Final Project Submission 1

2024-07-23

```
# installing needed packages
library(tidyverse)
```

```
## -- 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 errors
```

```
library(reshape2)
```

```
##
## Attaching package: 'reshape2'
##
## The following object is masked from 'package:tidyr':
##
##     smiths
```

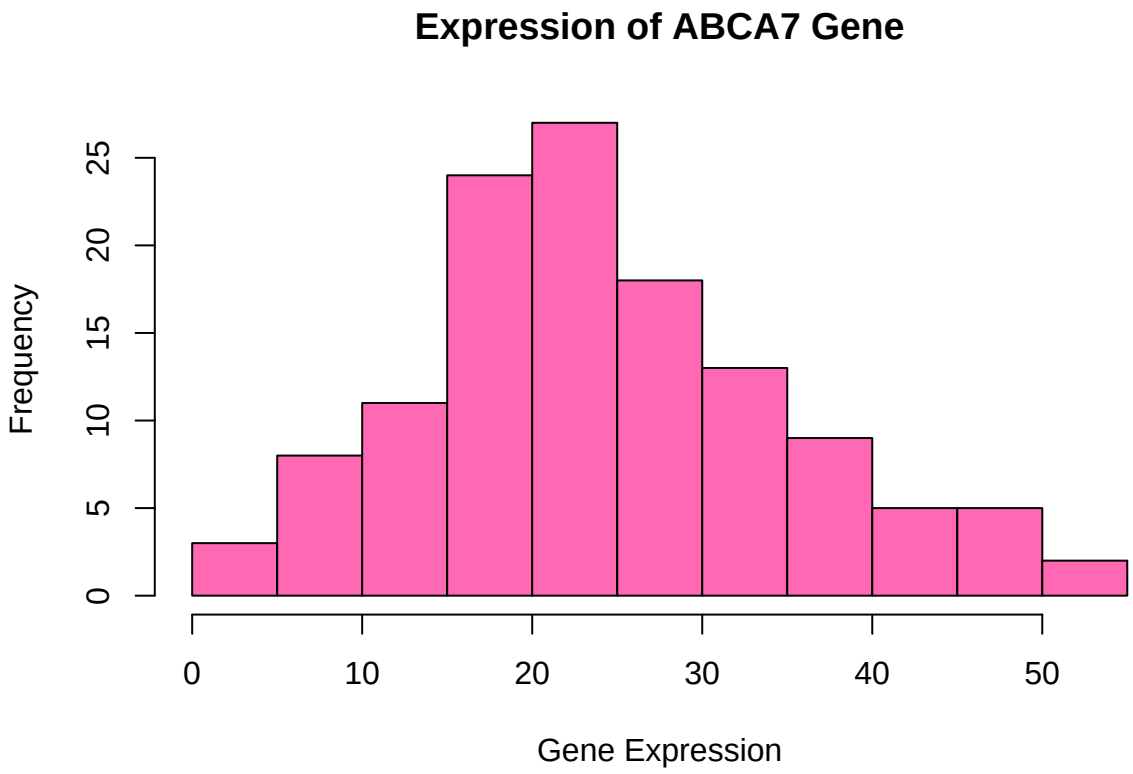
```
library(ggplot2)

# setting my working directory
setwd("/Users/hannahbahrampour/Desktop")

# checking to see where I am
getwd()
```

```
## [1] "/Users/hannahbahrampour/Desktop"
```

```
# using read.csv to read in both of the files and assign them to shorter variable names

genes <- read.csv(file = "QBS103_GSE157103_genes.csv")

series_matrix <- read.csv(file = "QBS103_GSE157103_series_matrix.csv")

# melting the genes data into the long format
# Jaini helped me understand the concept of melting and why it is
# necessary in this case
gene_long <- genes %>% tidyr::gather(key = "ParticipantID", value =
                                       "Expression", -X)

# rename a column in the series_matrix to match with genes_long
series_matrix <- series_matrix %>%
```

