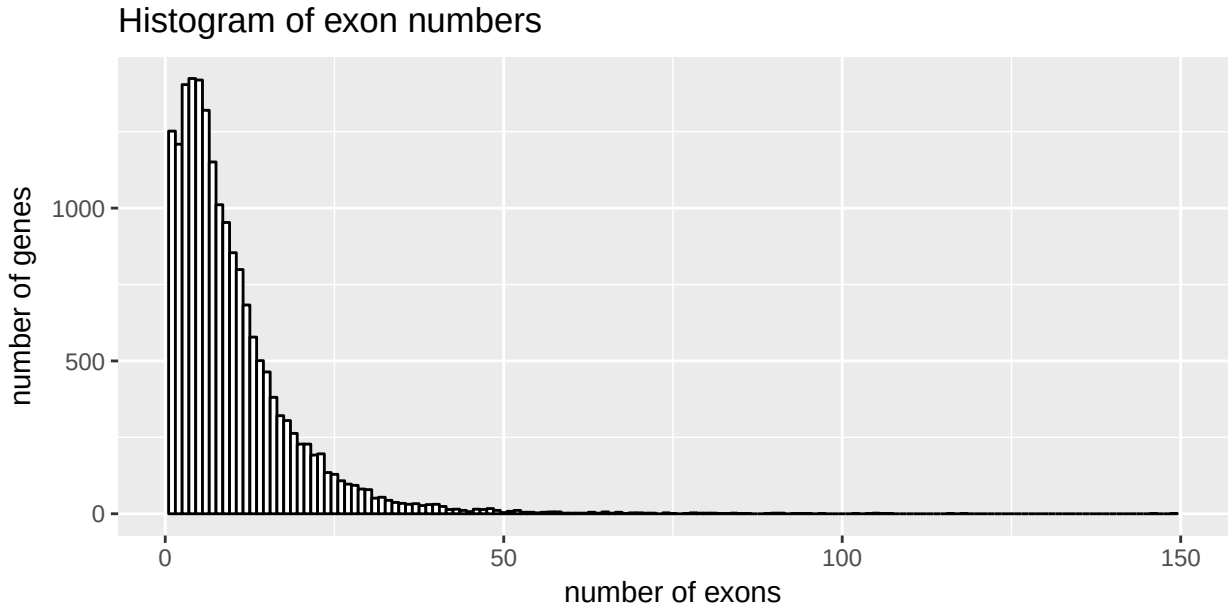
Homework 1, group 5

Ann C. Printz
Jesus V. Torresano Lominchar
Peter C. Holm
Sesilie B. Weiss
William P. Bullock

```
# Load packages
library(tidyverse) # includes ggplot2, magrittr, dplyr etc.
library(gridExtra)
```

1. Make a histogram that shows what the typical number of exons is. Adjust the bins so that we can pinpoint exactly what number of exons that is the most common. Comment the plot.

```
df <- read.table("gene_lengths_v2.txt", sep = "\t", h = T, quote = "")

df %>%
  ggplot(aes(x = exon_count)) +
  geom_histogram(binwidth = 1, color = "black", fill = "white") +
  labs(title = "Histogram of exon numbers",
           x = "number of exons",
           y = "number of genes")
```



Comments on the plot:

- The exon count is clearly not normally distributed, instead it resembles a poisson distirbution.
- The most common number of exons is 4.
- There is a lot of genes with only one exon.
- A few genes have a high number of exons.

2. Add an additional column to the dataframe that contains the total length of introns for each gene

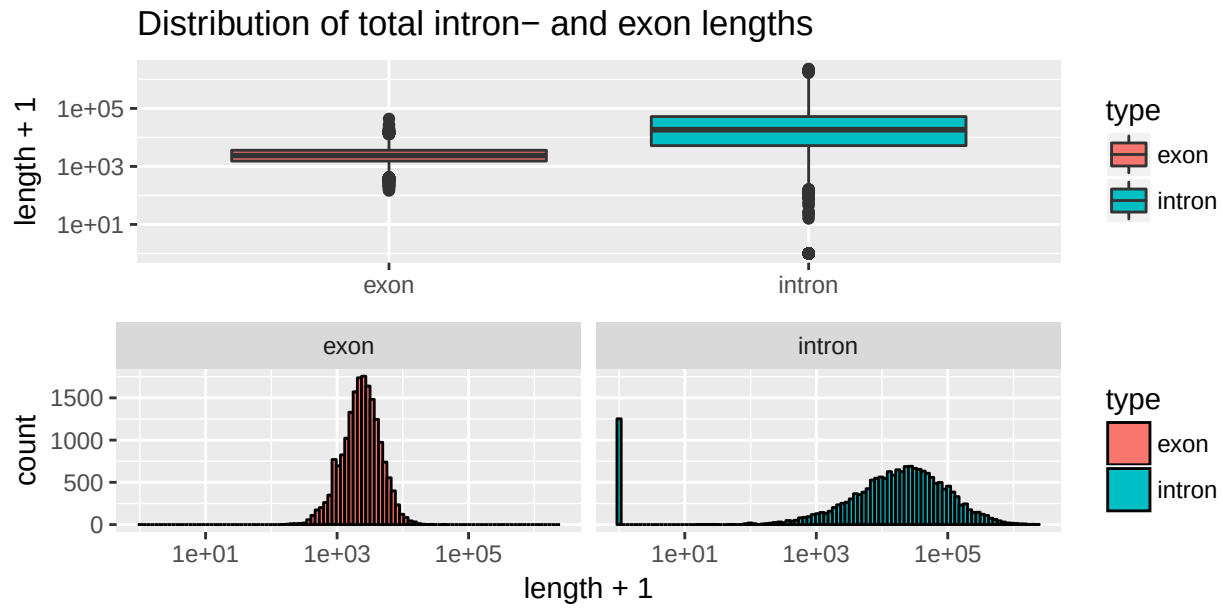
```
df$intron_length <- df$genome_length - df$mrna_length

head(df) %>% knitr::kable(caption = "First 6 rows of gene.lengths2")
```

