Final_Submision_103

R Markdown

```
#Convert genes into gene series
library(readr)
setwd("/Users/deepthi/Documents/GitHub/Final_103_Project_LDG")
# read in genes file
Genes<- read_csv("QBS103_GSE157103_genes.csv")
```

```
## New names:
## Rows: 100 Columns: 127
## -- Column specification
## -------------------------------------------------------- Delimiter: "," chr
## (1): ...1 dbl (126): COVID_01_39y_male_NonICU, COVID_02_63y_male_NonICU,
## COVID_03_33y_...
## i Use `spec()` to retrieve the full column specification for this data. i
## Specify the column types or set `show_col_types = FALSE` to quiet this message.
## * `` -> `...1`
```

```
#read in Gene Series file
Genes_Series <- read_csv("QBS103_GSE157103_series_matrix.csv")
```

```
## Rows: 126 Columns: 25
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (21): participant_id, geo_accession, status, !Sample_submission_date, la...
## dbl  (4): channel_count, charlson_score, ventilator-free_days, hospital-free...
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
#transpose gene data to put participant id as rows and genes as columns
TGenes <-as.data.frame(t(Genes))
# rearrange table to make gene names as column names instead of row names
names(TGenes) <- TGenes[1,]
#remove repetitive first column with genes
TGenes <- TGenes[-1,]
#create a column with the row names( participant ids)to merge with gene series file
TGenes$participant_id <- row.names(TGenes)
# Merge Gene and Gene Series tables by participant_id
MergedGenes <- merge(TGenes,Genes_Series, by = 'participant_id', all = TRUE)
row.names(MergedGenes) <- MergedGenes$participant_id

# change ICU status values for proper formatting of clean tables/plots
```

```
MergedGenes$icu_status[MergedGenes$icu_status =='no'] = 'No'
MergedGenes$icu_status[MergedGenes$icu_status =='yes'] = 'Yes'
