Check In 1

Savannah Wallis

2025-09-27

```
#READING IN

#all necessary 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(ggpubr)
```

```
## Loading required package: ggplot2
```

```
library(ggplot2)
library(gridExtra)
```

```
##
## Attaching package: 'gridExtra'

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

```
library(grid)
library(ComplexHeatmap)
```

```
## ========================================
## ComplexHeatmap version 2.24.1
## Bioconductor page: http://bioconductor.org/packages/ComplexHeatmap/
## Github page: https://github.com/jokergoo/ComplexHeatmap
## Documentation: http://jokergoo.github.io/ComplexHeatmap-reference
```

```
##
## If you use it in published research, please cite either one:
## - Gu, Z. Complex Heatmap Visualization. iMeta 2022.
## - Gu, Z. Complex heatmaps reveal patterns and correlations in multidimensional
##     genomic data. Bioinformatics 2016.
##
##
## The new InteractiveComplexHeatmap package can directly export static
## complex heatmaps into an interactive Shiny app with zero effort. Have a try!
##
## This message can be suppressed by:
##   suppressPackageStartupMessages(library(ComplexHeatmap))
## ========================================
```

```
library(circlize)
```

```
## ========================================
## circlize version 0.4.16
## CRAN page: https://cran.r-project.org/package=circlize
## Github page: https://github.com/jokergoo/circlize
## Documentation: https://jokergoo.github.io/circlize_book/book/
##
## If you use it in published research, please cite:
## Gu, Z. circlize implements and enhances circular visualization
##   in R. Bioinformatics 2014.
##
## This message can be suppressed by:
##   suppressPackageStartupMessages(library(circlize))
## ========================================
```

```
library(ggbeeswarm)
library(ggdist)


#reading in the counts
my_data <- read.csv("C:/Users/savan/Downloads/counts.csv", header = TRUE, row.names = 1, )

#this is just small portion of the data for me to view (head) is too large
my_data[1:12, 1:2]
```

```
##                   TCGA.GM.A2DL.01A.11R.A18M.07 TCGA.AC.A2QI.01A.12R.A19W.07
## ENSG00000000003.15                         1262                         2922
## ENSG00000000005.6                           120                            4
## ENSG00000000419.13                         1535                         1779
## ENSG00000000457.14                          885                         2574
## ENSG00000000460.17                          328                          586
## ENSG00000000938.13                          604                          625
## ENSG00000000971.16                         2737                         3771
## ENSG00000001036.14                         2370                         2517
## ENSG00000001084.13                         1139                         1688
## ENSG00000001167.14                         2543                         2153
## ENSG00000001460.18                          559                          845
## ENSG00000001461.17                         2204                         2653
```

```r
#READING IN

#reading in the metadata
my_metadata <- read.csv("C:/Users/savan/Downloads/meta_data.csv", header = TRUE,)
```

```r
#SUMMARY STATS

#I chose the gene on the 11th line and computed the summary statistics
#I had to make the data a numeric vector since it was a row
my_gene <- as.numeric(my_data[11,])

#summary
summary_mygene <- summary(my_gene)

#standard deviation
sd_mygene <- sd(my_gene)

#adding standard dev and summary into a vector
summary_mygene <- c(summary_mygene, "Standard Deviation" = sd(my_gene))
summary_mygene
```

```
##               Min.            1st Qu.             Median               Mean
##           105.0000           474.5000           652.0000           699.8773
##            3rd Qu.               Max. Standard Deviation
##           866.5000          2620.0000           318.0273
```

```r
#making a data fram of summary + sd
summary_df <- data.frame(summary_mygene)

#labeling the column of values
colnames(summary_df) <- "Value"

#adding my own row name
summary_df <- cbind(`Summary Statistics` = rownames(summary_df), summary_df)

#getting rid of default row names
rownames(summary_df ) <- NULL

#viewing the summary df
summary_df
```

```
##   Summary Statistics     Value
## 1               Min.  105.0000
## 2            1st Qu.  474.5000
## 3             Median  652.0000
## 4               Mean  699.8773
## 5            3rd Qu.  866.5000
## 6               Max. 2620.0000
## 7 Standard Deviation  318.0273
```

```r
#open PNG device
png("dataframe_image.png", width = 800, height = 400)

#converting df to table and loading it
#I used https://cran.r-project.org/web/packages/gridExtra/vignettes/tableGrob.html to figure this out

#Adding a title
grid.text("Summary Statistics for ENSG00000001460.18\n(RNA-Seq Counts)", y = 0.78, gp = gpar(fontsize =

#drawing the tabe
grid.table(summary_df, rows = NULL)

#close device
dev.off()
```

```
## pdf
##   2
```

```r
#SUMMARY STATS

#I chose the gene on the 12th line and computed the summary statistics
#I had to make the data a numeric vector since it was a row
my_gene2 <- as.numeric(my_data[12,])

#summary
summary_mygene2 <- summary(my_gene2)

#sd
sd_mygene2 <- sd(my_gene2)

#vector of all stats
summary_mygene2 <- c(summary_mygene2, "Standard Deviation" = sd(my_gene2))
summary_mygene2
```

```
##               Min.            1st Qu.             Median               Mean
##             232.00            1658.00            2516.00            3079.09
##            3rd Qu.               Max. Standard Deviation
##            3934.00           15355.00            2120.10
```

```r
#df of stats
summary_df2 <- data.frame(summary_mygene2)

#making my own column/row names again
colnames(summary_df2) <- "Value"
summary_df2 <- cbind(`Summary Statistics` = rownames(summary_df2), summary_df2)
rownames(summary_df2 ) <- NULL

#viewing
summary_df2
```

```
##   Summary Statistics    Value
## 1               Min.   232.00
```

```
## 2            1st Qu.  1658.00
## 3             Median  2516.00
## 4               Mean  3079.09
## 5            3rd Qu.  3934.00
## 6               Max. 15355.00
## 7 Standard Deviation  2120.10
```

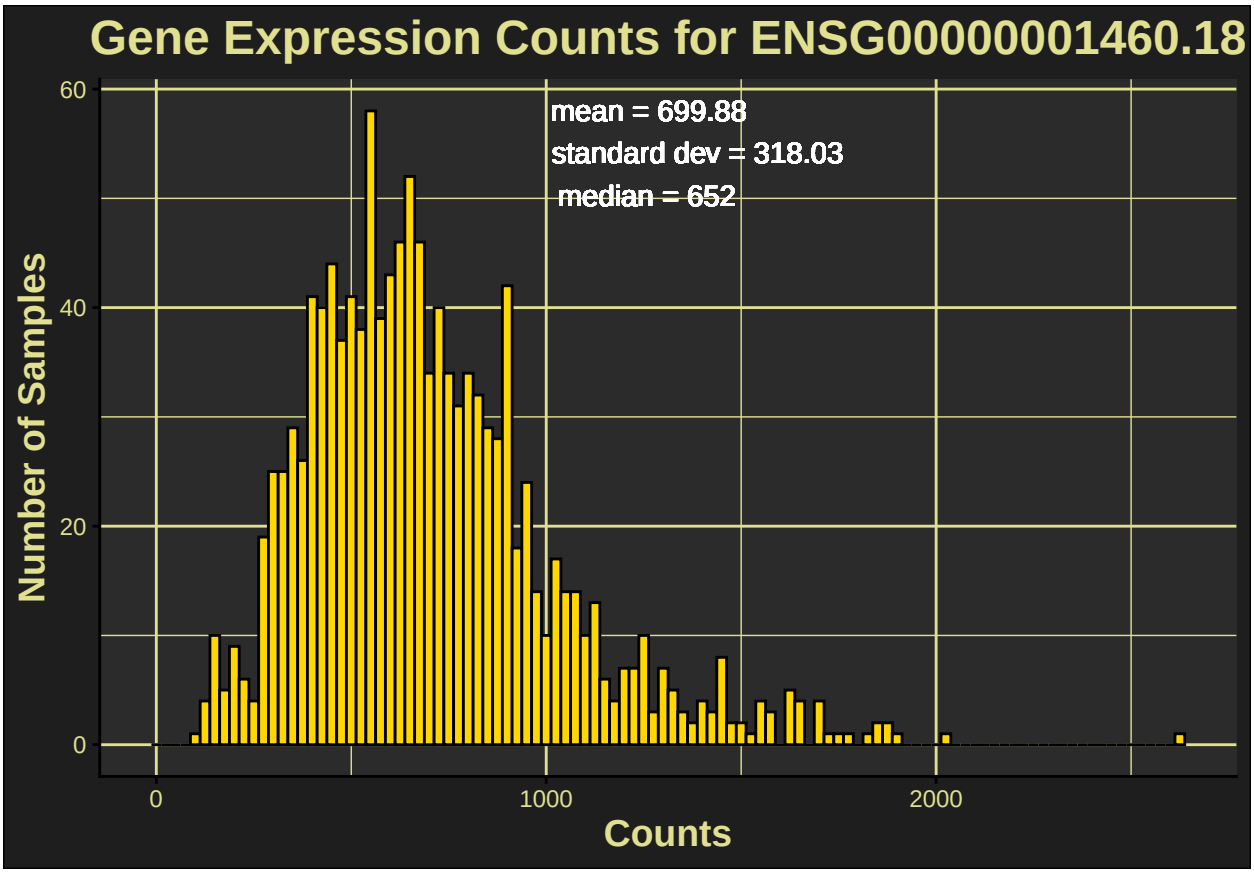
```
#open device
png("dataframe_image2.png", width = 800, height = 400)

