 4.22      0    0.00    0.00
## 5 0.17 0.00 0.00  0.05      0.07  0.00  0.00  8.45 1.17      0    0.00    0.00
## 6 0.21 0.00 0.08  0.04      0.00  0.00  0.03 19.60 3.15      0    0.01    0.00
##      AADACL4 AADAT AAGAB  AAK1 AAMDC  AAMP AANAT  AAR2 AARD AARS1 AARS2 AARSD1
## 1      0.00  0.00 22.93  7.12 17.19 61.08  0.31 21.59 0.18 13.52  2.43  9.63
## 2      0.00  0.00 21.69  6.46 13.06 54.54  0.00 19.22 0.08 13.91  2.15 11.80
## 3      0.00  0.00 18.27  3.92 13.35 25.19  0.65  8.72 0.05  5.60  1.08  4.04
## 4      0.00  0.00 26.69  8.74 17.53 67.95  0.08 20.83 0.03 15.34  2.69 14.61
## 5      0.00  0.03 17.02  7.26 10.14 18.29  1.02  7.46 0.10  6.29  0.55  5.32
## 6      0.00  0.00 17.50  6.65 12.98 45.09  0.12 13.60 0.09 13.70  2.21 11.30
##      AASDH AASDHPPT AASS  AATF  AATK  ABAT ABCA1 ABCA10 ABCA12 ABCA13 ABCA2 ABCA3
## 1  6.38      19.45 0.21 45.83  5.60  9.63 32.30  0.32  0.00  0.49  8.47  0.37
## 2  4.83      12.14 0.42 39.37  9.79 10.36 15.84  0.37  0.00  3.36  9.49  0.71
## 3  2.76       7.64 0.04 42.35  5.34  4.59 34.38  0.29  0.00  0.26 14.24  0.17
## 4  8.23      18.94 0.41 41.92  4.73 11.02 14.24  0.31  0.00  0.13  6.37  0.94
## 5  5.23      11.54 0.21 30.56  1.50  4.42 18.39  0.19  0.00  0.16  5.90  0.17
## 6  4.91      14.16 0.26 36.30  7.52  7.04  3.64  0.22  0.00  0.97  6.18  0.43
##      ABCA4 ABCA5 ABCA6 ABCA7 ABCA8 ABCA9 ABCB1 ABCB10 ABCB11 ABCB4 ABCB5 ABCB6
## 1  0.01  1.86  0.19 39.31  0.00  0.27  1.61 15.59  0.38  0.01  0.04  2.77
## 2  0.00  2.81  0.11 30.42  0.00  0.20  1.68 10.98  0.25  0.19  0.04  2.52
## 3  0.00  2.17  0.07 54.85  0.00  0.33  0.59  4.04  0.14  0.09  0.15  3.14
## 4  0.00  2.94  0.02 18.91  0.01  0.30  3.14 10.00  0.18  0.74  0.11  2.11
## 5  0.00  1.38  0.03 23.28  0.00  0.21  1.66  5.69  0.07  0.21  0.04  1.14
## 6  0.00  1.89  0.03 23.43  0.00  0.15  1.92  9.33  0.12  0.26  0.06  2.08
##      ABCB7 ABCB8 ABCB9 ABCC1 ABCC10 ABCC11 ABCC12 ABCC2 ABCC3 ABCC4 ABCC5 ABCC6
## 1  6.42  2.95  0.20 11.20  8.85  0.04  0.00  1.65  7.19  5.96 14.83  6.62
## 2  5.80  2.35  0.51  9.39  4.46  0.01  0.00  1.31 16.21  4.49 22.21  5.87
## 3  2.59  1.57  0.05  4.74  4.08  0.09  0.00  1.72  1.44  0.58 13.02  3.13
## 4  7.42  2.71  0.31 13.05  5.75  0.02  0.01  1.55 11.11  4.09 17.49  3.37
## 5  3.40  0.71  0.12  4.29  1.91  0.02  0.00  2.56  3.13  7.37  5.53  1.66
## 6  4.87  2.41  0.75  8.45  3.88  0.00  0.00  0.85  4.94  4.12 14.04  3.96
##      ABCC8 ABCC9 ABCD1 ABCD2 ABCD3 ABCD4 ABCE1 ABCF1 ABCF2 ABCF2-H2BE1 ABCF3 ABCG1
## 1  0.00  1.53 11.26  1.55 10.62 11.19 14.09 24.40  8.77      11.67 23.37 32.76
## 2  0.00  1.89  4.65  1.08 11.23  9.86 15.52 16.84  7.21      8.84 18.63 23.76
## 3  0.00  1.97  5.83  0.39  3.99  7.83  4.50  7.89  3.12      3.06 11.57 41.14
## 4  0.00  2.82  4.80  1.74 12.38 13.60 21.20 23.92  8.26      12.51 23.62 15.72
## 5  0.00  1.84  1.93  0.59  8.19  5.43  7.16 12.33  2.76      4.72  9.86  9.43

```