# change sex values for proper formatting of clean tables/plots
MergedGenes$sex[MergedGenes$sex=='female'] = 'Female'
MergedGenes$sex[MergedGenes$sex =='male'] = 'Male'
MergedGenes$sex[MergedGenes$sex =='unknown'] = 'Unknown'
```

Generate a table formatted in LaTeX of summary statistics for all the covariates you looked at and 2 additional continuous (3 total) and 1 additional categorical variable (3 total). (5 pts) o Stratifying by one of your categorical variables o Tables should report n (%) for categorical variables o Tables should report mean (sd) or median [IQR] for continuous variables

```
# load in libraries
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(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## 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
```

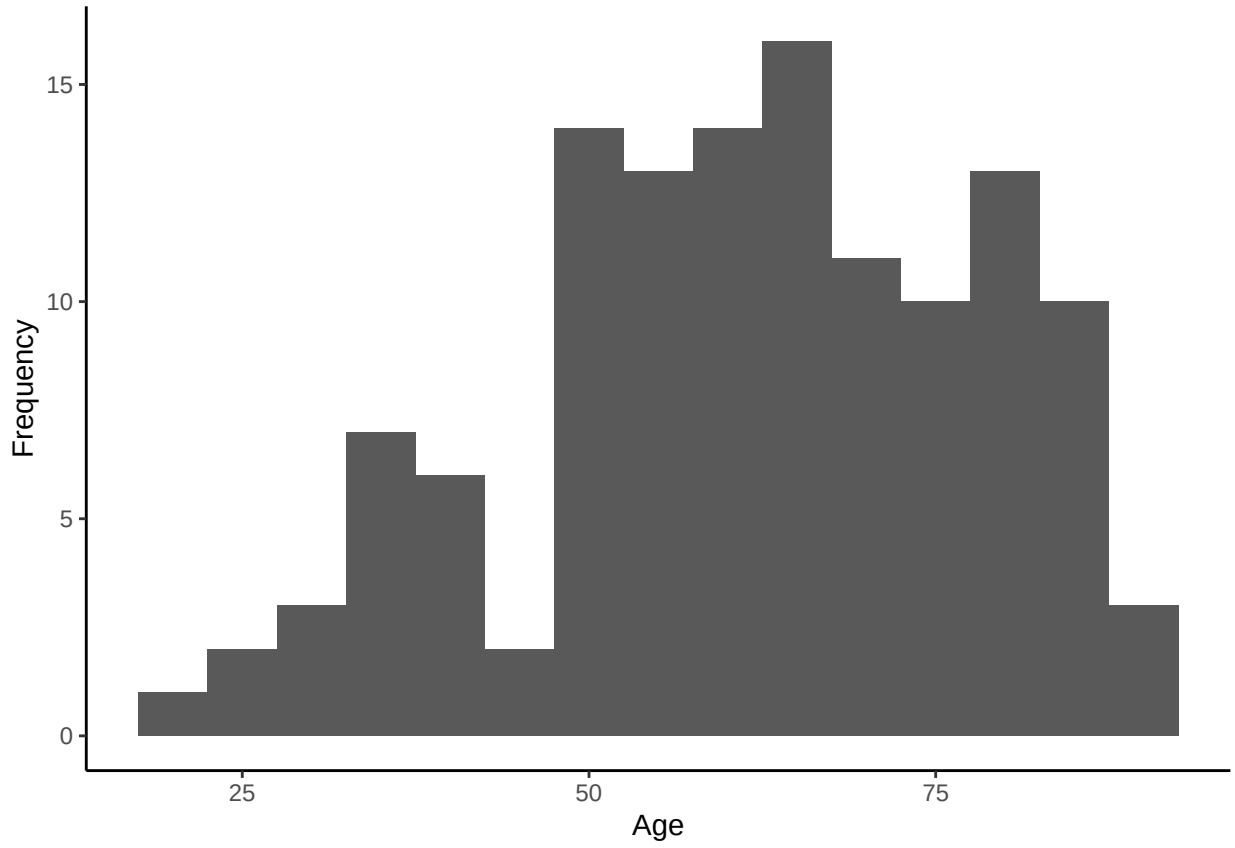
```
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
# create a subset of df with necessary continuous and categorical variables
Genes_CoVariate_Data_3_3 <- MergedGenes %>%
   select(age, sex,icu_status,disease_status,`hospital-free_days_post_45_day_followup`,`ventilator-free_
# convert the >89 age to 90 years old
Genes_CoVariate_Data_3_3$age[Genes_CoVariate_Data_3_3$age== ">89"] <- 90

#convert continuous column to numeric type to numeric data
Genes_CoVariate_Data_3_3$age <- as.numeric(Genes_CoVariate_Data_3_3$age)
```

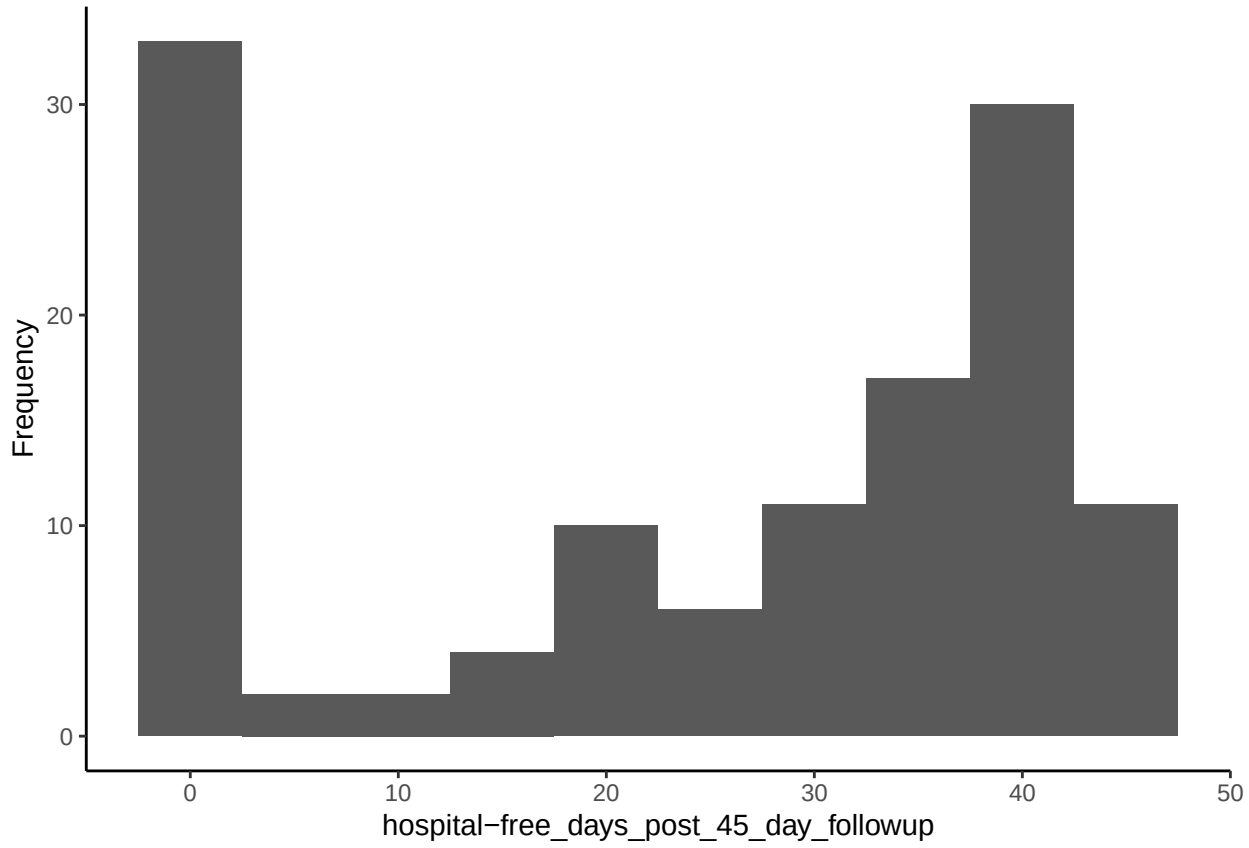
```
## Warning: NAs introduced by coercion
```

```
Genes_CoVariate_Data_3_3$`hospital-free_days_post_45_day_followup` <- as.numeric( Genes_CoVariate_Data_3
Genes_CoVariate_Data_3_3$`ventilator-free_days` <- as.numeric( Genes_CoVariate_Data_3_3$`ventilator-free


#check distribution of each of the continuous variables to decide whether to use mean/sd or median/IQR
Age <- ggplot(data = Genes_CoVariate_Data_3_3,aes(x = age))+
  geom_histogram(binwidth = 5)+
  labs(x = 'Age',y = 'Frequency') +
  theme_classic()
print(Age)
```

