Final_Project_Presentation_1

Sai_Lakkireddy

2023-07-24

R Markdown

```
setwd("../")
data_gene_Expression <- read.csv("data/QBS103_finalProject_geneExpression.csv", header=TRUE)
# head(data_gene_Expression)


data_meta <- read.csv("data/QBS103_finalProject_metadata.csv", header=TRUE)
#head(data_meta)

#Since we have so many columns that are the data we need to combine our df with other file, we convert


data_gene_Expression.longForm <- data_gene_Expression %>%
  pivot_longer(cols = starts_with(c("COVID_","NONCOVID_")),
  names_to = "participant_id",
  values_to = "gene_expression_value"
  )

#make a final data frame by combining the two data sets

final_df <- data_gene_Expression.longForm %>% inner_join( data_meta,
           by=c('participant_id'))
```

Including Plots

You can also embed plots, for example:

```
#rename with x column with gene

final_df <- rename(final_df, gene = X)
#convert all the unknown strings in the data from to NAs
#final_df$apacheii <- na_if(final_df$apacheii, ' unknown')

final_df[, 16:27][final_df[, 16:27] == ' unknown' | final_df[, 16:27] == 'unknown'] <- NA

#format the disease status column to just include the status
```

```r
final_df$disease_status <- sub('disease state: ', '', final_df$disease_status)

#convert the column type of disease_status, sex, icu_status and mechanical_ventilation to factor
final_df <- final_df %>%
  mutate_at(vars(disease_status, sex, icu_status, mechanical_ventilation), as.factor)

#the class of age, charlson_score is character where it should be numerical



#Convert it to integer

#final_df$age <- as.integer(final_df$age)

final_df <- final_df %>%
  mutate_at(vars(age, apacheii,ferritin.ng.ml., crp.mg.l., ddimer.mg.l_feu., procalcitonin.ng.ml.., lac
```

```
## Warning: There were 2 warnings in `mutate()`.
## The first warning was:
## i In argument: `age = .Primitive("as.integer")(age)`.
## Caused by warning:
## ! NAs introduced by coercion
## i Run `dplyr::last_dplyr_warnings()` to see the 1 remaining warning.
```

```r
#lets see the unique genes are there in the data frame
unique(final_df$gene)
```

```
##   [1] "APOA1"        "APOA2"        "APOM"         "PRTN3"        "LCN2"
##   [6] "CD24"         "BPI"          "CTSG"         "DEFA1"        "DEFA4"
##  [11] "MMP8"         "MPO"          "AGT"          "FBLN5"        "NID1"
##  [16] "SERPINB1"     "GPLD1"        "CLEC3B"       "VWF"          "A1BG"
##  [21] "A1CF"         "A2M"          "A2ML1"        "A3GALT2"      "A4GALT"
##  [26] "A4GNT"        "AAAS"         "AACS"         "AADAC"        "AADACL2"
##  [31] "AADACL3"      "AADACL4"      "AADAT"        "AAGAB"        "AAK1"
##  [36] "AAMDC"        "AAMP"         "AANAT"        "AAR2"         "AARD"
##  [41] "AARS1"        "AARS2"        "AARSD1"       "AASDH"        "AASDHPPT"
##  [46] "AASS"         "AATF"         "AATK"         "ABAT"         "ABCA1"
##  [51] "ABCA10"       "ABCA12"       "ABCA13"       "ABCA2"        "ABCA3"
##  [56] "ABCA4"        "ABCA5"        "ABCA6"        "ABCA7"        "ABCA8"
##  [61] "ABCA9"        "ABCB1"        "ABCB10"       "ABCB11"       "ABCB4"
##  [66] "ABCB5"        "ABCB6"        "ABCB7"        "ABCB8"        "ABCB9"
##  [71] "ABCC1"        "ABCC10"       "ABCC11"       "ABCC12"       "ABCC2"
##  [76] "ABCC3"        "ABCC4"        "ABCC5"        "ABCC6"        "ABCC8"
##  [81] "ABCC9"        "ABCD1"        "ABCD2"        "ABCD3"        "ABCD4"
##  [86] "ABCE1"        "ABCF1"        "ABCF2"        "ABCF2-H2BE1"  "ABCF3"
##  [91] "ABCG1"        "ABCG2"        "ABCG4"        "ABCG5"        "ABCG8"
##  [96] "ABHD1"        "ABHD10"       "ABHD11"       "ABHD12"       "ABHD12B"
## [101] "ABHD13"       "ABHD14A"      "ABHD14A-ACY1" "ABHD14B"      "ABHD15"
## [106] "ABHD16A"      "ABHD16B"      "ABHD17A"      "ABHD17B"      "ABHD17C"
## [111] "ABHD18"       "ABHD2"        "ABHD3"        "ABHD4"        "ABHD5"
## [116] "ABHD6"        "ABHD8"        "ABI1"         "ABI2"
```