```

## 6  0.00  1.37  2.56  2.64  8.52 10.19 14.20 19.56  5.57          6.47 16.81 13.84
##   ABCG2 ABCG4 ABCG5 ABCG8 ABHD1 ABHD10 ABHD11 ABHD12 ABHD12B ABHD13 ABHD14A
## 1  0.17  0.01  0.00  0.02  0.04 14.74  8.50 14.66  0.55 15.58  6.89
## 2  0.00  0.07  0.00  0.01  0.14 10.57  8.85  9.02  1.18 14.33  7.08
## 3  0.15  0.00  0.00  0.00  0.00  6.84  8.22  4.84  0.88  9.97  2.12
## 4  0.14  0.00  0.00  0.04  0.31 16.73  9.64 13.22  0.68 16.70  9.21
## 5  0.03  0.00  0.00  0.04  0.04  6.93  4.02  3.36  0.68 21.02  1.95
## 6  0.13  0.00  0.00  0.00  0.21 10.02  6.51  9.33  0.48 11.72  8.84
##   ABHD14A-ACY1 ABHD14B ABHD15 ABHD16A ABHD16B ABHD17A ABHD17B ABHD17C ABHD18
## 1          0.00 18.53  6.82 77.48  0.29 13.79  8.78  1.37  6.93
## 2          0.00 18.78  5.76 57.11  0.19 11.89  7.38  3.34  5.10
## 3          0.00  7.63  2.41 77.73  0.08  6.88  5.59  1.64  4.06
## 4          0.00 28.23  8.12 56.77  0.05 12.28  9.89  1.24  6.57
## 5          0.00  6.49  2.26 63.66  0.04  4.11  6.64  1.17  7.94
## 6          0.00 23.99  4.99 44.47  0.12 11.68  7.10  2.52  3.93
##   ABHD2  ABHD3  ABHD4  ABHD5  ABHD6  ABHD8  ABI1  ABI2          participant_id
## 1 46.50  73.51 44.47  60.14  3.47  3.27 75.36  3.22 COVID_01_39y_male_NonICU
## 2 63.70  80.70 44.43  79.17  4.54  0.99 60.41  3.68 COVID_02_63y_male_NonICU
## 3 36.05 113.96 37.71  77.50  2.15  2.59 61.63  0.67 COVID_03_33y_male_NonICU
## 4 54.52 106.44 41.75  69.89  7.84  1.86 66.52  4.99 COVID_04_49y_male_NonICU
## 5 72.44 190.95 29.65  51.39  1.45  1.11 79.84  2.12 COVID_05_49y_male_NonICU
## 6 45.66  83.74 34.00  60.38  4.30  1.27 54.23  3.47 COVID_06_:y_male_NonICU
##   geo_accession      disease_status  age  sex  icu_status charlson_score
## 1   GSM4753021 disease state: COVID-19  39 male          no              0
## 2   GSM4753022 disease state: COVID-19  63 male          no              2
## 3   GSM4753023 disease state: COVID-19  33 male          no              2
## 4   GSM4753024 disease state: COVID-19  49 male          no              1
## 5   GSM4753025 disease state: COVID-19  49 male          no              1
## 6   GSM4753026 disease state: COVID-19 <NA> male          no              1
##   mechanical_ventilation ventilator.free_days
## 1                      yes                   0
## 2                      no                    28
## 3                      no                    28
## 4                      no                    28
## 5                      yes                    23
## 6                      no                    28
##   hospital.free_days_post_45_day_followup ferritin.ng.ml. crp.mg.l.
## 1                      0                946        73.1
## 2                      39               1060        <NA>
## 3                      18               1335        53.2
## 4                      39                583       251.1
## 5                      27                800       355.8
## 6                      36                563       129.1
##   procalcitonin.ng.ml..
## 1                      36
## 2                      0.37
## 3                      0.07
## 4                      0.98
## 5                      4.92
## 6                      0.67

```