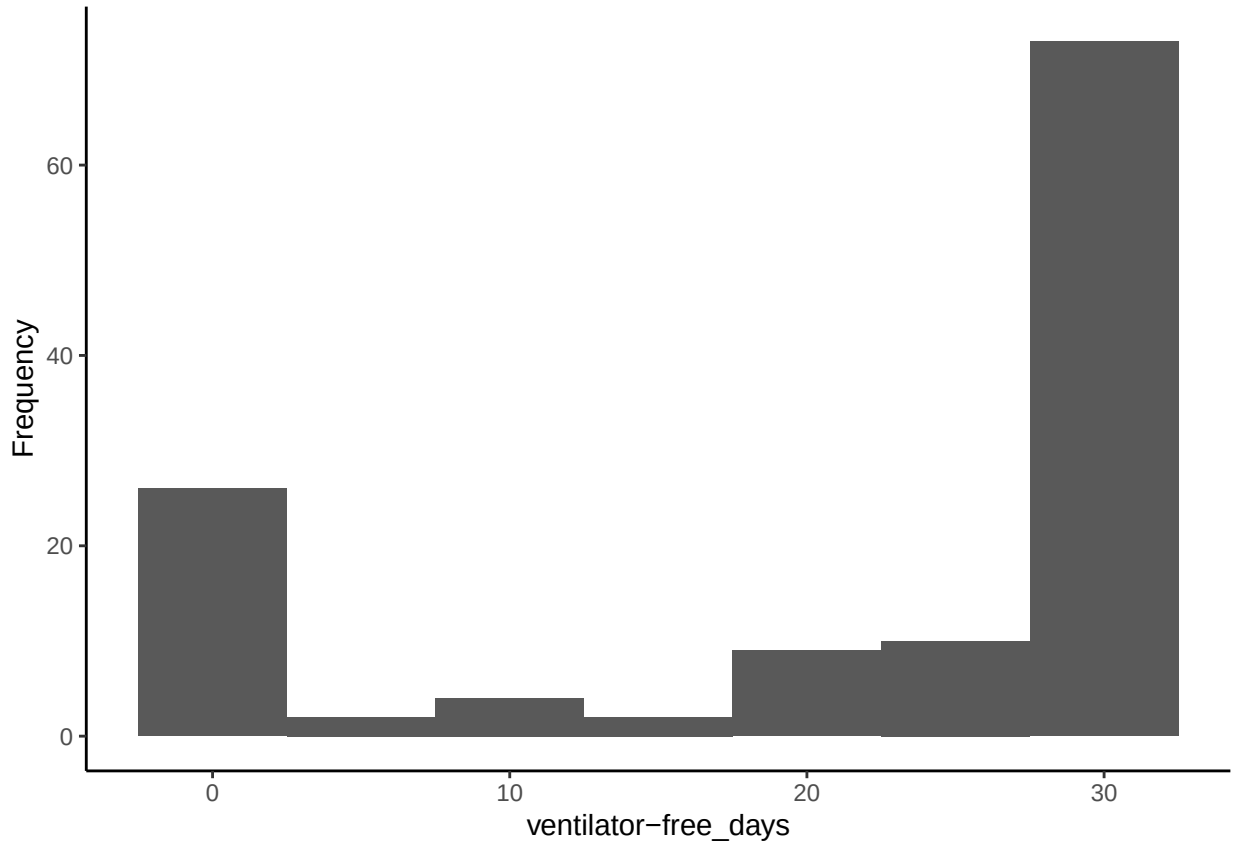
```
## Warning: Removed 1 row containing non-finite outside the scale range
## (‘stat_bin()‘).
```




```
hospital_free_days <- ggplot(data = Genes_CoVariate_Data_3_3,aes(x = `hospital-free_days_post_45_day_fo
  geom_histogram(binwidth = 5) +
  labs(x ='hospital-free_days_post_45_day_followup', y = 'Frequency') +
  theme_classic()
print(hospital_free_days)
```




```
ventilator_free_days <- ggplot(data = Genes_CoVariate_Data_3_3, aes(x = `ventilator-free_days`))+
    geom_histogram(binwidth = 5)+
    labs(x ='ventilator-free_days',y = 'Frequency')+
    theme_classic()
print(ventilator_free_days)
```




```
library(kableExtra)
```

```
##
## Attaching package: 'kableExtra'
```

```
## The following object is masked from 'package:dplyr':
##
##     group_rows
```

```
library(table1)
```

```
##
## Attaching package: 'table1'
```

```
## The following objects are masked from 'package:base':
##
##     units, units<-
```

```
#Format column names and factor values to create neat tables

# change col names
names(Genes_CoVariate_Data_3_3) <- c('Age',"Sex","ICU_Status", "Disease_Status","Hospital_Free_Days_Pos
# change disease status values
Genes_CoVariate_Data_3_3$Disease_Status[Genes_CoVariate_Data_3_3$Disease_Status =='disease state: non-C
```

```r
Genes_CoVariate_Data_3_3$Disease_Status[Genes_CoVariate_Data_3_3$Disease_Status =="disease state: COVID


