main

2024-07-20

```r
library(tidyverse)
```

```
## Warning: package 'tidyverse' was built under R version 4.3.3

## Warning: package 'ggplot2' was built under R version 4.3.3

## Warning: package 'tibble' was built under R version 4.3.3

## Warning: package 'tidyr' was built under R version 4.3.3

## Warning: package 'readr' was built under R version 4.3.3

## Warning: package 'purrr' was built under R version 4.3.3

## Warning: package 'dplyr' was built under R version 4.3.3

## Warning: package 'stringr' was built under R version 4.3.3

## Warning: package 'forcats' was built under R version 4.3.3

## Warning: package 'lubridate' was built under R version 4.3.3

## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.5
## v forcats   1.0.0     v stringr   1.5.1
## v ggplot2   3.5.1     v tibble    3.2.1
## v lubridate 1.9.3     v tidyr     1.3.1
## v purrr     1.0.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become error
```

```r
library(rlang)
```

```
## Warning: package 'rlang' was built under R version 4.3.3

##
## Attaching package: 'rlang'
##
## The following objects are masked from 'package:purrr':
##
##     %@%, flatten, flatten_chr, flatten_dbl, flatten_int, flatten_lgl,
##     flatten_raw, invoke, splice
```

```r
library(ggpubr)
```

```
## Warning: package 'ggpubr' was built under R version 4.3.3
```

```r
# import the csv

series_matrix<- read.csv('./data/QBS103_GSE157103_series_matrix.csv')
genes <- read.csv('./data/QBS103_GSE157103_genes.csv')
```

```r