#drawing and customizing the table
grid.text("Summary Statistics for ENSG00000001461.17\n(RNA-Seq Counts)", y = 0.78, gp = gpar(fontsize =
grid.table(summary_df2, rows = NULL)

#close device
dev.off()
```

```
## pdf
##   2
```

```
#HISTOGRAM FOR GENE 1

#turned my gene data into a data frame
my_gene_df <- data.frame(my_gene)


#making histogram. I am using various elements to change the color because it is fun!
#My resource for color customization is https://www.sthda.com/english/wiki/ggplot2-themes-and-backgroun
plot<- ggplot(my_gene_df, aes(x = my_gene)) +
  geom_histogram(binwidth = 25, fill = "gold", color = "black") +
  labs(title = "Gene Expression Counts for ENSG00000001460.18",
       x = "Counts",
       y = "Number of Samples") + theme_classic() +
  theme(
   plot.background = element_rect(fill = "#1e1e1e", color = "black"),
  panel.background = element_rect(fill = "#2b2b2b", color =  NA),
  plot.title = element_text(face = "bold", size = 18, hjust = 0.5, color = "#e0e090"),   # gold
  axis.title = element_text(face = "bold", size = 14, color = "#e0e090"),
  axis.text = element_text(color = "#e0e090"),
  panel.grid.major = element_line(color = "#e0e090"),
  panel.grid.minor = element_line(color = "#e0e090"), axis.line.x = element_line(color = "black"))

#taking the mean of the counts
mean1 <-  round(mean(my_gene),2)

#taking the sd of the counts
sd1 <-  round(sd(my_gene), 2)

#taking the median of the counts
median1 <- round(median(my_gene),2)

#I added some important statistical annotations
#looked up and figured out geom_text from https://stackoverflow.com/questions/53799878/how-to-format-gg
```

```
histogram <- plot +
             geom_text(aes(x = 3.6, y = Inf), label = paste0("mean = ", mean1),
                       vjust = 2,hjust = -2, size = 4, color = "white", inherit.aes =
                         FALSE) +
             geom_text(aes(x = 3.6, y = Inf), label = paste0("standard dev = ", sd1),
                       vjust = 4,hjust = -1.35, size = 4,color = "white", inherit.aes =
                         FALSE )  +
             geom_text(aes(x = 3.6, y = Inf), label = paste0("median = ", median1),
                       vjust = 6,hjust = -2.24, size = 4,color = "white", inherit.aes =
                         FALSE)

#printing entire histogram
histogram
```

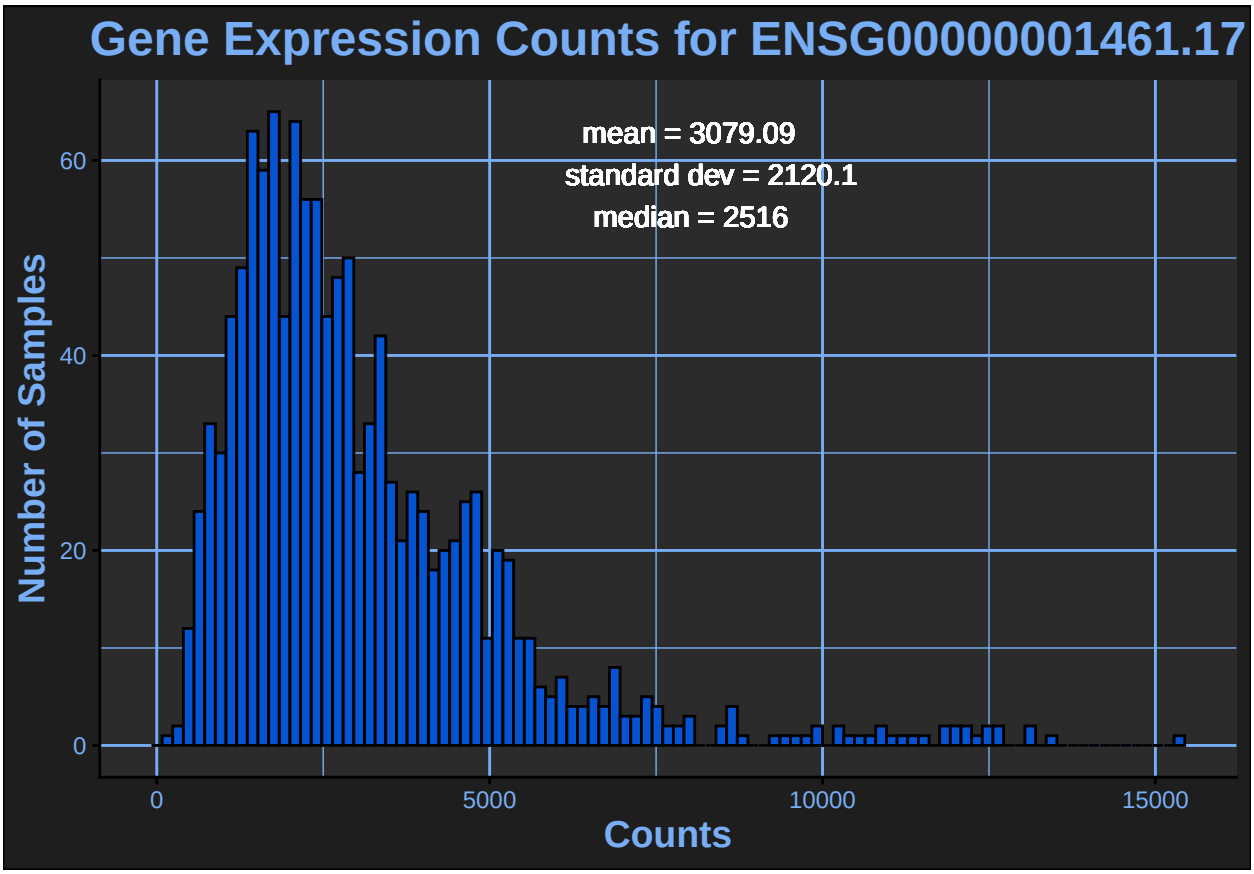
```
## Warning in geom_text(aes(x = 3.6, y = Inf), label = paste0("mean = ", mean1), : All aesthetics have l
## i Please consider using `annotate()` or provide this layer with data containing
##   a single row.

## Warning in geom_text(aes(x = 3.6, y = Inf), label = paste0("standard dev = ", : All aesthetics have l
## i Please consider using `annotate()` or provide this layer with data containing
##   a single row.