```

#taken from class notes
# Define a function to calculate a mean or a median
contSummary <- function(x,normal = T) {

```

```

#x <- as.numeric(x, na.rm = TRUE)
# Calculate mean (sd) if normally distributed (the default)
if (normal == T) {
  # Calculate individual values
  myMean <- round(mean(x),2)
  mySD <- round(sd(x),2)
  # Combine values
  paste0(myMean, ' (' ,mySD, ')')
}
# Calculate median (IQR) if non-normally distributed
else {
  # Calculate individual values
  myMedian <- round(median(x, na.rm = TRUE), 2)
  myIQR <- round(IQR(x), 2)
  #myIQR1 <- round(quantile(x,1/4),digits = 2)
  #myIQR2 <- round(quantile(x,3/4),digits = 2)
  # Combine values
  paste0(myMedian, ' [' ,myIQR, ']')
}
}

```

Summary Statistics

```

#omitted NAs, which made the dim 97 instead of 126
table_data <- na.omit(clean_data %>%
  dplyr::select('mechanical_ventilation',
                'disease_status',
                'sex',
                'crp.mg.l.',
                'ferritin.ng.ml.',
                'procalcitonin.ng.ml..'))

table_data$disease_status <-
  factor(table_data$disease_status,
         levels = unique(table_data$disease_status))

table_data$mechanical_ventilation <-
  factor(table_data$mechanical_ventilation,
         levels = unique(table_data$mechanical_ventilation),
         labels = c('Mechanical Ventilator',
                    'No Mechanical Ventilator'))

table_data$sex <- factor(table_data$sex,
                        levels = unique(table_data$sex),
                        labels = c('Male', 'Female'))

table_data$crp.mg.l. <- as.numeric(table_data$crp.mg.l.)
table_data$ferritin.ng.ml. <- as.numeric(table_data$ferritin.ng.ml.)
table_data$procalcitonin.ng.ml.. <-
  as.numeric(table_data$procalcitonin.ng.ml..)

```

```

sn <- as.data.frame(table(table_data$sex,
                           table_data$disease_status))
sp <- as.data.frame(prop.table(table(table_data$sex,
                                     table_data$disease_status))) %>%
  dplyr::rename('Percent' = 'Freq')

sex <- dplyr::inner_join(sn, sp, by = join_by(Var1, Var2))

mn <- as.data.frame(table(table_data$mechanical_ventilation,
                           table_data$disease_status))
mp <- as.data.frame(prop.table(table(table_data$mechanical_ventilation,
                                     table_data$disease_status))) %>%
  dplyr::rename('Percent' = 'Freq')

mech <- dplyr::inner_join(mn, mp, by = join_by(Var1, Var2))

dsn <- as.data.frame(table(table_data$disease_status))
dsp <- as.data.frame(prop.table(table(table_data$disease_status))) %>%
  dplyr::rename('Percent' = 'Freq')

ds <- dplyr::inner_join(dsn, dsp, by = join_by(Var1))
ds$Var2 <- ds$Var1
ds$Var1 <- c('Disease State', 'Disease State')
ds <- ds[,c(1,4,2,3)]

all_cat <- dplyr::bind_rows(sex, mech) %>%
  dplyr::bind_rows(ds) %>%
  dplyr::mutate(as_str = paste0(Freq, ' (',
                                round(Percent*100,1), '%)')) %>%
  dplyr::select(Var1, Var2, as_str) %>%
  tidyr::pivot_wider(names_from = Var2, values_from = as_str)

all_cat_final <- all_cat %>%
  dplyr::mutate(Variables = paste0(Var1, '')) %>%
  dplyr::rename('Covid.Positive' = 'disease state: COVID-19',
                'Covid.Negative' = 'disease state: non-COVID-19') %>%
  dplyr::select(Variables, Covid.Positive, Covid.Negative)

cont_vars_pos <- table_data %>%
  dplyr::filter(disease_status == 'disease state: COVID-19') %>%
  dplyr::select('crp.mg.l.', 'ferritin.ng.ml.', 'procalcitonin.ng.ml.') %>%
  apply(MARGIN = 2, FUN = function(x) {contSummary(x, normal = F)})