# function to transpose/prepare the data from series_matrix and genes

prepare_dataframe <- function(series_matrix, genes){
  unique(series_matrix$age) # this one is covariate

  # let's merge these two data sets together

  # transpose the genese dataframe
  transposed_genes <- as.data.frame(t(genes))
  colnames(transposed_genes) <- genes$X
  transposed_genes <- transposed_genes[2:length(genes), ]


  for (i in colnames(transposed_genes)){
    series_matrix[[i]] <- as.numeric(transposed_genes[[i]]) # convert the transposed data to numeric
  }
  return(series_matrix)
}

dataframe <- prepare_dataframe(series_matrix, genes)
```

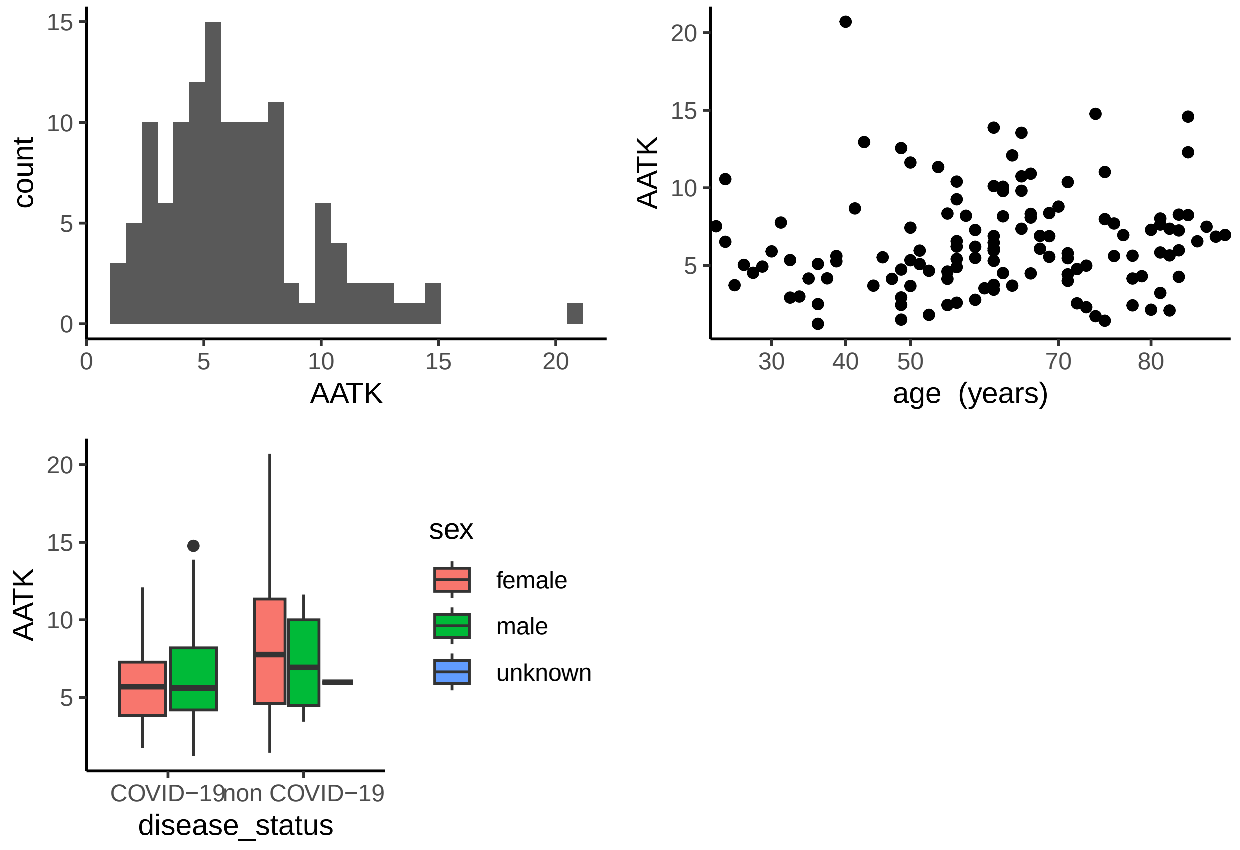
```r
# https://stackoverflow.com/questions/55132771/standard-eval-with-ggplot2-without-aes-string/55133909#5
# source used form !!sym (to pass a string as a symbol in an aes environment)


generate_plots <- function(dataframe, list_of_genes, continuous_variable, categorical_covariates)
  {
  # This the continuous variables
  # plots scatter

  for (gene in list_of_genes){
    print(gene)
    print(continuous_variable)
    scatter <- ggplot(dataframe, aes(x = !!sym(continuous_variable)
                                     , y = !!sym(gene))) +
    geom_point() +
    xlab(paste(continuous_variable, ' (years)')) +
    ylab(gene) +
    scale_x_discrete(breaks = seq(0, 100, by = 10)) +
    theme_classic()

    # this is for the two categorical variables
    # plot the box plot
    box <- ggplot(dataframe, aes(y = !!sym(gene),