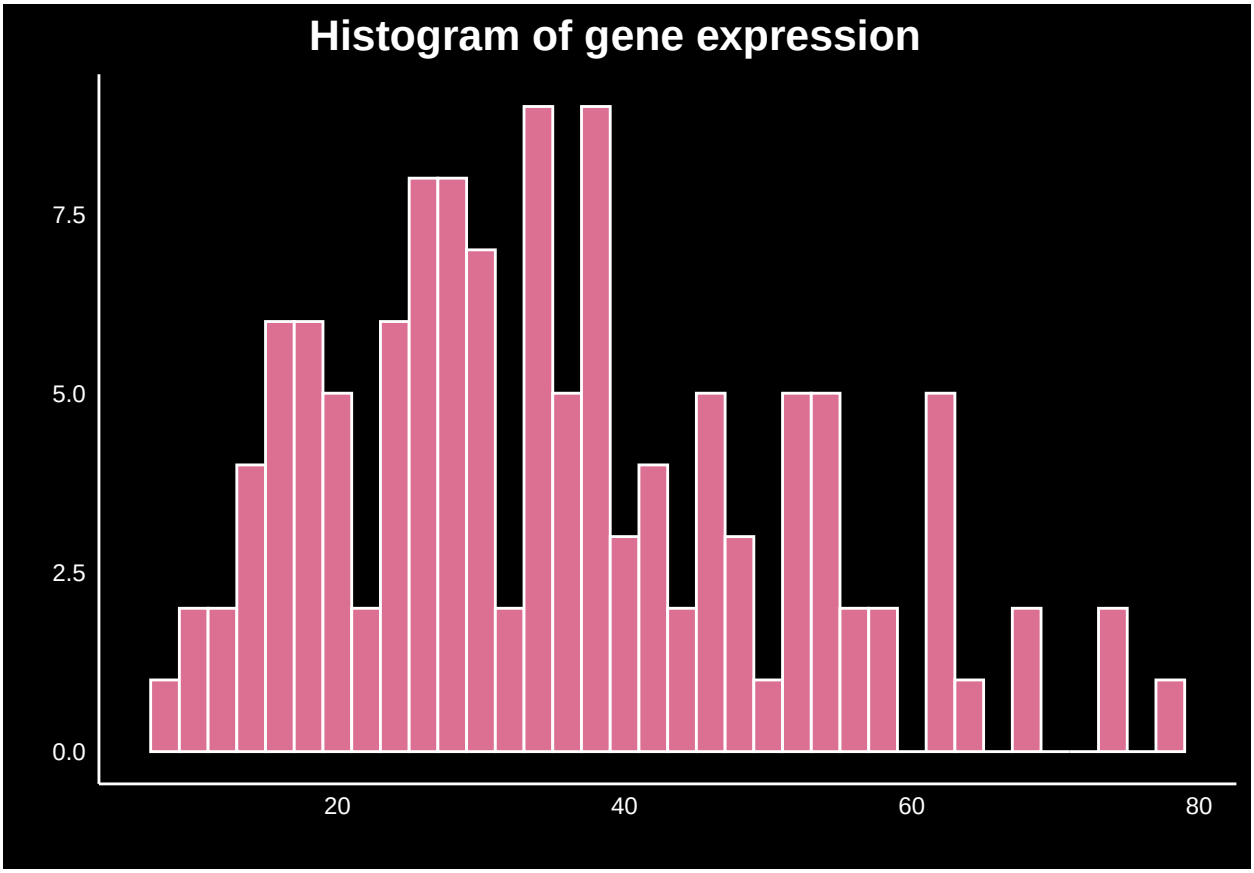
```r
#frequency_df <- data.frame('gene' = unique(final_df$gene),'')
#frequency_df <- as.data.frame(table(final_df$gene))
```

Identify one gene, one continuous covariate, and two categorical covariates in the provided dataset.

```r
# i will choose the  gene
# my one  continuous covariate would be age and the two categorical covariates would be ICU status and
#ABCB10


final_subset <- final_df[final_df$gene == 'AAMP', c('gene','gene_expression_value','age', 'icu_status',
```

