# Mid-Semester Check In

## 2025-07-12

```r
#Always good to get all the packages up and running before doing anything else
library(data.table)
library(ggplot2)
library(tidyverse)
```

```
## ── Attaching core tidyverse packages ───────────────────── tidyverse 2.0.0 ──
## ✔ dplyr      1.1.4      ✔ readr      2.1.5
## ✔ forcats    1.0.0      ✔ stringr    1.5.1
## ✔ lubridate  1.9.4      ✔ tibble     3.2.1
## ✔ purrr      1.0.2      ✔ tidyr      1.3.1
## ── Conflicts ──────────────────────────────────────── tidyverse_conflicts() ──
## ✖ dplyr::between()    masks data.table::between()
## ✖ dplyr::filter()     masks stats::filter()
## ✖ dplyr::first()      masks data.table::first()
## ✖ lubridate::hour()   masks data.table::hour()
## ✖ lubridate::isoweek() masks data.table::isoweek()
## ✖ dplyr::lag()        masks stats::lag()
## ✖ dplyr::last()       masks data.table::last()
## ✖ lubridate::mday()   masks data.table::mday()
## ✖ lubridate::minute() masks data.table::minute()
## ✖ lubridate::month()  masks data.table::month()
## ✖ lubridate::quarter() masks data.table::quarter()
## ✖ lubridate::second() masks data.table::second()
## ✖ purrr::transpose()  masks data.table::transpose()
## ✖ lubridate::wday()   masks data.table::wday()
## ✖ lubridate::week()   masks data.table::week()
## ✖ lubridate::yday()   masks data.table::yday()
## ✖ lubridate::year()   masks data.table::year()
## ℹ Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts
to become errors
```

```r
filename <- "/Users/nupoormarwah/Downloads/QBS103_GSE157103_series_matrix-1.csv"
meta_data <- fread(filename)

filename <- "/Users/nupoormarwah/Downloads/QBS103_GSE157103_genes.csv"

#Makes so that V1 col that contains the gene names are now the row names
gene_expr <- fread(filename) %>%
  column_to_rownames("V1")

#Add a col to metadata table. Use $ to choose the col name. Unlist turns it into a vector
meta_data$AASDHPPT <- unlist(gene_expr["AASDHPPT",])

#Now we should have another col in meta_data with the name "AASDHPPT" and all its associated row values from gene_expr