## Warning in geom_text(aes(x = 3.6, y = Inf), label = paste0("median = ", : All aesthetics have length
## i Please consider using `annotate()` or provide this layer with data containing
##   a single row.
```



```r
#turned my gene data for second gene into a data frame
my_gene_2 <- as.numeric(my_data[12,])
my_gene_df2 <- data.frame(my_gene2)


#making histogram. I am using various elements to change the color because it is fun!
plot2 <- ggplot(my_gene_df2, aes(x = my_gene2)) +
  geom_histogram(binwidth = 160, fill = "#0453d1", color = "black") +
  labs(title = "Gene Expression Counts for ENSG00000001461.17",
       x = "Counts",
       y = "Number of Samples") + theme_classic() +
  theme(
   plot.background = element_rect(fill = "#1e1e1e", color = "black"),
    panel.background = element_rect(fill = "#2b2b2b", color =  NA),
    plot.title = element_text(face = "bold", size = 18, hjust = 0.5, color = "#78aef5"),
  axis.title = element_text(face = "bold", size = 14, color = "#78aef5"),
  axis.text = element_text(color = "#78aef5"),
  panel.grid.major = element_line(color = "#78aef5"),
  panel.grid.minor = element_line(color = "#78aef5"),
  axis.line.x = element_line(color = "black"))

#taking the mean of the counts
mean2 <-  round(mean(my_gene2),2)

#taking the sd of the counts
sd2 <-  round(sd(my_gene2), 2)

#taking the median of the counts
median2 <- round(median(my_gene2),2)

#I added some important statistical annotations
histogram2 <- plot2 +
            geom_text(aes(x = 3.6, y = Inf), label = paste0("mean = ", mean2),
                      vjust = 3,hjust = -2, size = 4, color = "white", inherit.aes =
                        FALSE) +
            geom_text(aes(x = 3.6, y = Inf), label = paste0("standard dev = ", sd2),
                      vjust = 5,hjust = -1.4, size = 4,color = "white", inherit.aes =
                        FALSE )  +
            geom_text(aes(x = 3.6, y = Inf), label = paste0("median = ", median2),
                      vjust = 7,hjust = -2.24, size = 4,color = "white", inherit.aes =
                        FALSE)

#printing entire histogram
histogram2
```

```
## Warning in geom_text(aes(x = 3.6, y = Inf), label = paste0("mean = ", mean2), : All aesthetics have l
## i Please consider using `annotate()` or provide this layer with data containing
##   a single row.

## Warning in geom_text(aes(x = 3.6, y = Inf), label = paste0("standard dev = ", : All aesthetics have l
## i Please consider using `annotate()` or provide this layer with data containing
##   a single row.

## Warning in geom_text(aes(x = 3.6, y = Inf), label = paste0("median = ", : All aesthetics have length
```

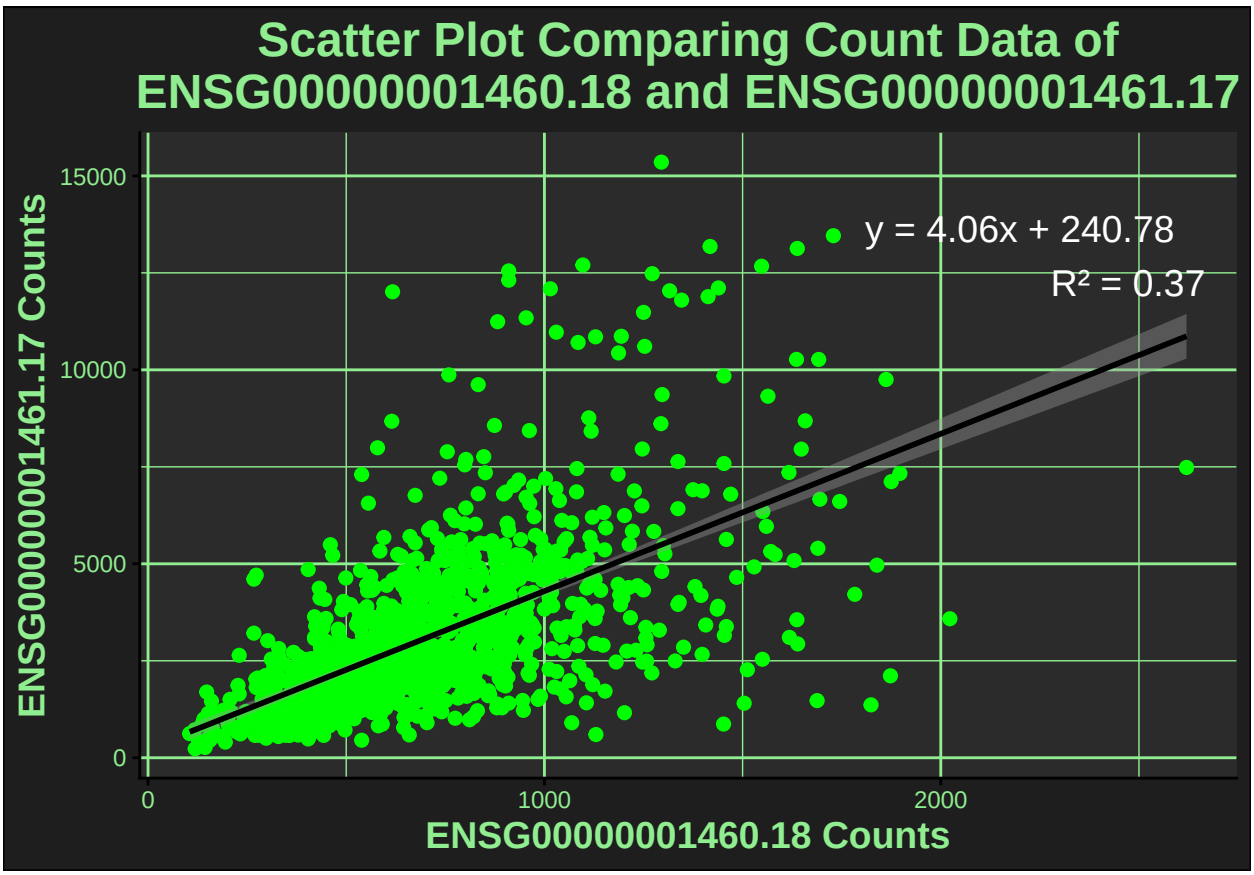
```
## i Please consider using ‘annotate()‘ or provide this layer with data containing
##   a single row.
```




```
#SCATTER PLOT

#Taking second gene and making it a numeric vector
my_gene_2 <- as.numeric(my_data[12,])

#making a df with both genes and ensuring the columns line up
scatter_df <- data.frame(my_gene, my_gene_2)

#beginning to add regresion line
model <- lm(my_gene_2 ~ my_gene, data = scatter_df)

#finding the R squared value. I used https://stackoverflow.com/questions/23519224/extract-r-square-valu
r_squared <- summary(model)$r.squared
r_label <- paste0("R² = ", round(r_squared, 2))

#extract coefficients
coefficients <- coef(model)
intercept <- coefficients[1]
slope <- coefficients[2]

#create equation label text
eq_label <- paste0("y = ",
                   round(slope, 2), "x + ",
```

```
                    round(intercept, 2))

#making and designing the scatter plot
ggplot(scatter_df, aes(x = my_gene, y = my_gene_2))+
  geom_point(color = "green", size = 2) +  geom_smooth(method = "lm", se = TRUE, color = "black") +
  labs(x = "ENSG00000001460.18 Counts", y = "ENSG00000001461.17 Counts", title = "Scatter Plot Comparing
           hjust = 1.2, vjust = -20, size = 5, color = "white")+ annotate("text", x = Inf, y = -Inf, lab
           hjust = 1.2, vjust = -18, size = 5, color = "white") +  theme_classic() + theme(
   plot.background = element_rect(fill = "#1e1e1e", color = "black"),
  panel.background = element_rect(fill = "#2b2b2b", color =  NA),
  plot.title = element_text(face = "bold", size = 18, hjust = 0.5, color = "lightgreen"),
  axis.title = element_text(face = "bold", size = 14, color = "lightgreen"),
  axis.text = element_text(color = "lightgreen"),
  panel.grid.major = element_line(color = "lightgreen"),
  panel.grid.minor = element_line(color = "lightgreen"), axis.line.x = element_line(color = "black"))
```

```
## `geom_smooth()` using formula = 'y ~ x'
```




```
#COMBINING COUNT DATA AND METADATA

#In order to ensure that the covariate and counts lined up in columns I decided to transpose my count d
counts_transposed <- as.data.frame(t(my_data))