cont_vars_neg <- table_data %>%
  dplyr::filter(disease_status == 'disease state: non-COVID-19') %>%
  dplyr::select('crp.mg.l.', 'ferritin.ng.ml.', 'procalcitonin.ng.ml.') %>%
  apply(MARGIN = 2, FUN = function(x) {contSummary(x, normal = F)})

pos <- as.data.frame(cont_vars_pos)
neg <- as.data.frame(cont_vars_neg)

```

```
continuous_variables_table <- merge(cont_vars_pos,
                                   cont_vars_neg,
                                   by= 'row.names')

cvt <- continuous_variables_table %>%
  dplyr::rename(Variables = Row.names,
               Covid.Positive = x, Covid.Negative = y)
cvt$Variables <- c('CRP (mg/L) Median [IQR]',
                  'Ferritin (ng/mL) Median[IQR]',
                  'Procalitonin (ng/mL) Median[IQR]')
```

```
# Define Table
```

```
# Print table using kable
```

```
tab <- kable(x = cvt, caption = 'Summary Table',
            format = 'latex',booktabs = T,
            col.names = c('Variable', 'Covid Positive', 'Covid Negative'),
            align = c('l', 'r'),escape = T) %>%
  #add_indent(positions = c(3,3),level_of_indent = 1)
```

```
tableLatex <- dplyr::bind_rows(cvt, all_cat_final)
#head(tableLatex)
tableLatex <- tableLatex %>% dplyr::add_row(Variables = 'Sex N(%)',
                                           Covid.Positive = ' ',
                                           Covid.Negative = ' ',
                                           .before = 4) %>%
  dplyr::add_row(Variables = 'Mechanical Ventilation N(%)',
                 Covid.Positive = ' ',
                 Covid.Negative = ' ',
                 .before = 7)
tableLatex <- tableLatex[c(10, 4,5,6,7,8,9, 1,2,3),]
tableLatex
```

```
##              Variables Covid.Positive Covid.Negative
## 10          Disease State      83 (85.6)      14 (14.4)
## 4              Sex N(%)
## 5              Male          53 (54.6)        5 (5.2)
## 6              Female        30 (30.9)        9 (9.3)
## 7  Mechanical Ventilation N(%)
## 8          Mechanical Ventilator      39 (40.2)        6 (6.2)
## 9          No Mechanical Ventilator      44 (45.4)        8 (8.2)
## 1         CRP (mg/L) Median [IQR] 129.6 [164.15]  44.65 [85.23]
## 2         Ferritin (ng/mL) Median[IQR]  686 [858.5]   142 [214.5]
## 3  Procalitonin (ng/mL) Median[IQR]   0.57 [1.6]    0.46 [0.39]
```

```
tab2 <- kable(x = tableLatex, caption = 'Summary Table',
             format = 'latex',booktabs = T,
             row.names = FALSE,
             col.names = c('Variable', 'Covid Positive', 'Covid Negative'),
             align = c('l', 'r'),escape = T) %>%
  add_indent(c(3,4,6,7), all_cols = FALSE, target_cols = 1 )
```

```

plots <- function(df, genes_list, cont_cov, cat_cov) {
  #filter dataframe (received error message indicating to use "all of")
  df_filtered <- df %>% dplyr::select(participant_id,
                                     all_of(genes_list),
                                     all_of(cont_cov),
                                     all_of(cat_cov))

  #cast to long
  df_filtered_long <- df_filtered %>% tidyr::pivot_longer(cols = genes_list,
                                                         names_to = 'Gene',
                                                         values_to = 'Expression')

  for (gene in genes_list){ #for gene in genes list

    one_gene <- df_filtered_long %>% dplyr::filter(Gene == gene) #get dataframe for one gene

    #get data for plotting, need mean, sd, and the 75th quantile for positioning annotation box
    mean_expression <- round(mean(one_gene$Expression), 2)
    sd_expression <- round(sd(one_gene$Expression), 2)
    x_min_box <- quantile(one_gene$Expression, 3/4)
    n <- length(one_gene$Expression)

    #create labels
    hist_title <- paste('Distribution of ', gene, ' expression across samples')
    hist_x <- paste(gene, ' Expression')