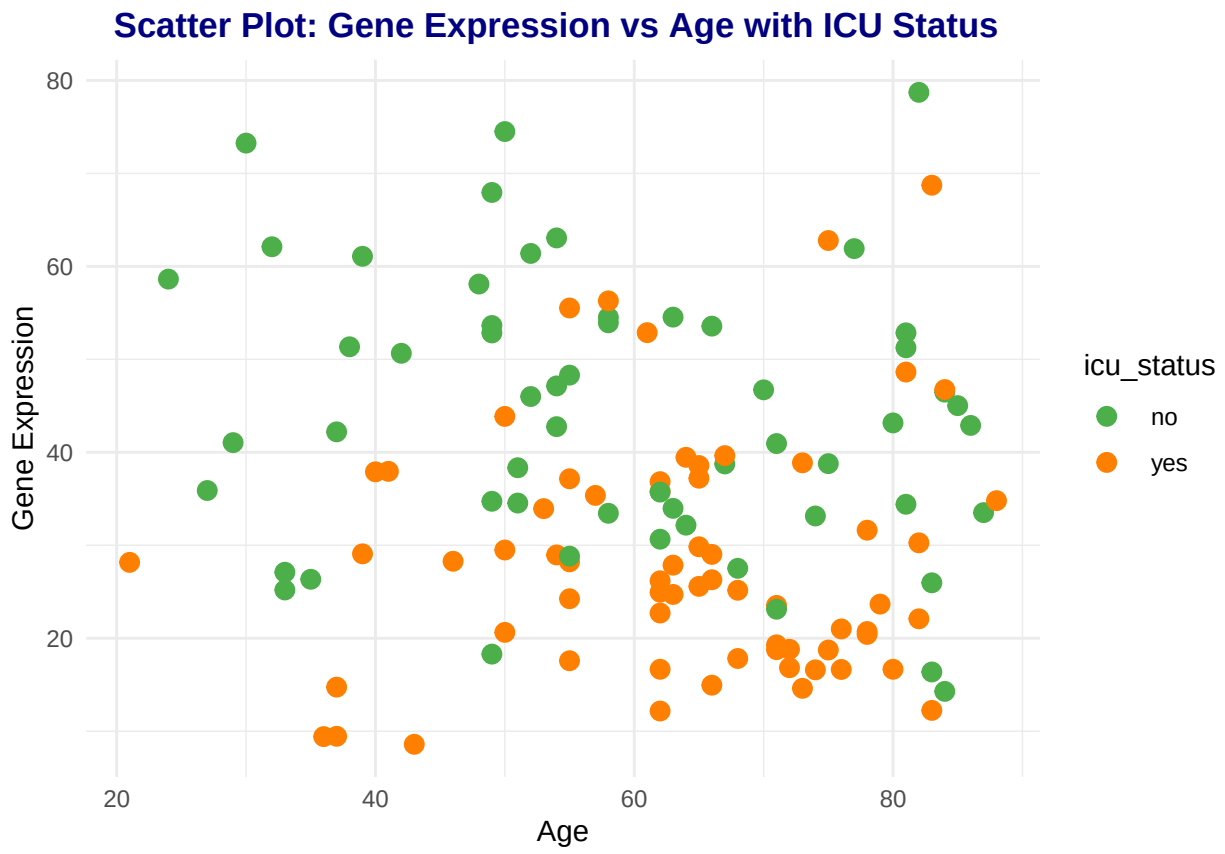
Note that the `echo = FALSE` parameter was added to the code chunk to prevent printing of the R code that generated the plot.

```r
# Create the histogram plot for Gene Expression
ggplot(final_subset, aes(x = gene_expression_value))  +
  geom_histogram(binwidth = 2, color = "white", fill = "#DB7093") +
  theme_minimal() +
    theme(
    panel.background = element_rect(fill = "black"),
    plot.background = element_rect(fill = "black"),
    axis.line = element_line(color = "white"),
    axis.text = element_text(color = "white"),
    panel.grid = element_blank(),
    plot.title = element_text(color = "white", size = 16, face = "bold", hjust = 0.4)
  ) +
  ggtitle("Histogram of gene expression") +
  xlab("gene expression")+
  ylab("Frequency")
```