#changing the periods for dashes for easy matchup
```

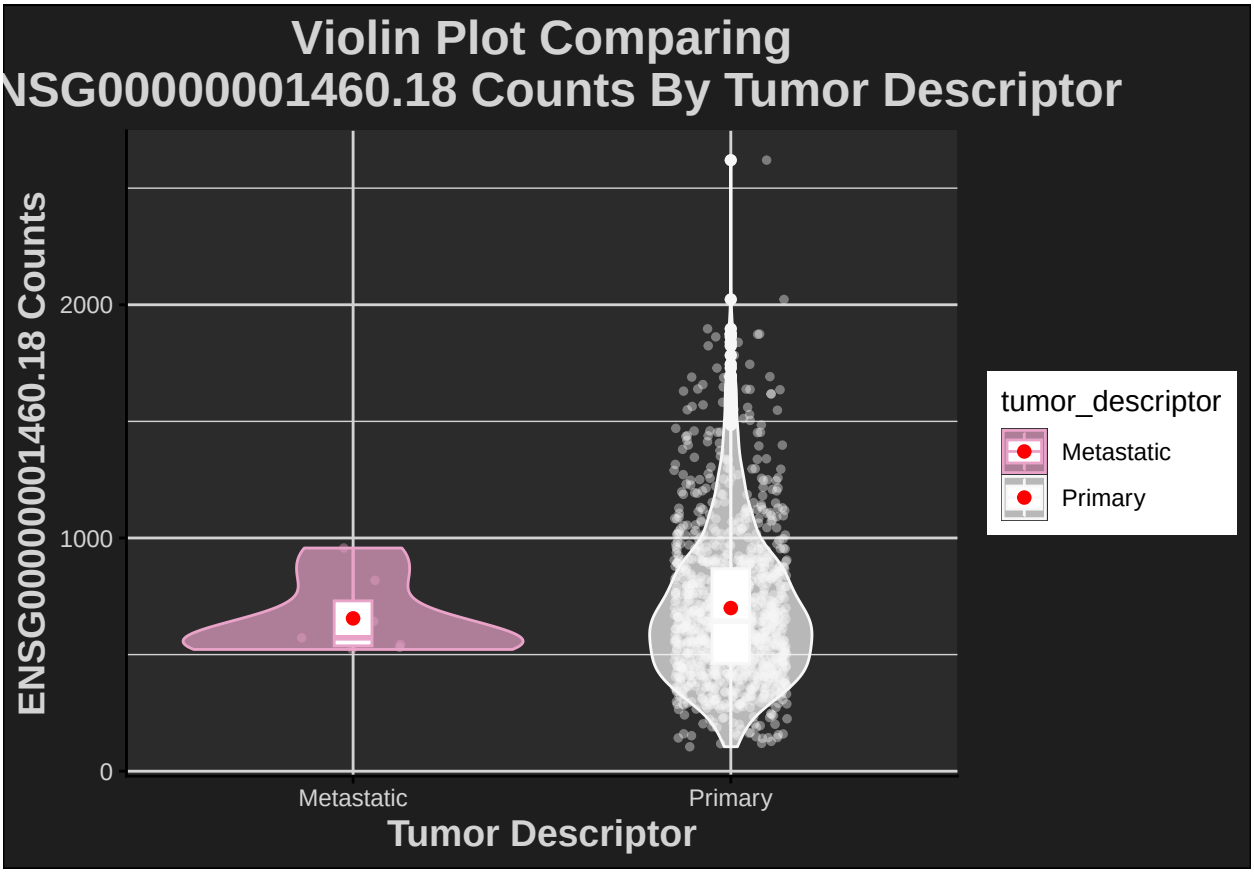
```r
rownames(counts_transposed) <- gsub("\\.", "-", rownames(counts_transposed))

#giving the samples a header for easy merging.
counts_transposed$barcode <- rownames(counts_transposed)

#merging both by the sample ID (barcode) in case they are in differet orders
main_df <- merge(counts_transposed, my_metadata, by = "barcode")

#checking that meta_data is at the end of my main data frame
main_df[1:12, 60740:60758]
```

```
##    paper_days_to_death paper_days_to_last_followup
## 1                   NA                        4047
## 2                   NA                        4005
## 3                   NA                        1474
## 4                   NA                        1448
## 5                   NA                         348
## 6                   NA                        1477
## 7                   NA                        1471
## 8                   NA                         303
## 9                   NA                         259
## 10                  NA                         437
## 11                  NA                        1321
## 12                  NA                        1463
##    paper_age_at_initial_pathologic_diagnosis paper_pathologic_stage
## 1                                         55                   <NA>
## 2                                         50               Stage_II
## 3                                         62               Stage_II
## 4                                         52                Stage_I
## 5                                         50              Stage_III
## 6                                         42               Stage_II
## 7                                         63               Stage_IV
## 8                                         52               Stage_II
## 9                                         70                Stage_I
## 10                                        59               Stage_II
## 11                                        56                Stage_I
## 12                                        54               Stage_II
##    paper_Tumor_Grade paper_BRCA_Pathology paper_BRCA_Subtype_PAM50
## 1                 NA                 <NA>                     LumA
## 2                 NA                 <NA>                     Her2
## 3                 NA                 <NA>                     LumB
## 4                 NA                 <NA>                     LumA
## 5                 NA                 <NA>                     LumA
## 6                 NA                 <NA>                     LumA
## 7                 NA                 <NA>                     LumA
## 8                 NA                 <NA>                     LumB
## 9                 NA                Other                   Normal
## 10                NA                  IDC                     LumA
## 11                NA                  ILC                     LumA
## 12                NA                  IDC                     LumA
##    paper_MSI_status paper_HPV_Status paper_tobacco_smoking_history
## 1                NA               NA                           NA
## 2                NA               NA                           NA
```

```
## 3                NA                NA                           NA
## 4                NA                NA                           NA
## 5                NA                NA                           NA
## 6                NA                NA                           NA
## 7                NA                NA                           NA
## 8                NA                NA                           NA
## 9                NA                NA                           NA
## 10               NA                NA                           NA
## 11               NA                NA                           NA
## 12               NA                NA                           NA
##    paper_CNV.Clusters paper_Mutation.Clusters paper_DNA.Methylation.Clusters
## 1                  C6                    C7                             C1
## 2                  C6                    C9                             C2
## 3                  C6                    C4                             C2
## 4                  C1                    C5                             C2
## 5                  C6                    C4                             C1
## 6                  C1                    C9                             C1
## 7                <NA>                    C9                             C1
## 8                  C6                    C6                             C2
## 9                  C2                    C4                             C3
## 10                 C4                    C4                             C1
## 11                 C1                    C4                             C1
## 12                 C5                    C4                             C1
##    paper_mRNA.Clusters paper_miRNA.Clusters paper_lncRNA.Clusters
## 1                   C1                   C3                  <NA>
## 2                   C2                   C3                  <NA>
## 3                   C2                   C2                  <NA>
## 4                   C2                   C2                  <NA>
## 5                   C2                   C2                  <NA>
## 6                   C2                   C2                  <NA>
## 7                   C2                 <NA>                  <NA>
## 8                   C2                   C3                  <NA>
## 9                   C4                   C4                  <NA>
## 10                  C1                   C3                  <NA>
## 11                  C1                   C3                  <NA>
## 12                  C1                   C3                    C2
##    paper_Protein.Clusters paper_PARADIGM.Clusters paper_Pan.Gyn.Clusters
## 1                    <NA>                     C5                   <NA>
## 2                      C2                     C4                     C4
## 3                    <NA>                     C4                   <NA>
## 4                      C2                     C6                     C4
## 5                      C2                     C6                     C1
## 6                    <NA>                     C6                   <NA>
## 7                      C2                   <NA>                     C1
## 8                      C2                     C8                     C5
## 9                    <NA>                     C6                   <NA>
## 10                   <NA>                     C6                   <NA>
## 11                   <NA>                     C5                   <NA>
## 12                     C1                     C6                     C1
```

```
#VIOLIN PLOT

#Taking out non applicable values because I just want to focus on primary and metastatic
main_df <- main_df[main_df$tumor_descriptor != "Not Applicable", ]
```

