

QBS103

2024-07-29

```
# Select the first gene
gene <- genes_data[1, ]
# save the gene name
gene_name <- as.character(gene[1,1])
# transpose the selected gene for better merging the dataset and rename the column name
transposed_gene <- transpose(gene[-1]) %>% rename(!!gene_name := V1)
# combine two datasets with participant_id and gene
# such that we have a transposed gene dataset with two column, prepare for merging
transposed_gene_data <- cbind(participant_id = colnames(gene[-1]), transposed_gene)
# merge the dataset
merged_data <- merge(metadata, transposed_gene_data, by = "participant_id")
```

```
# select one continuous covariate and two categorical covariate
continuous_covariate <- "age"
categorical_covariate1 <- "sex"
categorical_covariate2 <- "disease_status"
```

```
### Plots ###
# Histogram for gene expression
gene_histogram <- ggplot(merged_data, aes(x = !!sym(gene_name))) +
  geom_histogram(aes(y = after_stat(density)),
    binwidth = 0.05, color = "#1dddeb",
    fill = "#1dddeb", alpha = 0.7) +
  geom_density(aes(y = after_stat(density)), color = "white", linewidth = 1) +
  labs(x = gene_name, y = "Density") +
  theme_classic() +
  theme(
    plot.background = element_rect(fill = '#333333'),
    panel.background = element_rect(fill = '#333333'),
    text = element_text( color = 'white', size = 20),
    axis.title = element_text(size = 14),
    axis.text = element_text(size = 12, color = 'white'),
    axis.ticks = element_line(color = "white"),
    plot.title = element_text(size = 16, hjust = 0.5, color = 'white')
  ) +
  ggtitle("Histogram of Gene Expression")

# Save the histogram
ggsave("plots/gene_histogram.png", plot = gene_histogram, width = 16, height = 9)

# Print the histogram
print(gene_histogram)
```



This aesthetic is inspired by <https://waterdata.usgs.gov/blog/ggplot-jazz/>

Notice that there is a ":" value in the age, we have to filter it out first
merged_data\$age <- as.numeric(merged_data\$age)