# create a function that specifies what parameters to include for continuous values in the table1 gener
## documentation used to generate function : https://cran.r-project.org/web/packages/table1/vignettes/t
render_Calc<- function(x, namecol, ...) {
    if (!is.numeric(x)) return(render.categorical.default(x))
    what <- switch(namecol,
        # keep mean for age and median for hospital and ventilator free days based on distribution of v
        Age = "Mean (SD)",
        Hospital_Free_Days_Post_45_Day_Followup = "Median [Min, Max]",
        Ventilator_Free_Days = "Median [Min, Max]")
    parse.abbrev.render.code(c("", what))(x)
}


# create a summary table use tableone package to get summary statistics and include render_Calc() funct
Sum_Tab <-table1(~Age+ `Hospital_Free_Days_Post_45_Day_Followup`+ `Ventilator_Free_Days` + ICU_Status+ D
Sum_Tab
```

	Female	Male	Unknown
	(N=51)	(N=74)	(N=1)
Age			
Mean (SD)	59.9 (18.3)	62.7 (14.7)	NA
Hospital_Free_Days_Post_45_Day_Followup			
Median [Min, Max]	34.0 [0, 44.0]	28.0 [0, 44.0]	30.0 [30.0, 30.0]
Ventilator_Free_Days			
Median [Min, Max]	28.0 [0, 28.0]	28.0 [0, 28.0]	28.0 [28.0, 28.0]
ICU_Status			
No	27 (52.9%)	33 (44.6%)	0 (0%)
Yes	24 (47.1%)	41 (55.4%)	1 (100%)
Disease_Status			
COVID-19	38 (74.5%)	62 (83.8%)	0 (0%)
Non-COVID-19	13 (25.5%)	12 (16.2%)	1 (100%)

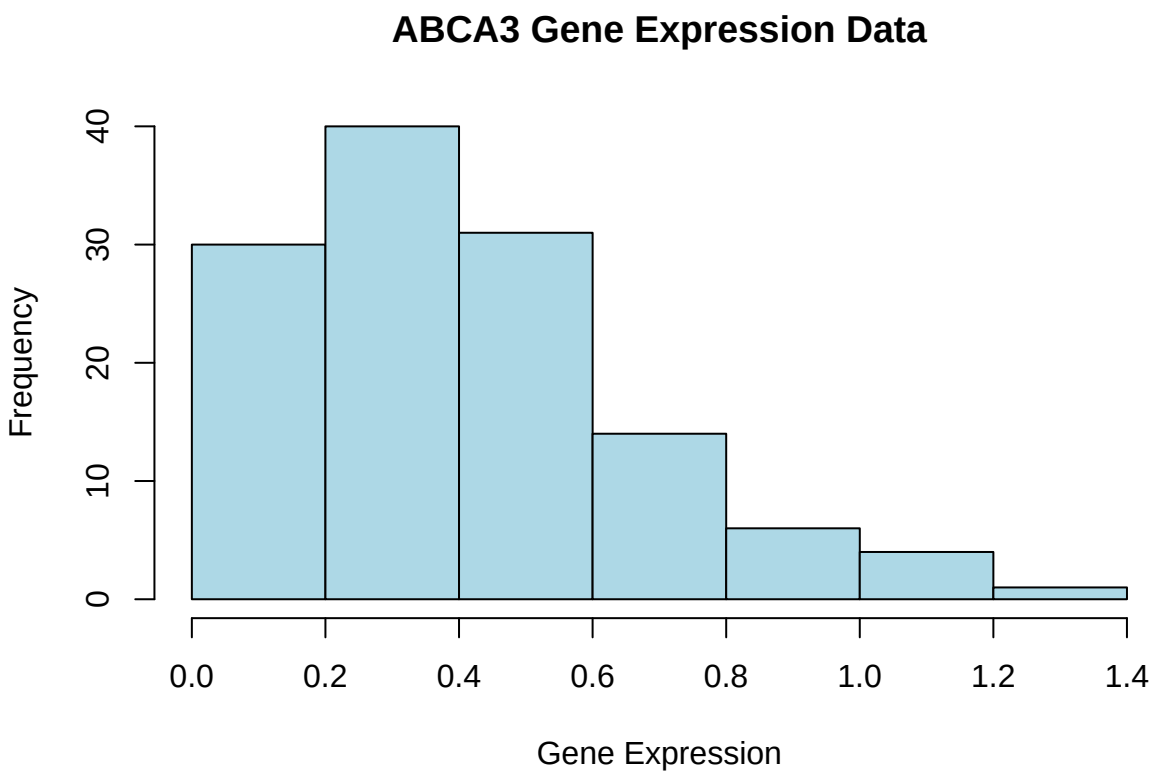
```r
# use kable() to generate latex format of summary statistics table
tab <- kable(x = Sum_Tab, caption = 'Summary Table',
             format = 'latex',booktabs = T,