```r
#putting together the violin plot for my gene and tumor descriptor
ggplot(main_df, aes(x = tumor_descriptor, y = ENSG00000001460.18, fill = tumor_descriptor, color = tumo
  geom_violin(alpha =0.7) + geom_jitter(width = 0.15, size = 1, alpha = 0.4) + geom_boxplot(width=0.1,
    stat_summary(fun = mean, geom = "point", shape = 20, size = 3, color = "red") +
  labs(x = "Tumor Descriptor", y = "ENSG00000001460.18 Counts", title = "Violin Plot Comparing\nENSG000
   plot.background = element_rect(fill = "#1e1e1e", color = "black"),
  panel.background = element_rect(fill = "#2b2b2b", color =  NA),
  plot.title = element_text(face = "bold", size = 18, hjust = 0.5, color = "lightgrey"),
  axis.title = element_text(face = "bold", size = 14, color = "lightgrey"),
  axis.text = element_text(color = "lightgrey"),
  panel.grid.major = element_line(color = "lightgrey"),
  panel.grid.minor = element_line(color = "lightgrey"), axis.line.x = element_line(color = "black")) +
```



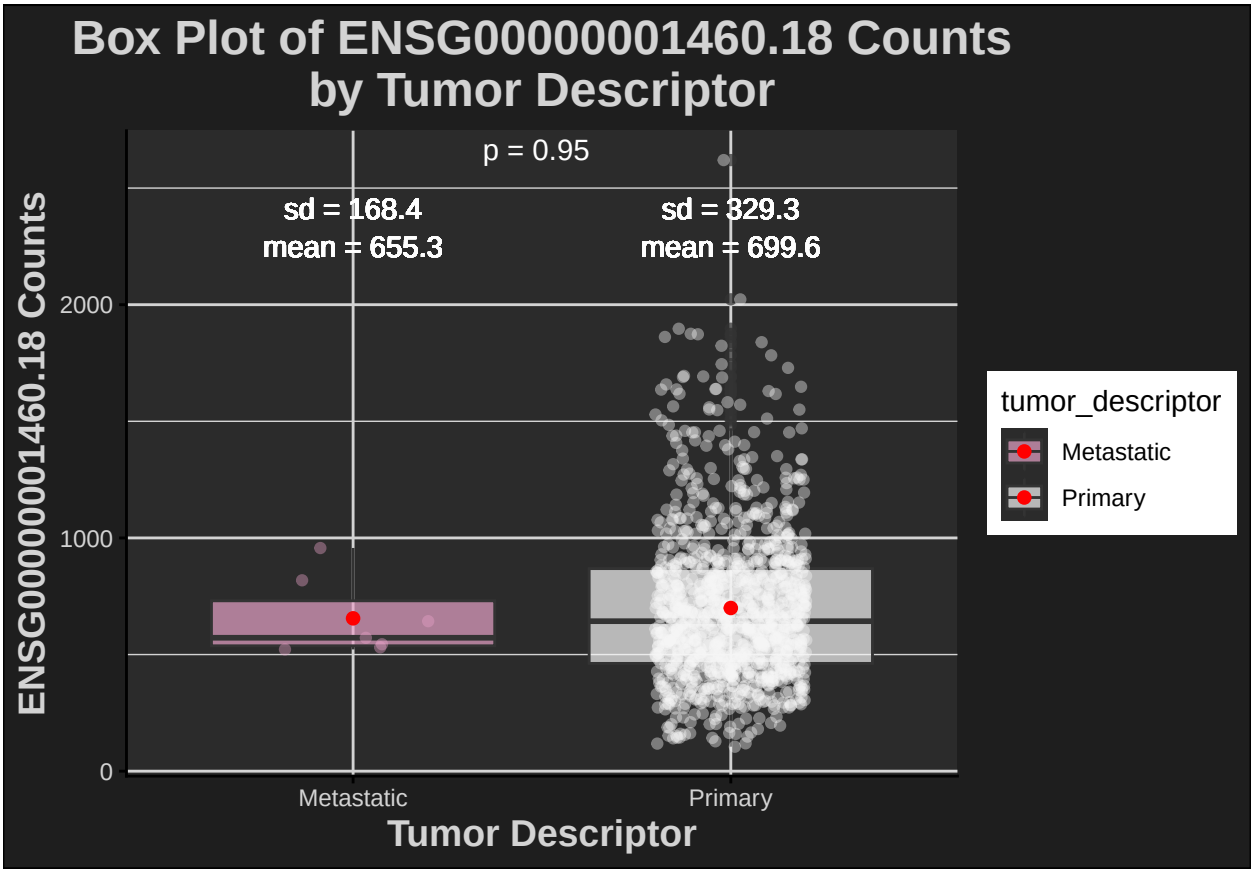

```r
#BOX PLOT

#finding mean of primary
mean_prim <- mean(main_df$ENSG00000001460.18[main_df$tumor_descriptor == "Primary"], na.rm = TRUE)

#finding mean of metastatic
mean_met <- mean(main_df$ENSG00000001460.18[main_df$tumor_descriptor == "Metastatic"], na.rm = TRUE)

#sd of primary
sd_prim <- sd(main_df$ENSG00000001460.18[main_df$tumor_descriptor == "Primary"], na.rm = TRUE)

#sd of metastatic
```

```r
sd_met <- sd(main_df$ENSG00000001460.18[main_df$tumor_descriptor == "Metastatic"], na.rm = TRUE)


#box plot of the same violin plot above
  p1 <- ggplot(main_df, aes(x = tumor_descriptor, y = ENSG00000001460.18, fill = tumor_descriptor)) +
    geom_boxplot(alpha = 0.7) + geom_jitter(aes(color = tumor_descriptor), width = 0.2, alpha = 0.4, si

#adding mean point here
    stat_summary(fun = mean, geom = "point", shape = 20, size = 3, color = "red") +

    labs(title = paste("Box Plot of ENSG00000001460.18 Counts\nby Tumor Descriptor"),
         x = "Tumor Descriptor",
         y = "ENSG00000001460.18 Counts") + theme_classic() + theme(
   plot.background = element_rect(fill = "#1e1e1e", color = "black"),
  panel.background = element_rect(fill = "#2b2b2b", color =  NA),
  plot.title = element_text(face = "bold", size = 18, hjust = 0.5, color = "lightgrey"),
  axis.title = element_text(face = "bold", size = 14, color = "lightgrey"),
  axis.text = element_text(color = "lightgrey"),
  panel.grid.major = element_line(color = "lightgrey"),
  panel.grid.minor = element_line(color = "lightgrey"), axis.line.x = element_line(color = "black")) + s

#adding the p value to the plot
p2 <- p1 + stat_compare_means(method = "wilcox", label = "p.format",color = "white", label.x = 1.4, size

#looked up and figured out geom_text from https://stackoverflow.com/questions/53799878/how-to-format-gg
p3 <- p2 +
   #metastatic labels
  geom_text(aes(x = 1, y = Inf),
            label = paste0("mean = ", round(mean_met, 1)),
            vjust = 6, hjust = 0.5, size = 4, color = "white", inherit.aes = FALSE) +
  geom_text(aes(x = 1, y = Inf),
            label = paste0("sd = ", round(sd_met, 1)),
            vjust = 4.2, hjust = 0.5, size = 4, color = "white", inherit.aes = FALSE) +

  #primary labels
  geom_text(aes(x = 2, y = Inf),
            label = paste0("mean = ", round(mean_prim, 1)),
            vjust = 6, hjust = 0.5, size = 4, color = "white", inherit.aes = FALSE) +
  geom_text(aes(x = 2, y = Inf),
            label = paste0("sd = ", round(sd_prim, 1)),
            vjust = 4.2, hjust = 0.5, size = 4, color = "white", inherit.aes = FALSE)

#final boxplot
p3
```

```
## Warning in geom_text(aes(x = 1, y = Inf), label = paste0("mean = ", round(mean_met, : All aesthetics
## i Please consider using `annotate()` or provide this layer with data containing
##   a single row.

## Warning in geom_text(aes(x = 1, y = Inf), label = paste0("sd = ", round(sd_met, : All aesthetics have
## i Please consider using `annotate()` or provide this layer with data containing
##   a single row.