Warning: NAs introduced by coercion

```
filtered_data <- merged_data %>% filter(!is.na(age))
# Scatter Plot
scatter_plot <- ggplot(filtered_data, aes(x = age, y = !!sym(gene_name))) +
  geom_point(color = "#F8B195", size = 2) +
  geom_smooth(method = "loess", color = "#95ebf8") + # Add a LOESS curve
  geom_rug(color = "white") + # Add a rug plot
  labs(x = "Age", y = gene_name) +
  theme_classic() +
  theme(
    plot.background = element_rect(fill = '#333333'),
    panel.background = element_rect(fill = '#333333'),
    text = element_text( color = 'white', size = 20),
    axis.title = element_text( size = 14, color = 'white'),
    axis.text = element_text( size = 12, color = 'white'),
    axis.ticks = element_line(color = "white"),
    plot.title = element_text( size = 16, hjust = 0.5, color = 'white'),
    panel.grid.major = element_line(color = "gray", size = 0.1),
    panel.grid.minor = element_line(color = "gray", size = 0.1)
```

```

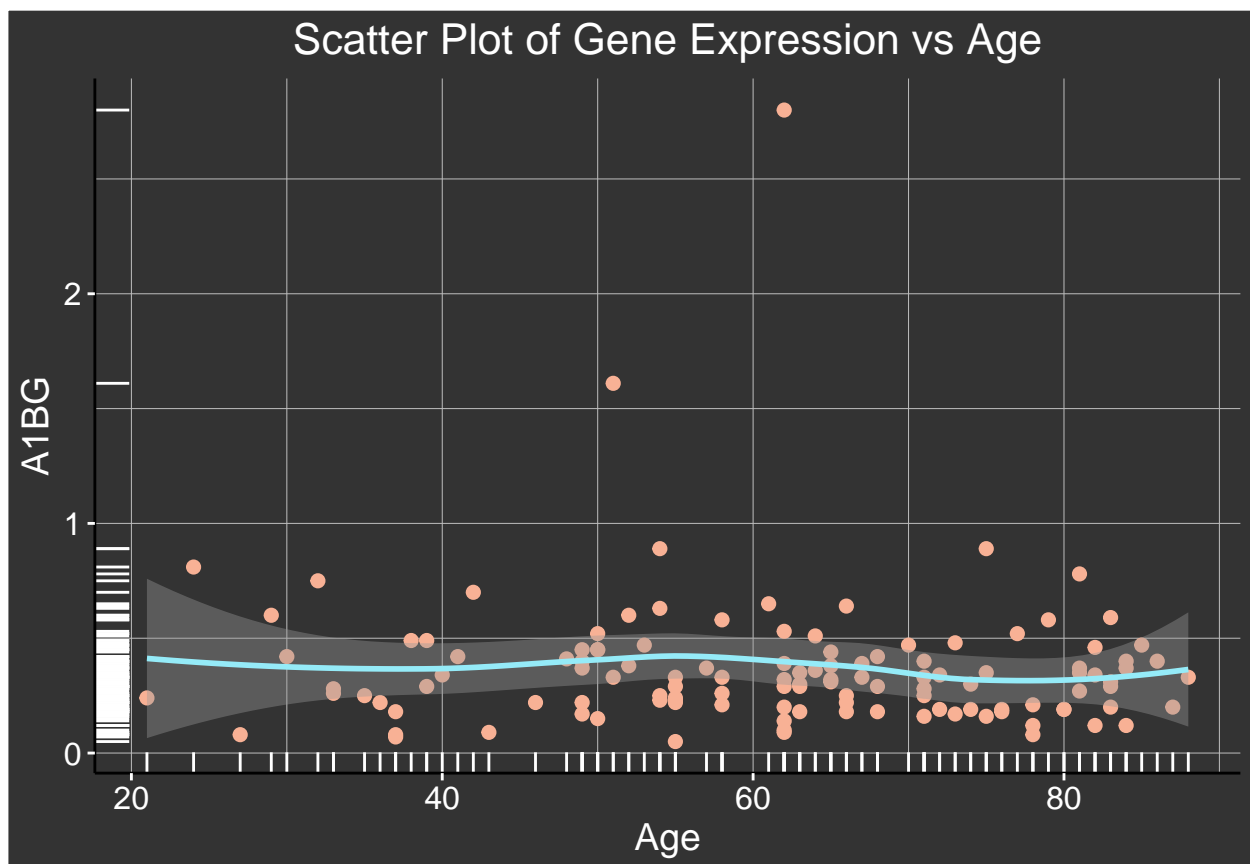
) +
  ggtitle("Scatter Plot of Gene Expression vs Age")

## Warning: The 'size' argument of 'element_line()' is deprecated as of ggplot2 3.4.0.
## i Please use the 'linewidth' argument instead.
## This warning is displayed once every 8 hours.
## Call 'lifecycle::last_lifecycle_warnings()' to see where this warning was
## generated.

# Save the scatter plot
ggsave("plots/scatter_plot.png", plot = scatter_plot, width = 16, height = 9)

# Print the scatter plot
print(scatter_plot)

```



This aesthetic is inspired by <https://waterdata.usgs.gov/blog/ggplot-jazz/>

```

# filter out data points with unknown sex
merged_data <- merged_data %>% filter(sex != "unknown")
# Define a color palette and labels for the legend
palette <- c("disease state: COVID-19" = "#66c2a5", "disease state: non-COVID-19" = "#fc8d62")
labels <- c("disease state: COVID-19" = "COVID-19", "disease state: non-COVID-19" = "non-COVID-19")

# Plot

```

```

boxplot <- ggplot(merged_data, aes_string(x = categorical_covariate1,
                                          y = gene_name,
                                          fill = categorical_covariate2)) +
  geom_boxplot(outlier.color = "red", color = "white") +
  labs(x = "Sex", y = "Gene Expression", fill = "Disease Status") +
  theme_classic() +
  theme(
    plot.background = element_rect(fill = '#333333'),
    panel.background = element_rect(fill = '#333333'),
    text = element_text(color = 'white', size = 14),
    axis.title = element_text(size = 12, color = 'white'),
    axis.text = element_text(size = 12, color = 'white'),
    axis.ticks = element_line(color = "gray"),
    plot.title = element_text(size = 16, color = 'white', hjust=0.5),
    legend.position = 'top',
    legend.background = element_rect(fill = "#333333"),
    legend.text = element_text(color = 'white', size = 12),
    legend.title = element_text(color = 'white', size = 12),
    legend.key = element_rect(fill = "#333333"),
    panel.grid.major = element_line(color = "gray", size = 0.5)
  ) +
  scale_fill_manual(values = palette, labels = labels) +
  ggtitle("Boxplot of Gene Expression by Sex and Disease Status")

```

```

## 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.

```

```

# Save the boxplot
ggsave("plots/boxplot.png", plot = boxplot, width = 16, height = 9)

# Print the boxplot
print(boxplot)

```