                                 x = !!sym(categorical_covariates[1]),
                                 fill = !!sym(categorical_covariates[2]))) +
      geom_boxplot(names(c('COVID', 'NON-COVID'))) +
      labs(x = categorical_covariates[1], y = gene) +
      scale_x_discrete(labels = c("disease state: COVID-19" = "COVID-19",
                                  "disease state: non-COVID-19" = "non COVID-19")) +
      theme(axis.text.x = element_text(angle = 0, hjust = 2, size = 8))  +
      theme_classic()
    # plot the histogram
    print(gene)
    histo <- ggplot(dataframe, aes(x = !!sym(gene))) +
    geom_histogram() +
    theme_classic()

    print(ggarrange(histo, scatter, box, ncol = 2, nrow = 2))
  }
}



generate_plots(dataframe, c('AATK', 'A1BG', 'A2M'), 'age', c('disease_status', 'sex'))
```

```
## [1] "AATK"
## [1] "age"
## [1] "AATK"
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

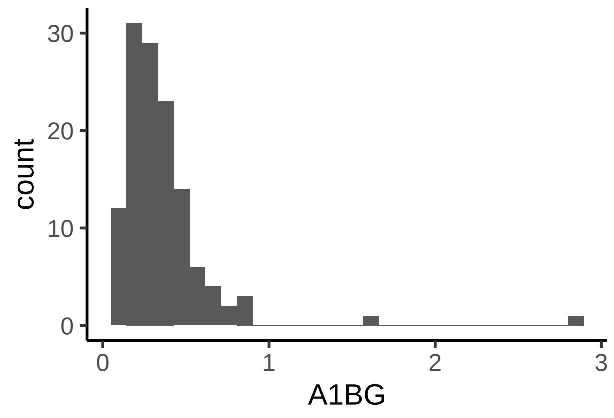
```
## [1] "A1BG"
## [1] "age"
## [1] "A1BG"
```

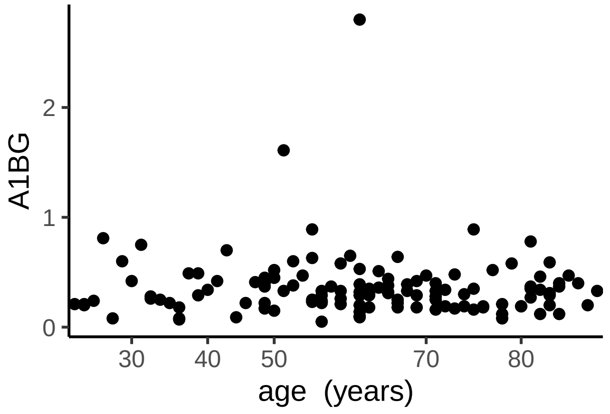
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

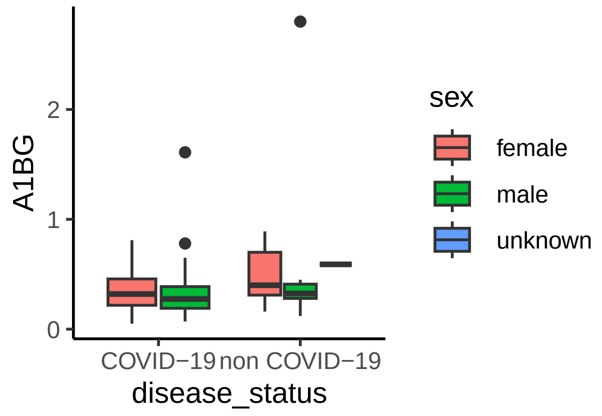


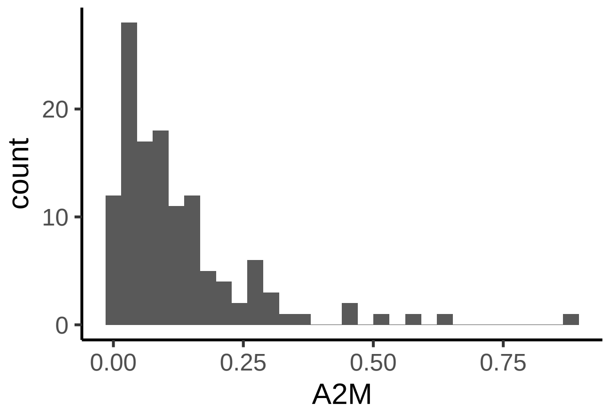
```
## [1] "A2M"
## [1] "age"
## [1] "A2M"
```

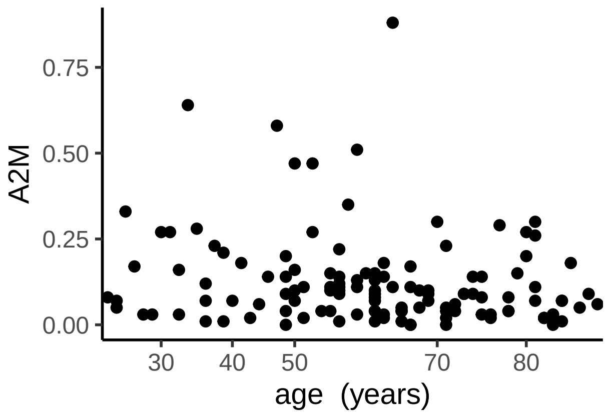
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

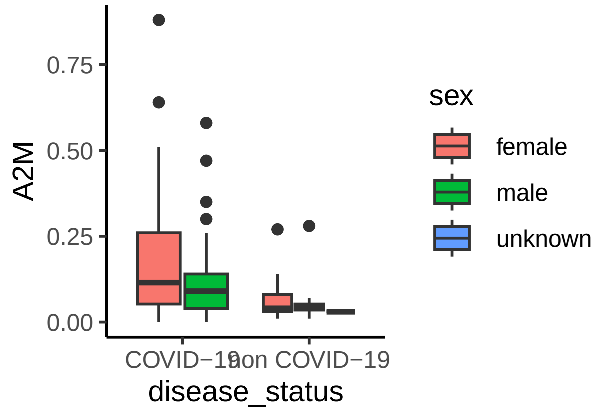







count
20
10
0
0.00
0.25
0.50
0.75
A2M


A2M
0.75
0.50
0.25
0.00
30
40
50
70
80
age (years)


A2M
0.75
0.50
0.25
0.00
COVID−19
non COVID−19
disease_status
sex
female
male
unknown