## Warning in geom_text(aes(x = 2, y = Inf), label = paste0("mean = ", round(mean_prim, : All aesthetics
```

```
## i Please consider using ‘annotate()‘ or provide this layer with data containing
##   a single row.
```

```
## Warning in geom_text(aes(x = 2, y = Inf), label = paste0("sd = ", round(sd_prim, : All aesthetics hav
## i Please consider using ‘annotate()‘ or provide this layer with data containing
##   a single row.
```




```
#heatmap libraries are at the top

#I found this package and the functions within it to help me resize my heat map via https://stackoverfl

#selecting 10 genes
genes_to_plot <- colnames(main_df)[11:20]
head(genes_to_plot)
```

```
## [1] "ENSG00000001167.14" "ENSG00000001460.18" "ENSG00000001461.17"
## [4] "ENSG00000001497.18" "ENSG00000001561.7"  "ENSG00000001617.12"
```

```
#subsetting those genes
counts_subset <- main_df[,genes_to_plot]

#making sure to transpose them back into rows for the heat map
counts_subset <- t(counts_subset)
```

```r
#adding back samples
colnames(counts_subset) <- main_df$barcode


#heat map annotaitons
column_ha <- HeatmapAnnotation(
  tumor_descriptor = main_df$tumor_descriptor,
  col = list(
    tumor_descriptor = c(Primary = "lightgreen", Metastatic = "darkred", `Not Applicable` = "darkgrey"))
  )

#create heat map
hm <-Heatmap(
  counts_subset,
  name = "Counts",
  top_annotation = column_ha,
  show_row_names = TRUE,
  show_column_names = TRUE,
  column_names_gp = grid::gpar(fontsize= 3),
  column_names_centered = TRUE,
  cluster_columns = TRUE,
  cluster_rows = TRUE,
  width = unit(120, "cm"), #changing the width since its so many samples
  column_title = "Samples",
  row_title = "Genes",
  heatmap_legend_param = list(title = "Count")
)
```

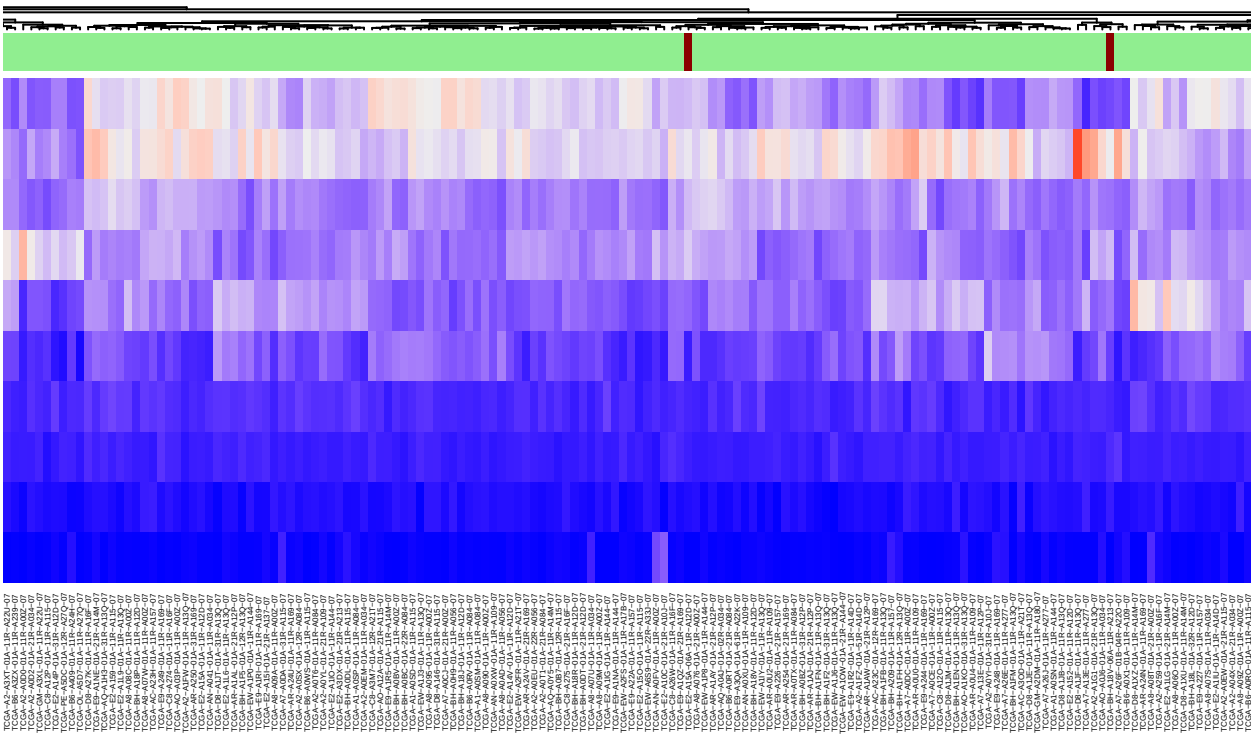
```
## The automatically generated colors map from the 1^st and 99^th of the
## values in the matrix. There are outliers in the matrix whose patterns
## might be hidden by this color mapping. You can manually set the color
## to ‘col‘ argument.
##
## Use ‘suppressMessages()‘ to turn off this message.
```

```r
#creating a heat map without samples for smaller visual
hm_no_samples <- Heatmap(
  counts_subset,
  name = "Counts",
  top_annotation = column_ha,
  show_row_names = TRUE,
  show_column_names = FALSE,
  cluster_columns = TRUE,
  cluster_rows = TRUE,
  width = unit(6, "cm"),
  column_title = "Samples",
  row_title = "Genes",
  heatmap_legend_param = list(title = "Count"))
```

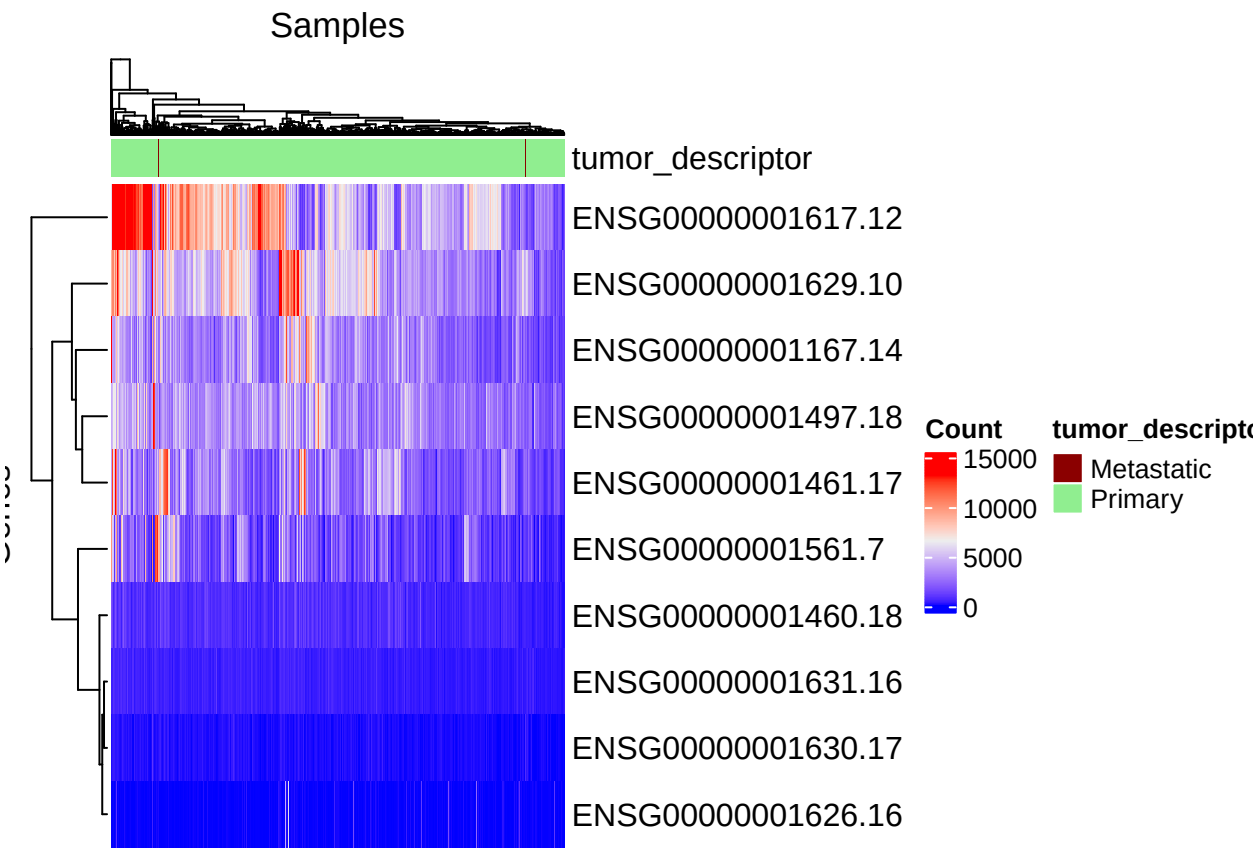
```
## The automatically generated colors map from the 1^st and 99^th of the
## values in the matrix. There are outliers in the matrix whose patterns
## might be hidden by this color mapping. You can manually set the color
## to ‘col‘ argument.
```