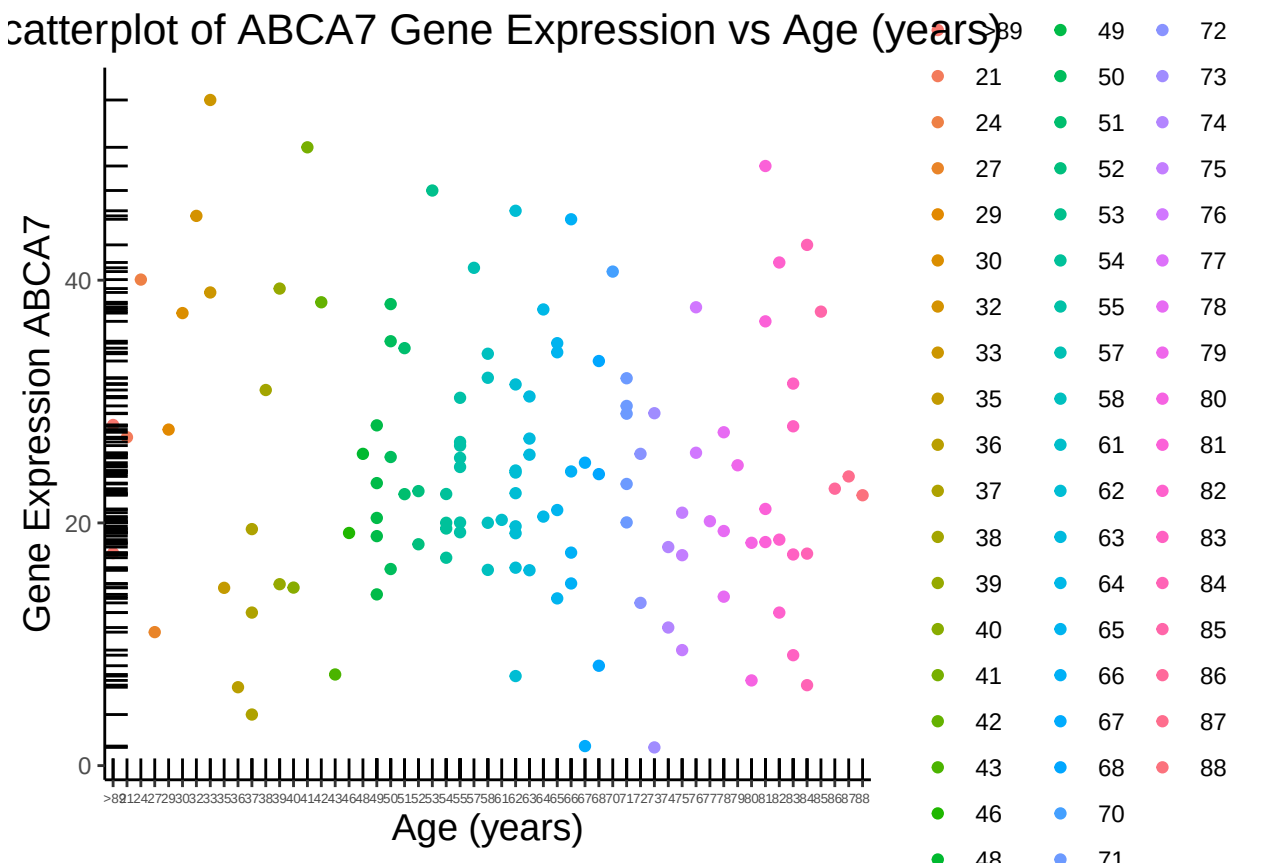
```
  rename(ParticipantID = participant_id)

# merge the data together
data_merged <- merge(gene_long, series_matrix, by = "ParticipantID")
```

```
# selecting my gene and filtering for it
# assign this clean selected gene data to a variable
clean_data <- data_merged %>%
  filter(X == "ABCA7") %>% # gene selection
  select(X, ParticipantID, Expression, age, sex, icu_status)

# ensure values are numeric if not already
clean_data$Expression <- as.numeric(clean_data$Expression)

# creating a histogram to show my gene expression
# making the histogram hot pink and labeling it
hist(clean_data$Expression, main = paste("Expression of ABCA7 Gene"),
breaks=10, col = "hotpink", xlab = "Gene Expression")
```