```

```r
#First layer in ggplot function is data. Next is mapping to tell R where to put the cols
and rows on the plot. "aes" is part of the mapping layer and tells R what to make the x
and y-axis. Since this is a histogram, we only need to assign x-axis
ggplot(data = meta_data, mapping = aes(x = AASDHPPT)) +
  #After mapping, the next step is to add a geom layer to tell R what type of plot you want to make. You can use "fill" to change the color of the entire shape or "color" to change the outline
  geom_histogram(fill = "purple") +
  #"labs" gives labels to x and y-axis and you can include a title
  labs(x = "AASDHPPT Expression", y = "Count", title = "Histogram of AASDHPPT Gene Expression") +
  #We can choose any theme. BW gives black axis and text and white background
  theme_bw() +
  #Use theme again to center title and bold it. "Element text" function says that the element in the plot title is a text element. "hjust" is horizontal adjustment (0 = all the way to the left, 1 = all the way to the right, so center = 0.50)
  theme(
    plot.title = element_text(face = "bold", hjust = 0.50)
  )
```
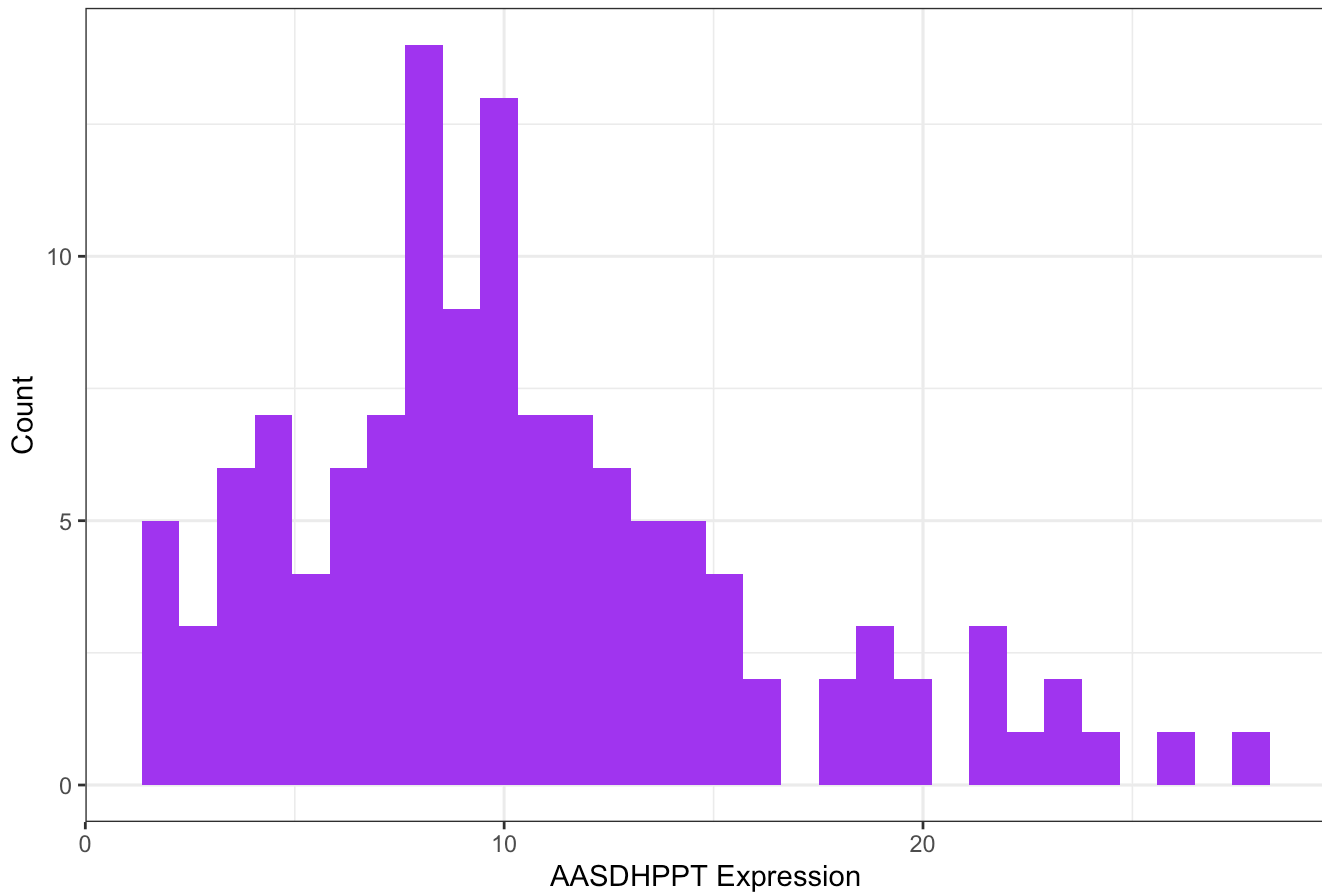
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```
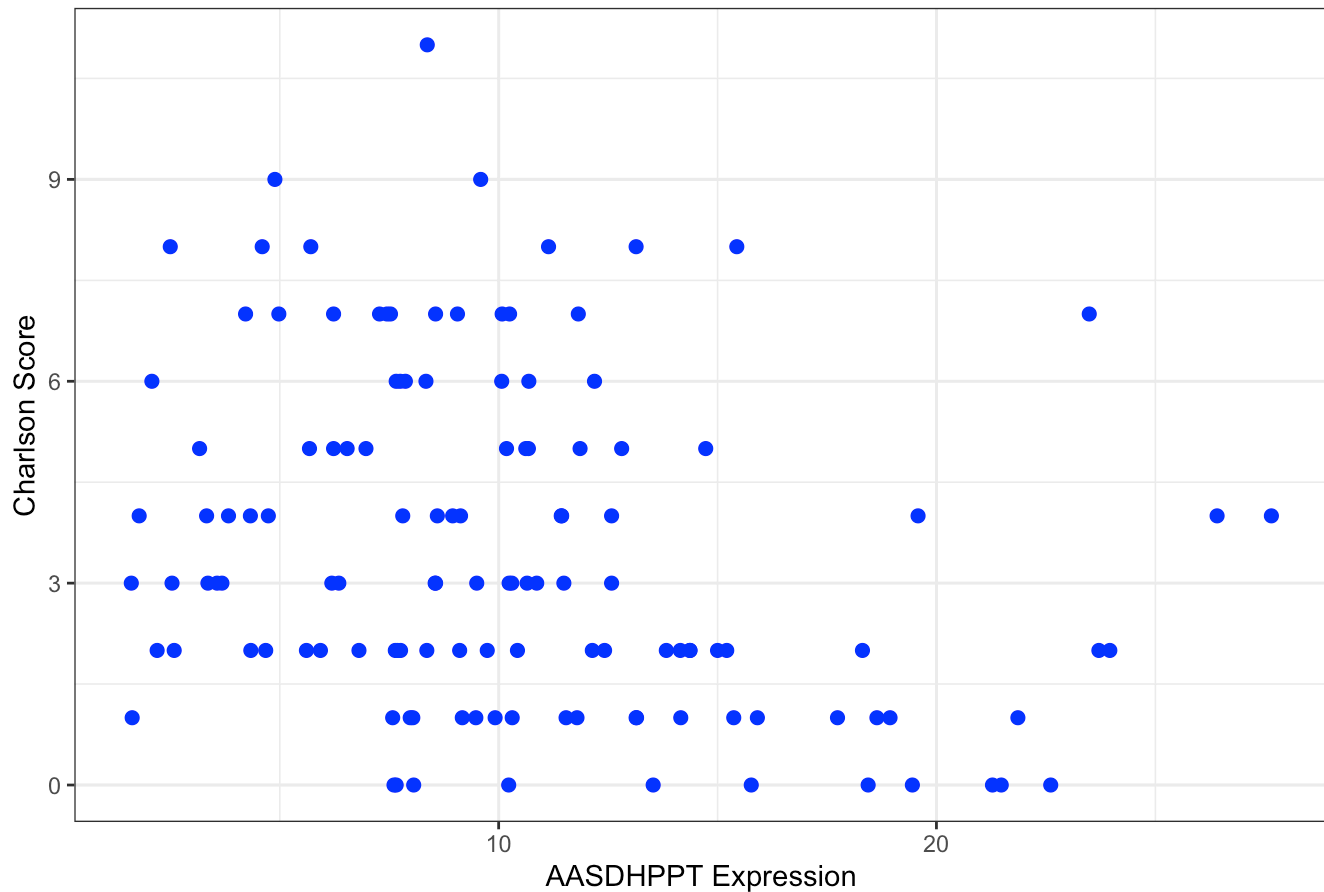
# Histogram of AASDHPPT Gene Expression



```
#First layer in ggplot function is data. Next is mapping to tell R where to put the cols
and rows on the plot. "aes" is part of the mapping layer and tells R what to make the x-
and y-axis. Since this is a scatterplot, we need both x- and y-axis. I chose Charlson Sc
ore as the continuous variable here
ggplot(data = meta_data, mapping = aes(x = AASDHPPT, y = charlson_score)) +
  #After mapping, the next step is to add a geom layer to tell R what type of plot you w
ant to make. You can use "color" to change the color of the dots in the scatterplot
  geom_point(color = "blue", size = 2) +
  #"labs" gives labels to x and y-axis and you can include a title