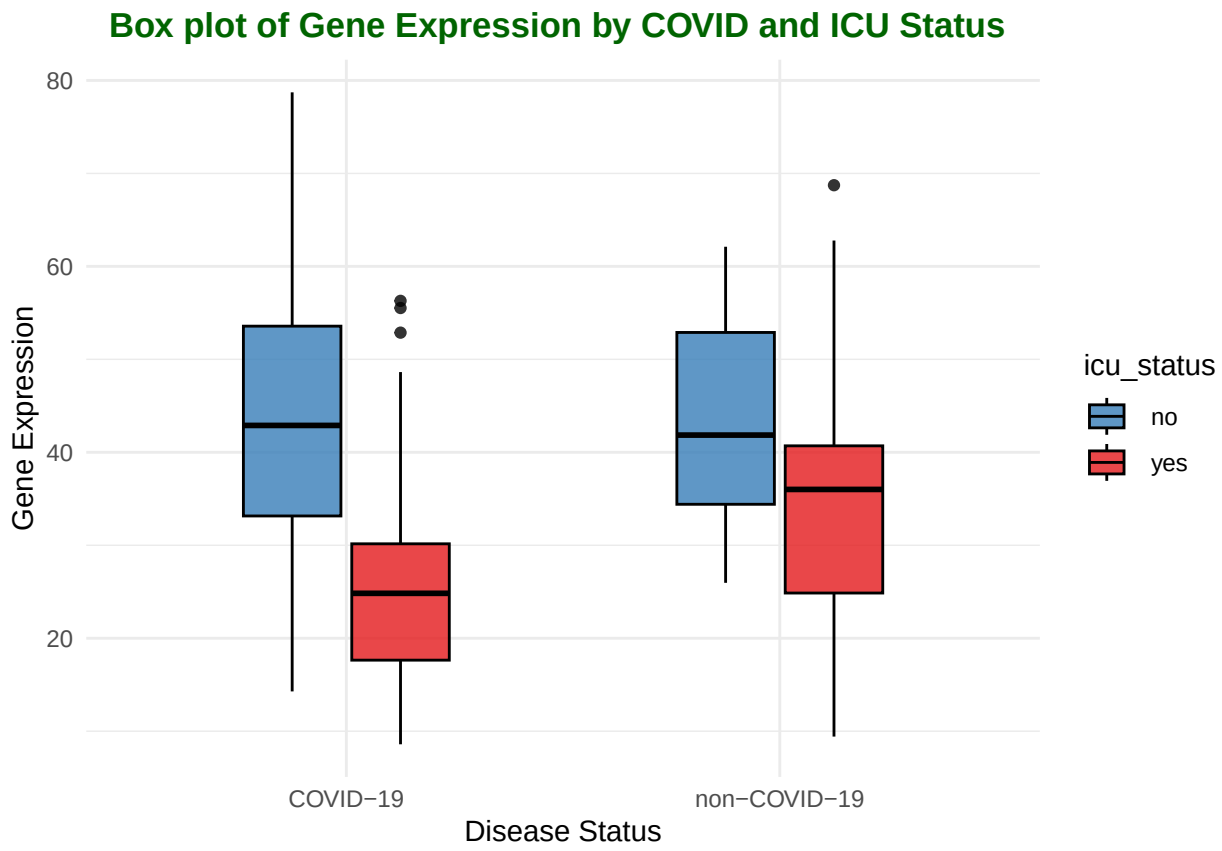



```
# ggplot(final_subset, aes(x = age, y = gene_expression_value)) +
#   geom_point(color = "blue") +
#   ggtitle("Scatter Plot of Gene Expression vs. Age") +
#   xlab("Age") +
#   ylab("Gene Expression")



my_colors_1 <- c("#4DAF4A", "#FF7F00")


# Create the scatter plot with custom color scheme
ggplot(final_subset, aes(x = age, y = gene_expression_value, color = icu_status)) +
  geom_point(size = 3) +
  scale_color_manual(values = my_colors_1) +
  theme_minimal() +
      theme(
    plot.title = element_text(color = "navy", size = 13, face = "bold", hjust = 0.4)
  ) +
  ggtitle("Scatter Plot: Gene Expression vs Age with ICU Status") +
  xlab("Age") +
  ylab("Gene Expression")
```

```
## Warning: Removed 2 rows containing missing values (‘geom_point()‘).
```




```r
my_colors_2 <- c("#377eb8", "#e41a1c")
ggplot(final_subset, aes(x = disease_status, y = gene_expression_value, fill = icu_status)) +
  geom_boxplot(color = "black", width = 0.5, alpha = 0.8) +
  scale_fill_manual(values = my_colors_2) +
  theme_minimal() +
    theme_minimal() +
      theme(
    plot.title = element_text(color = "darkgreen", size = 13, face = "bold", hjust = 0.4)
  )+
  ggtitle("Box plot of Gene Expression by COVID and ICU Status") +
  xlab("Disease Status") +
  ylab("Gene Expression")
```


Box plot of Gene Expression by COVID and ICU Status
80
60
40
20
Gene Expression
COVID−19
non−COVID−19
Disease Status
icu_status
no
yes