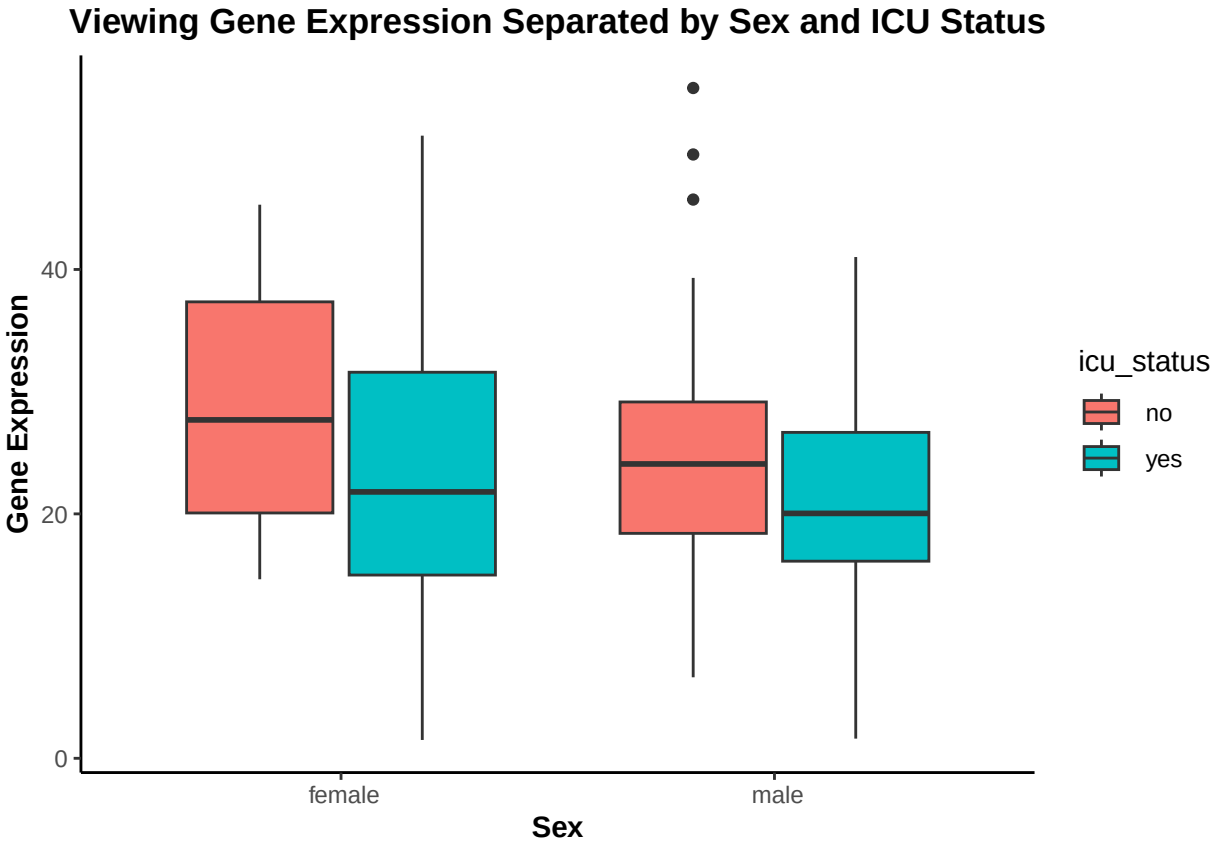



```
# creating scatterplot for gene expression and continuous covariate (age)
ggplot(clean_data, aes(x = age, y = Expression)) +
  geom_point(aes(color = age)) + # adding points and color dependent on age
  geom_rug() + # adding rug
  labs(title = "Scatterplot of ABCA7 Gene Expression vs Age (years)",
       x = "Age (years)", y = "Gene Expression ABCA7") + # label the scatterplot
  theme_classic() + # getting rid of the background grid
  theme( # adjusting text sizes
    plot.title = element_text(hjust = 0.5, size = 16),
    axis.title = element_text(size = 14),
    legend.title = element_text(hjust = 0.5),
```

```
    axis.text.x = element_text(size = 5) # trying to make the x-axis more readable
  )
```




```
# cleaning out data for unknown sex
new_clean_data <- clean_data %>%
  filter(sex == " male" | sex == " female")

# boxplot of gene expression separated by two categorical covariates (sex and ICU status)
ggplot(new_clean_data, aes(x = sex, y = Expression, fill = icu_status)) +
  geom_boxplot() + # adding in the boxplot
  labs(title = "Viewing Gene Expression Separated by Sex and ICU Status",
       x = "Sex", y = "Gene Expression") + # labeling things
  scale_alpha_manual(name = "ICU Status") +
  theme_classic() + # getting rid of background grid
  theme( # adjusting the title and axis title
    plot.title = element_text(hjust = 0.5, face = "bold"),
    axis.title = element_text(face = "bold")
  )
```


Viewing Gene Expression Separated by Sex and ICU Status
Gene Expression
40
20
0
female
male
Sex
icu_status
no
yes