  labs(x = "AASDHPPT Expression", y = "Charlson Score", title = "Scatterplot of Charlson
Score vs. AASDHPPT Gene Expression") +
  #We can choose any theme. BW gives black axis and text and white background
  theme_bw() +
  #Use theme again to center title and bold it. "Element text" function says that the el
ement in the plot title is a text element. "hjust" is horizontal adjustment (0 = all the
way to the left, 1 = all the way to the right, so center = 0.50)
  theme(
    plot.title = element_text(face = "bold", hjust = 0.50)
  )
```

# Scatterplot of Charlson Score vs. AASDHPPT Gene Expression



```
#First layer in ggplot function is data. Next is mapping to tell R where to put the cols
and rows on the plot. "aes" is part of the mapping layer and tells R what to make the x
and y-axis. Since this is a boxplot, we need both x- and y-axis. I chose sex and disease
status as my categorical variables
ggplot(data = meta_data, mapping = aes(x = sex, y = AASDHPPT, fill = sex)) +
  #After mapping, the next step is to add a geom layer to tell R what type of plot you w
ant to make
  geom_boxplot() +
  #Add a facet. A facet lets us separate based on the values of a variable. Here, it wou
ld be the "disease_status" var
  facet_wrap(~disease_status) +
  #"labs" gives labels to x and y-axis and you can include a title. To change the legen
d, use "fill" since we used that up top
  labs(x = "Sex", y = "AASDHPPT Gene Expression", title = "Boxplot of Gene Expression by
Disease Status and Sex",
       fill = "Sex") +
  #We can choose any theme. BW gives black axis and text and white background
  theme_bw() +
  #Use theme again to center title and bold it. "Element text" function says that the el
ement in the plot title is a text element. "hjust" is horizontal adjustment (0 = all the
way to the left, 1 = all the way to the right, so center = 0.50)
  theme(
    plot.title = element_text(face = "bold", hjust = 0.50)
  )
```

# Boxplot of Gene Expression by Disease Status and Sex