    #plot histogram using theme parameters from previous assignment
    hist <- one_gene %>% ggplot(aes(Expression)) +
      geom_histogram(bins = 30, fill = 'slateblue1') +
      theme_minimal() +
      theme(plot.title = element_text(hjust = .4)) +
      labs(title = hist_title,
           x = hist_x,
           y = 'Frequency') +
      scale_color_manual(values = c('slateblue1')) +
      annotate('rect',
              xmin = x_min_box + 5,
              xmax = x_min_box + 25,
              ymin = 10,
              ymax = 14, fill = 'grey', alpha = .9)+
      annotate(geom = 'text', x = x_min_box + 15, y = 12, label = paste('N:', n, ' samples',
                                                                    '\n Mean Expression:', mean_expression,
                                                                    '\n sd:', sd_expression), size = 2.5)

    #print to display the histogram
    print(hist)

    #filter the data to create the scatter plot
    #to avoid warnings of coerced NAs and to ensure cont_cov is numeric
    scatter_filtered <- one_gene %>% filter(one_gene[[cont_cov]] != ' unknown')
    scatter_filtered[cont_cov] <- as.numeric(scatter_filtered[[cont_cov]])
    #assign the levels for crp
    if (cont_cov == 'crp.mg.l.') {

```

```

scatter_filtered$crp.level <- cut(scatter_filtered$crp.mg.l.,
                                breaks = c(0, 3, 10, 100, 500000),
                                labels = c('normal', 'moderate', 'high', 'severe'))
}

#set color palette and titles
colorPalette_scatter <- c('azure4', 'skyblue2', 'slateblue2', 'midnightblue')
title_scatter <- paste(toupper(cont_cov), ' vs ', gene, ' Expression')
x_scatter <- paste(toupper(cont_cov), '(mg/L)')
y_scatter <- paste(gene, ' Expression')

#create scatterplot with parameters from previous assignment
scatter <- scatter_filtered %>% ggplot(aes(x = .data[[cont_cov]],
                                           y = Expression,
                                           color = crp.level)) +

  geom_point(na.rm = TRUE) +
  theme(
    axis.text.x = element_text(angle=90),
    plot.title = element_text(hjust = .4),
    legend.position = c(.85,.75)) +
  labs(title = title_scatter, x = x_scatter, y = y_scatter) +
  scale_color_manual(labels =
    c('Normal (<3mg/L)', 'Moderate (3-10mg/L)',
      'High (10 -100mg/L)', 'Severe (>100mg/L)'),
    values = colorPalette_scatter)

#print scatterplot
print(scatter)

#create titles for box plot
title_box <- paste(gene, ' Expression across ', cat_cov[1], ' and ', cat_cov[2])
x_lab_box <- cat_cov[1]
y_lab_box <- paste(gene, ' Expression')
#get num_covid positive and negative for annotations
n_covid_pos <- length(dplyr::filter(
  one_gene, disease_status == 'disease state: COVID-19')$Expression)
n_covid_neg <- length(dplyr::filter(
  one_gene, disease_status != 'disease state: COVID-19')$Expression)
#set the y value for the annotations as the max of expression so that it will not cover the data
y_annot <- max(one_gene$Expression)
labels_legend <- c(paste('no ', cat_cov[2]), cat_cov[2])

#generate box plot using same parameters as previous assignment
box_filtered <- one_gene %>% dplyr::filter(one_gene[[cat_cov[2]]] != ' unknown')
box <- box_filtered %>% ggplot(aes_string(x = cat_cov[1], y = 'Expression', fill = cat_cov[2])) +
  geom_boxplot() +
  theme_minimal() +
  theme(legend.position = 'bottom') +
  labs(title = title_box,
       x = x_lab_box,
       y = y_lab_box) +
  scale_fill_manual(labels = labels_legend,

```



```

        values = c('ghostwhite', 'mediumvioletred')) +
scale_x_discrete(labels = c('COVID', 'NON-COVID')) +
annotate(geom = 'text', x = 1, y = y_annot + 5,
        label = paste('Covid positive: N = ', n_covid_pos),
        color = 'black')+
annotate(geom = 'text', x = 2, y = y_annot + 5,
        label = paste('Covid negative: N = ', n_covid_neg),
        color = 'black')
print(box)