            # col.names = c("Female", "Male","Unknown", "Overall"),
             align = c('l','r'),escape = T) %>%
  add_indent(positions = c(3,4),level_of_indent = 1)
```

Generate final histogram, scatter plot, and boxplot from submission 1 (i.e. only for your first gene of interest) incorporating all feedback from your presentations)

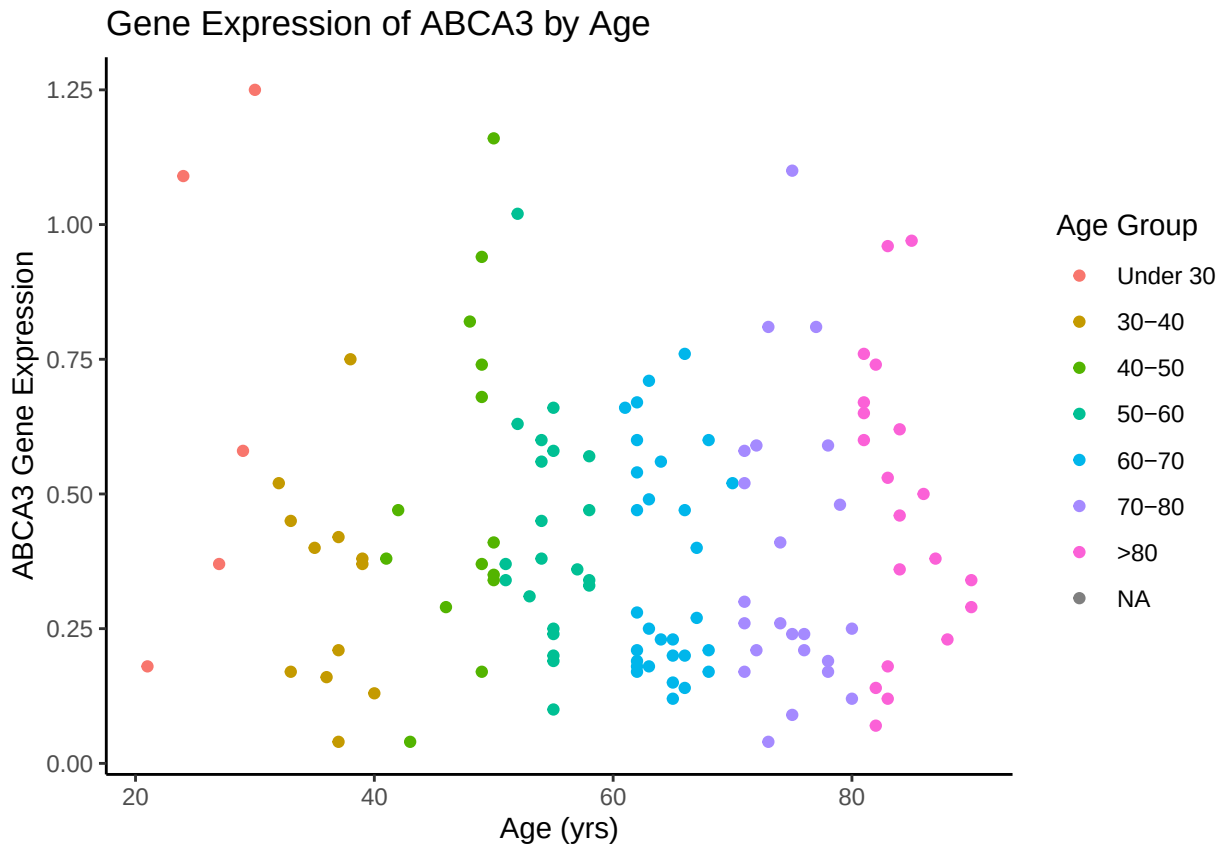
```r
library(dplyr)
library(ggplot2)

# function to generate final histogram, scatter plot, and boxplot from submission 1
GenePlots <- function(df, geneName, Cont, Cat1, Cat2) {
  # Pull out gene expression data, continuous co variate, and categorical data
  geneName <- geneName[[1]]  # Assumes geneName is a single-element list
  Genes_CoVariate_Data <- df %>%
    select(all_of(c(geneName, Cont, Cat1, Cat2))) %>%
    as.data.frame()
  # Convert gene expression column to numeric
  Genes_CoVariate_Data[[geneName]] <- as.numeric(Genes_CoVariate_Data[[geneName]])
  # Plot histogram of gene expression data
  histograms <- hist(Genes_CoVariate_Data[[geneName]],
      main = paste(geneName, 'Gene Expression Data'), xlab = 'Gene Expression', col = "light blue")
  print(histograms)
  # Replace age values greater than 89 with 90
  Genes_CoVariate_Data[[Cont]][Genes_CoVariate_Data[[Cont]] == ">89"] <- 90
  # Convert continuous column to numeric
  Genes_CoVariate_Data[[Cont]] <- as.numeric(Genes_CoVariate_Data[[Cont]])

  # Create age groups
  Genes_CoVariate_Data$AgeGroup <- cut(Genes_CoVariate_Data[[Cont]],