```
##
## Use 'suppressMessages()' to turn off this message.
```

```
#PLEASE READ: MY HEATMAP IS SO LARGE YOU HAVE TO DOWNLOAD AS PNG TO SEE IT. PLEASE SEE ATTACHED IN ASSI
hm
```




```
hm_no_samples
```




```
png("myheatmap-final.png", width = 17000, height = 2100, res = 300)
hm
dev.off()
```

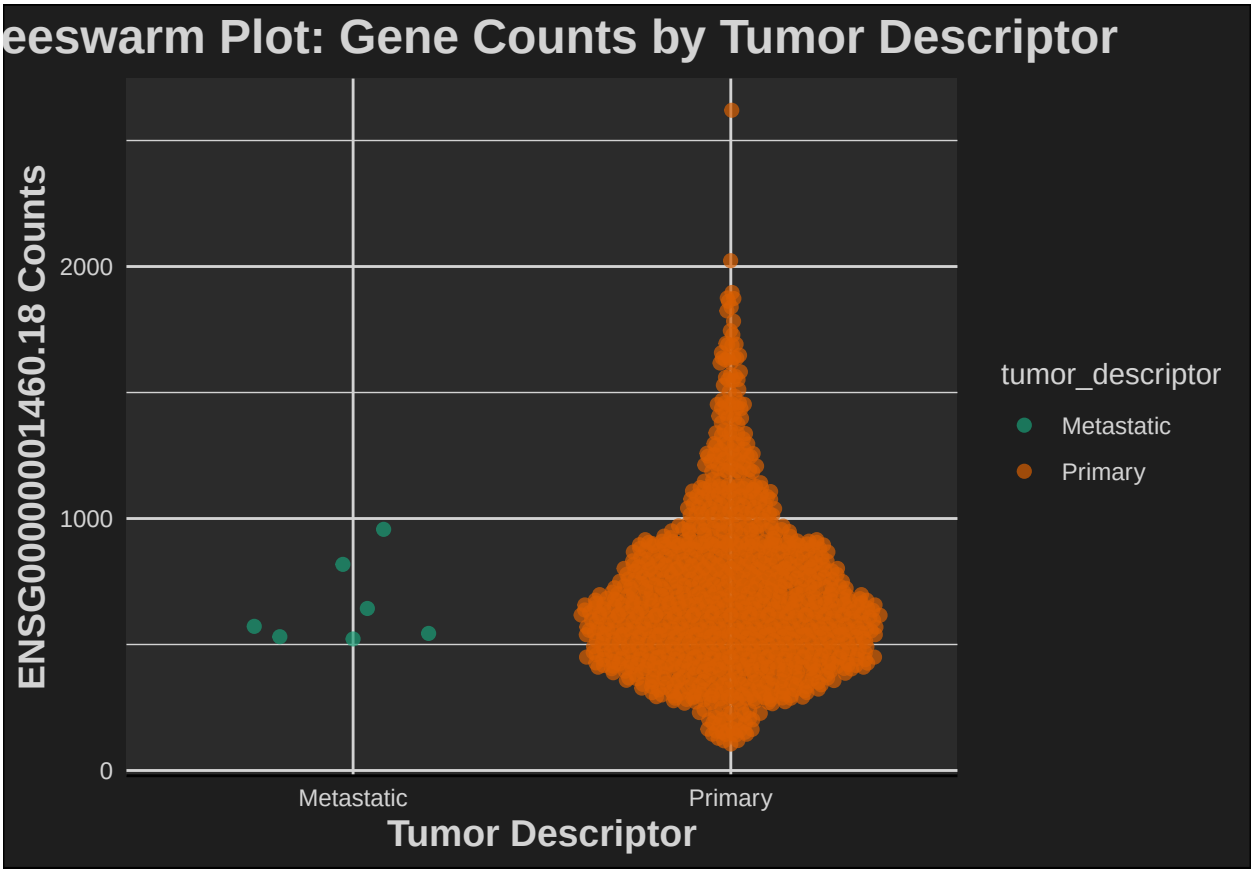
```
## pdf
##   2
```

```
png("myheatmap-final_no_colnames.png")
hm_no_samples
dev.off()
```

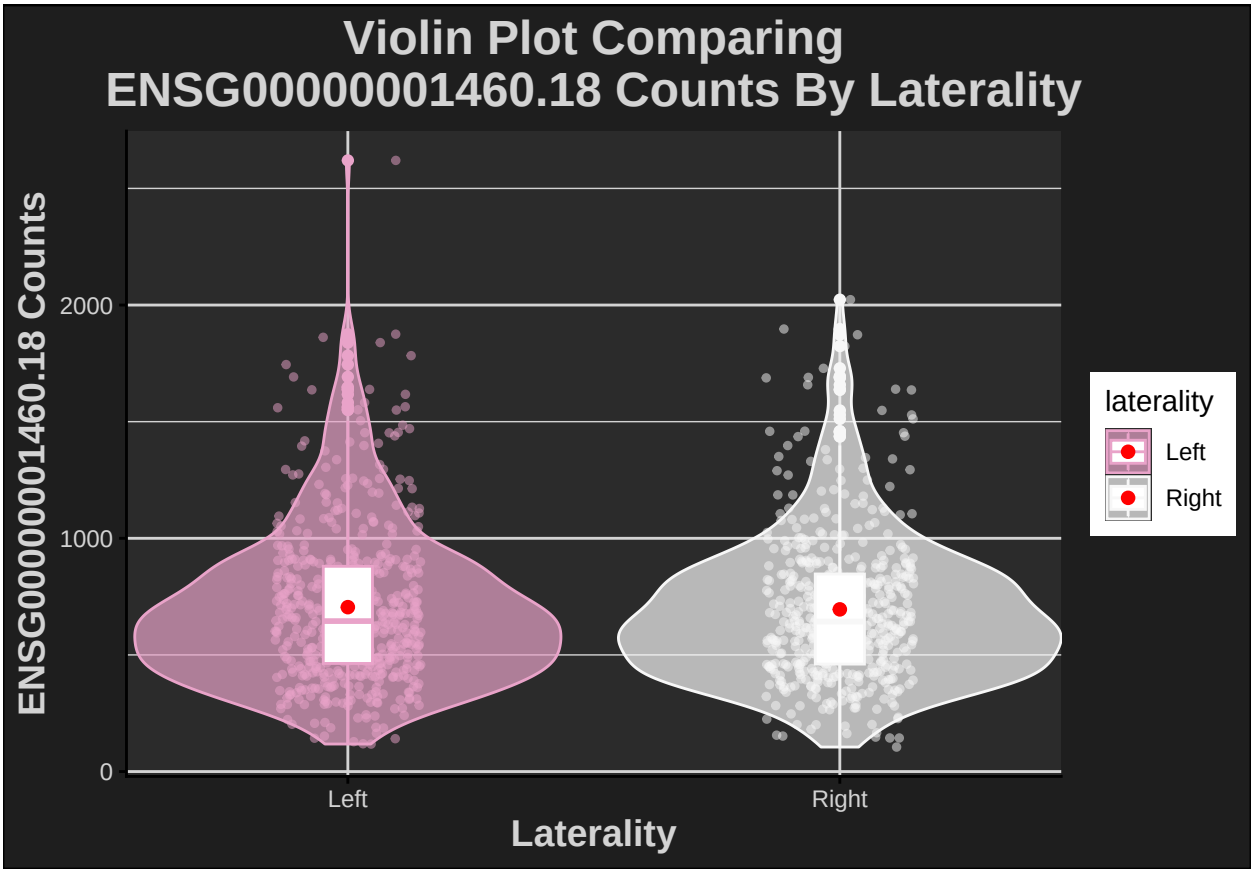
```
## pdf
##   2
```