#just an idea I had for putting it all together
#all_plots <- ggpubr::ggarrange(hist, ggarrange(scatter, box, ncol = 2,
#labels = c('B', 'C')), nrow = 2, labels = c('A'))
#plotlist = list(box, scatter, hist), ncol = 3, nrow = 1, widths = c(5,5,5))
#print(all_plots)
}
}

```

```

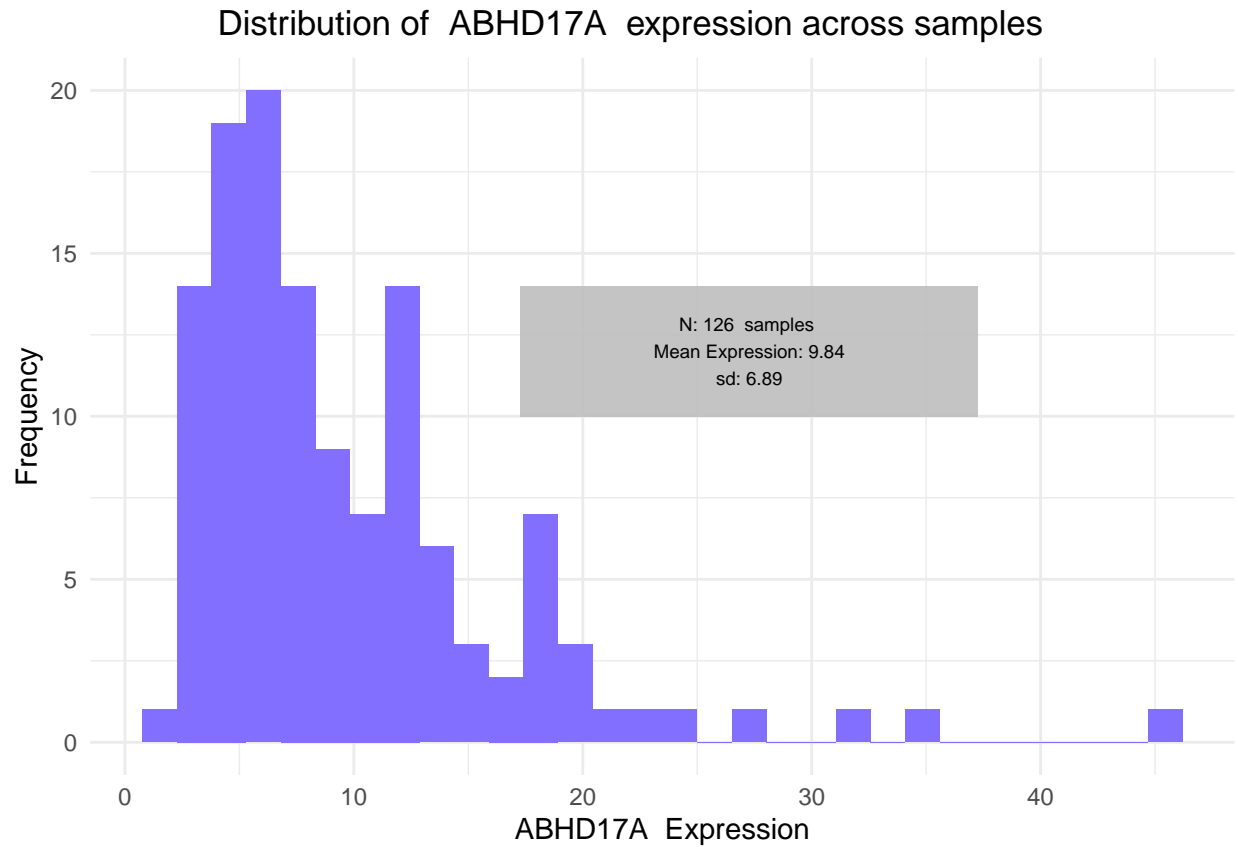
#a few warnings about deprecated methods, did not cause
#problems so decided to suppress for final output
plots(all_data, c('ABHD17A'),
      c('crp.mg.l.'),
      c('disease_status', 'mechanical_ventilation'))

```

```

## Warning: Using an external vector in selections was deprecated in tidysselect 1.1.0.
## i Please use 'all_of()' or 'any_of()' instead.
##   # Was:
##   data %>% select(genes_list)
##
##   # Now:
##   data %>% select(all_of(genes_list))
##
## See <https://tidysselect.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.

```



```
## 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.
```