                                       breaks = c(0, 30, 40, 50, 60, 70, 80, 90),
                                       labels = c('Under 30', '30-40', '40-50', '50-60', '60-70', '70-8

  # Plot scatter plot of gene expression by age
  Scatterplot <- ggplot(Genes_CoVariate_Data, aes(x = !!sym(Cont), y = !!sym(geneName), color = AgeGroup
    geom_point() +
    labs(title = paste('Gene Expression of', geneName, 'by Age'),
         x = 'Age (yrs)',
         y = paste(geneName, 'Gene Expression'),
         color = "Age Group") +
    theme_classic()
  print(Scatterplot)

# add color pallete to color boxplots
  colorPalette = c('light blue',' maroon')

# create box plots to show Gene Expression of ABCA3 by Sex and ICU Status
  boxPlots <- ggplot(Genes_CoVariate_Data, aes(x = !!sym(Cat1), y=!!sym(geneName), fill= !!sym(Cat2)) )+
    # Add box plot
    geom_boxplot() +
    # add colors
    scale_fill_manual(values= colorPalette) +
    # Change axis labels
    labs(title =paste('Gene Expression of',geneName,'by Sex and ICU Status'), x = 'Sex',y = 'Gene Expres
    theme_classic()
  print(boxPlots)}

# specify gene name 'ABCA3'
GeneNames <- c('ABCA3')
#run Gene Plots function
```

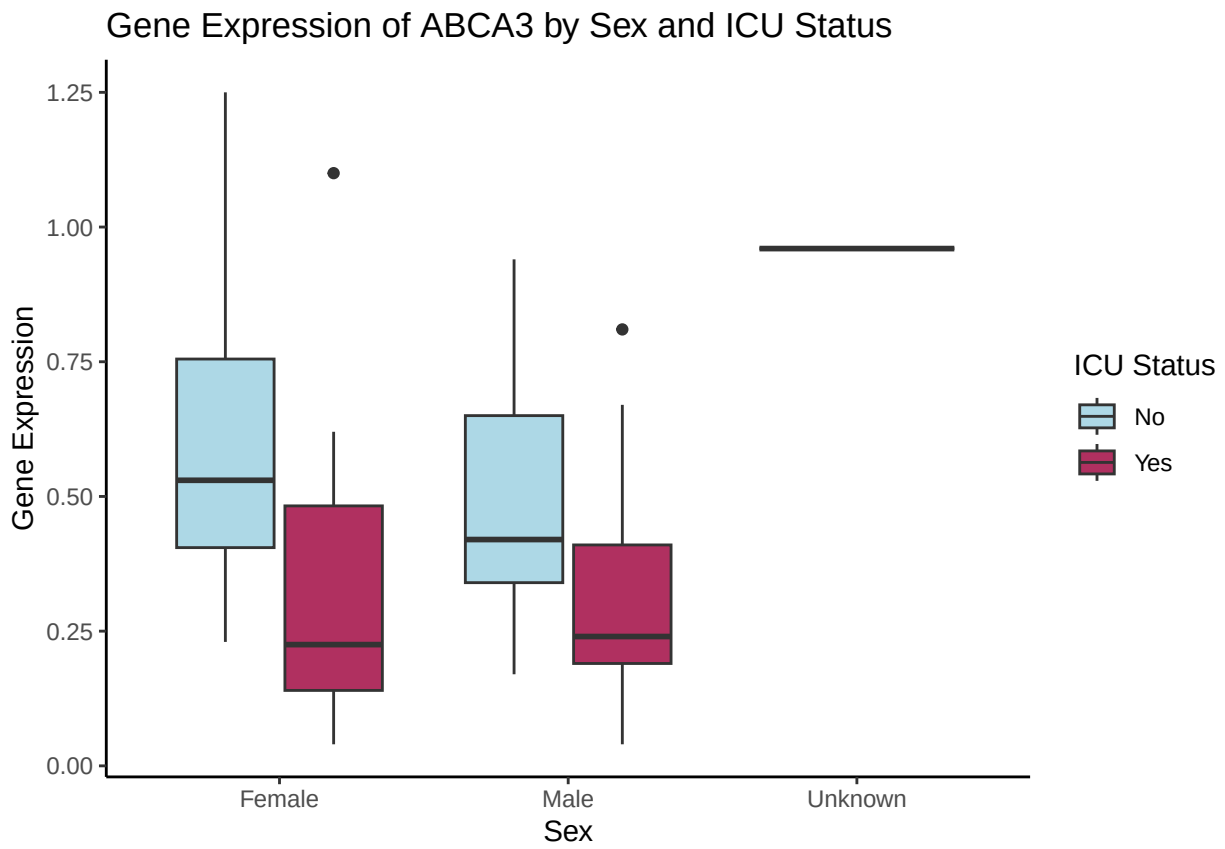
```
GenePlots(MergedGenes, GeneNames, 'age', 'sex', 'icu_status')
```

```