```
#BEESWARM PLOT

#loading package (ggbeeswarm) at top

#making a beeswarm plot of the same gene and tumor description to see density
ggplot(main_df, aes(x = tumor_descriptor, y = ENSG00000001460.18, color = tumor_descriptor )) +
  geom_quasirandom(dodge.width = 0.75, size = 2, alpha = 0.7) +  theme_minimal() +
  theme(
   plot.background = element_rect(fill = "#1e1e1e", color = "black"),
  panel.background = element_rect(fill = "#2b2b2b", color =  NA),
```

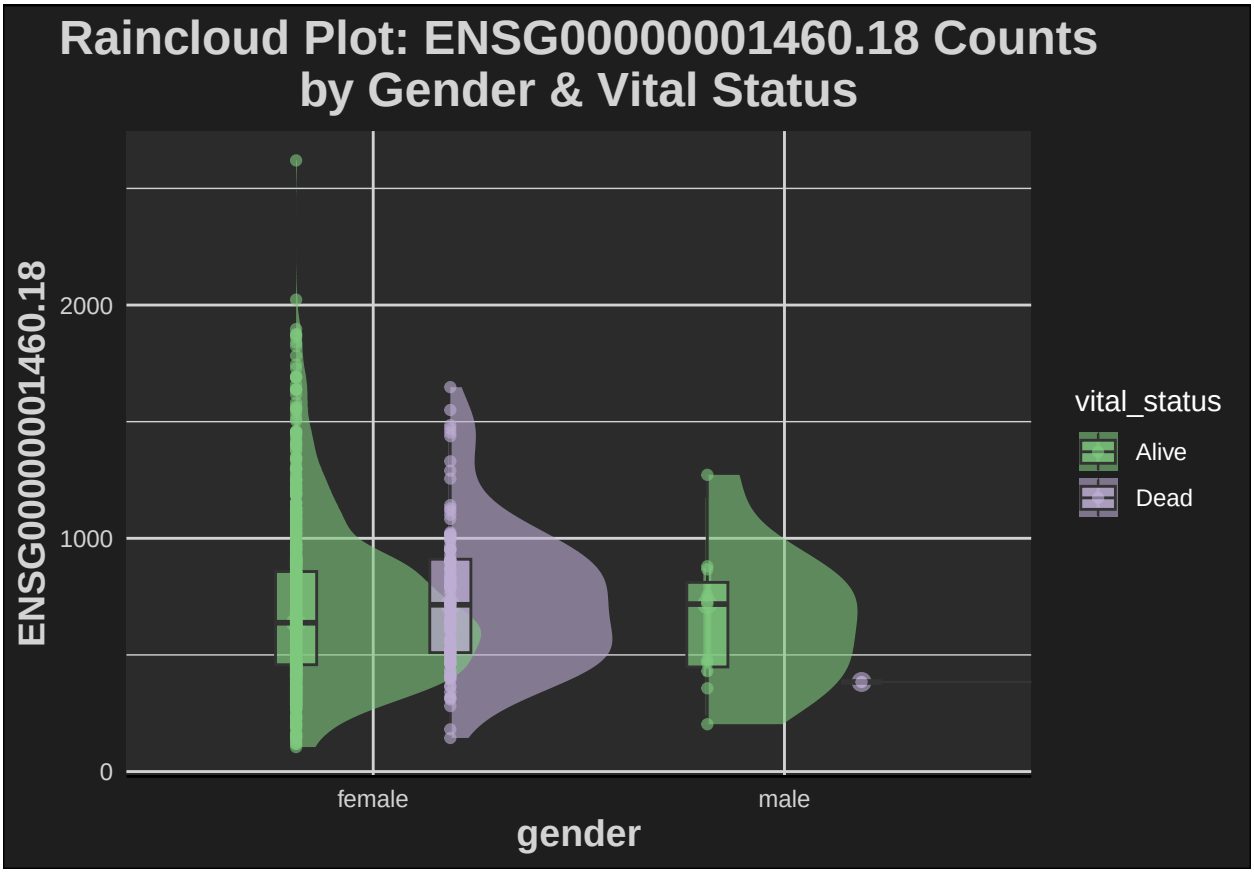
```
  plot.title = element_text(face = "bold", size = 18, hjust = 0.5, color = "lightgrey"),legend.text = el
  axis.title = element_text(face = "bold", size = 14, color = "lightgrey"),
  axis.text = element_text(color = "lightgrey"),
  panel.grid.major = element_line(color = "lightgrey"),
  panel.grid.minor = element_line(color = "lightgrey"), axis.line.x = element_line(color = "black")) +
  labs(x = "Tumor Descriptor", y = "ENSG00000001460.18 Counts", title = "Beeswarm Plot: Gene Counts by T
  scale_color_brewer(palette = "Dark2")
```




```
#VIOLIN PLOT DIFFERENT COVARIATE

#getting rid of NAs
main_df <- main_df[(is.na(main_df$laterality) == FALSE),]

#putting together the violin plot of laterality for my gene
ggplot(main_df, aes(x = laterality, y = ENSG00000001460.18 , fill = laterality, color = laterality))+
  geom_violin(alpha =0.7, position = position_dodge(width = 2)) + geom_jitter(width = 0.15, size = 1, al
    stat_summary(fun = mean, geom = "point", shape = 20, size = 3, color = "red") +
  labs(x = "Laterality", y = "ENSG00000001460.18 Counts", title = "Violin Plot Comparing\nENSG0000000140
   plot.background = element_rect(fill = "#1e1e1e", color = "black"),
  panel.background = element_rect(fill = "#2b2b2b", color =  NA),
  plot.title = element_text(face = "bold", size = 18, hjust = 0.5, color = "lightgrey"),
  axis.title = element_text(face = "bold", size = 14, color = "lightgrey"),
  axis.text = element_text(color = "lightgrey"),
  panel.grid.major = element_line(color = "lightgrey"),
  panel.grid.minor = element_line(color = "lightgrey"), axis.line.x = element_line(color = "black")) + s
```




```
#RAINDROP PLOT

#loading library library(ggdist) at the top

#getting rid of NAs
main_df <- main_df[(is.na(main_df$vital_status) == FALSE),]
main_df <- main_df[(is.na(main_df$gender) == FALSE),]

#creating a half eye/rain cloud plot
ggplot(main_df, aes(x = gender, y = ENSG00000001460.18, fill = vital_status)) +
  stat_halfeye(aes(color = vital_status), position = position_dodge(width = 0.75), alpha = 0.6) +
  geom_boxplot(width = 0.2, position = position_dodge(width = 0.75), outlier.shape = NA, alpha = 0.7) +
  geom_jitter(aes(color = vital_status), position = position_dodge(width = 0.75), size = 1.5, alpha = 0
  theme_minimal() + theme(
   plot.background = element_rect(fill = "#1e1e1e", color = "black"),
  panel.background = element_rect(fill = "#2b2b2b", color =  NA),
  plot.title = element_text(face = "bold", size = 18, hjust = 0.5, color = "lightgrey"), legend.title =
  axis.title = element_text(face = "bold", size = 14, color = "lightgrey"),
  axis.text = element_text(color = "lightgrey"),
  panel.grid.major = element_line(color = "lightgrey"),
  panel.grid.minor = element_line(color = "lightgrey"), axis.line.x = element_line(color = "black")) +
  labs(title = "Raincloud Plot: ENSG00000001460.18 Counts\nby Gender & Vital Status") +
  scale_fill_brewer(palette = "Accent") +
  scale_color_brewer(palette = "Accent")
```


Raincloud Plot: ENSG00000001460.18 Counts by Gender & Vital Status
ENSG00000001460.18
2000
1000
0
female
male
gender
vital_status
Alive
Dead