## $breaks
## [1] 0.0 0.2 0.4 0.6 0.8 1.0 1.2 1.4
##
## $counts
## [1] 30 40 31 14  6  4  1
##
## $density
## [1] 1.19047619 1.58730159 1.23015873 0.55555556 0.23809524 0.15873016 0.03968254
##
## $mids
## [1] 0.1 0.3 0.5 0.7 0.9 1.1 1.3
##
## $xname
## [1] "Genes_CoVariate_Data[[geneName]]"
##
## $equidist
## [1] TRUE
##
## attr(,"class")
## [1] "histogram"

## Warning in GenePlots(MergedGenes, GeneNames, "age", "sex", "icu_status"): NAs
## introduced by coercion
```



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

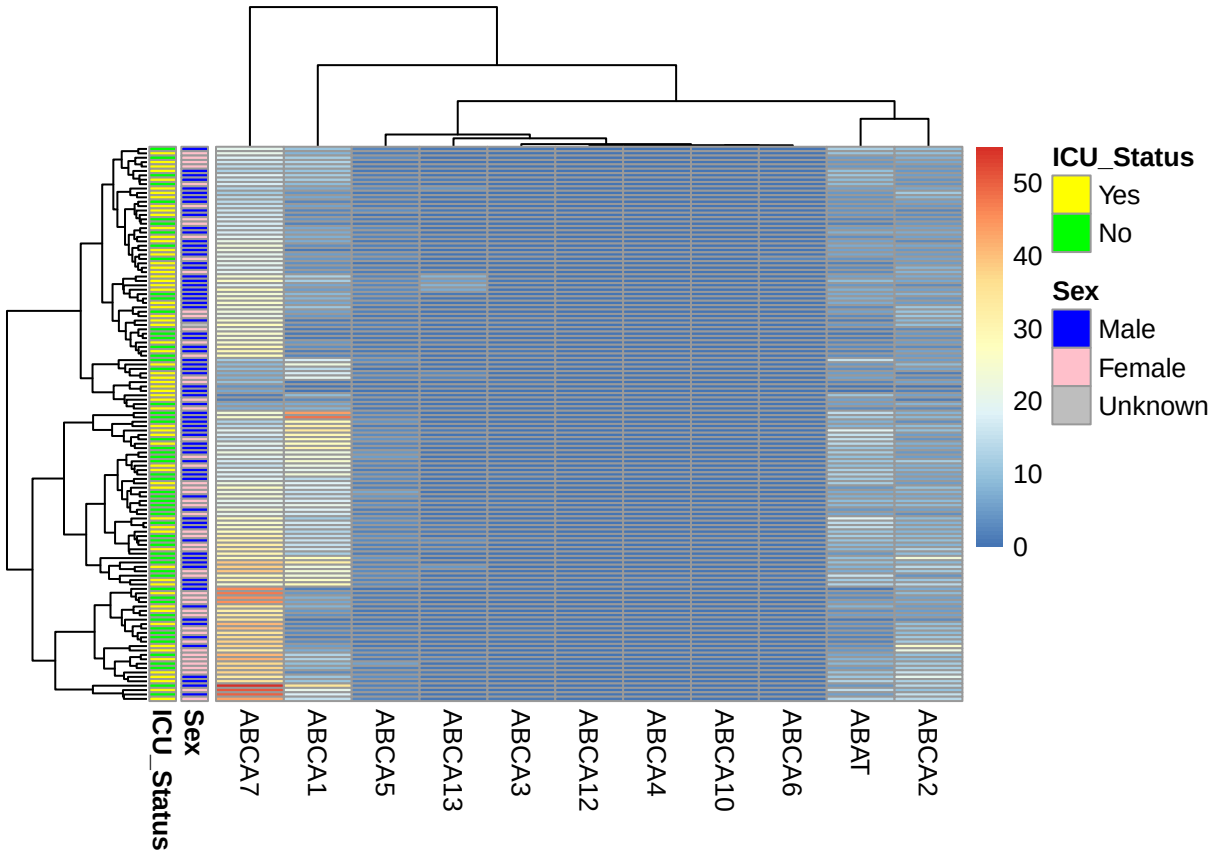




Generate a heatmap (5 pts) o Heatmap should include at least 10 genes o Include tracking bars for the 2 categorical covariates in your boxplot o Heatmaps should include clustered rows and columns

```
library(pheatmap)

# create annotation row data by extracting sex and ice_status col data
annotation_rows<- data.frame(
  Sex = MergedGenes$sex,
  ICU_Status = MergedGenes$icu_status
)

# make row names of original table and annotation_rows table same
rownames(annotation_rows) <- rownames(MergedGenes)

#create a color list for each of the values in in sex
color1 <- c( "Male" = "blue","Female" = "pink", "Unknown" = "gray")
#create a color list for each of the values in icu status
color2 <- c("Yes" = 'yellow', "No" = 'green')

# assign color list to categorical variables
annotation_colorss <- list(
  Sex = color1,
  ICU_Status = color2
)

#Use apply function to convert all 10 selected gene columns to numeric format
```

```
Genes_numeric_df<- as.data.frame(apply(MergedGenes[,31:41],MARGIN = 2, function(x) {as.numeric(x)}))

# make row names of original table and 10 genes subset table same
row.names(Genes_numeric_df) <- rownames(MergedGenes)

# generate heat map and cluster rows and cols by euclidean algorithm, and add annotations for sex and i
pheatmap(mat=Genes_numeric_df,
         show_rownames = F,
         clustering_distance_cols = 'euclidean',
         clustering_distance_rows = 'euclidean',
         annotation_row = annotation_rows,
         annotation_colors = annotation_colorss)
```



Going through the documentation for ggplot2, generate a plot type that we did not previously discuss in class that describes your data in a new and unique way

```
# create a subset of data with gene of interest, age, sex, and icu status
Genes_CoVariate_Data <- MergedGenes %>%
  select(ABCA3,age,sex,icu_status)

#install.packages("ggbeeswarm")

#load ggbeeswarm package
library(ggbeeswarm)

# ensure gene expression data is numeric
```

```r
Genes_CoVariate_Data$ABCA3 <- as.numeric(Genes_CoVariate_Data$ABCA3 )

# Create a swarm plot with age, sex, and icu_status
ggplot(Genes_CoVariate_Data, aes(x = interaction(sex, icu_status), y = ABCA3, color = interaction(sex, i
  # change point size
  geom_beeswarm(alpha = 1) +
  #change main, x, and y axis titles
  labs(title = "Beeswarm Plot of ABCA3 Gene Expression by Sex and ICU Status",
       x = "Sex and ICU Status",
       y = "Gene Expression", color= "Sex and ICU Status (Sex.ICU_Status)") +
  theme_minimal()
```