Table 1: First 6 rows of gene.lengths2

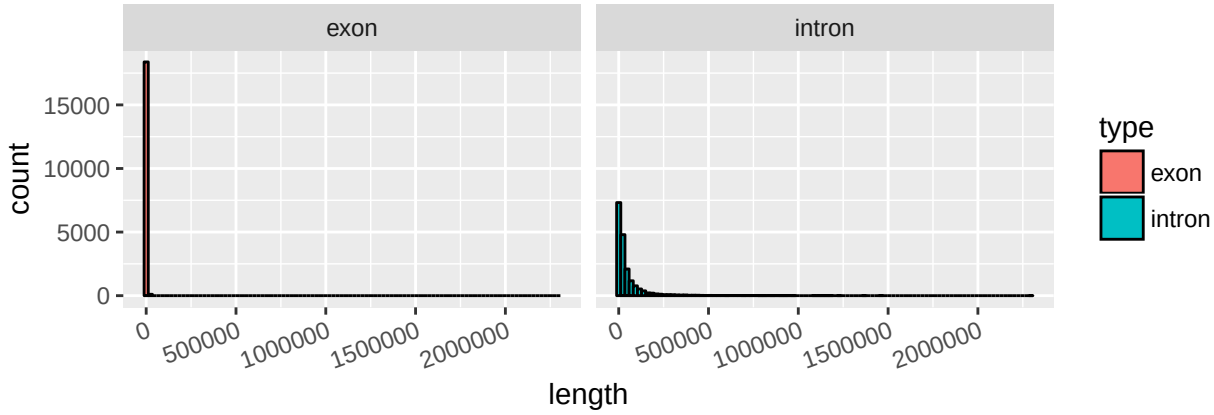
name	mrna_length	genome_length	exon_count	intron_length
PP8961	2596	2596	1	0
FLJ00038	794	2615	6	1821
OR4F5	918	918	1	0
OR4F3	937	937	1	0
OR4F16	937	937	1	0
SAMD11	2555	18842	14	16287

3. Make histograms and boxplots showing the distribution of total exon and total intron lengths, all as subplots in the same larger plot, where each dataset have a different color. On the histograms, the number of bins should be exactly the same, and the x-axis should have the same scale. Comment the plot. Are exons larger than introns or vice versa?

```
p0 <- df %>%
  gather(mrna_length, intron_length, key = "type", value = "length") %>%
  mutate(type = replace(type, type == "mrna_length", "exon")) %>%          # Renaming
  mutate(type = replace(type, type == "intron_length", "intron")) %>%      # Renaming
  ggplot(aes(fill = type))

p1 <- p0 + geom_boxplot(aes(type, length + 1)) +  # Creates boxplot
  labs(x = NULL, title = "Distribution of total intron- and exon lengths") +
  scale_y_log10()  # log10 scale on the y-axis

p2 <- p0 + geom_histogram(aes(x = length + 1), color = "black", bins = 100) + # Creates histogram
  facet_wrap(~type) + scale_x_log10()  # log10 scale on the x-axis

grid.arrange(p1, p2) # put two plots one on top of the other
```




```
# a non-transformed version of the histogram for comparison
p0 + geom_histogram(aes(x = length), color = "black", bins = 100) +
  facet_wrap(~type) +
  theme(axis.text.x = element_text(angle = 20, hjust = 1))
```



Comments:

- As seen by the non-transformed histograms, the data is highly skewed. This means that it is difficult to properly visualize the data. For this reason we have created the plots with logarithmic axes.
- As the data set contains a lot of zero-values for intron lengths, we have used log10(x + 1) as to not lose these values during transformation.
- From the plots it can be seen that the total intron length are generally much longer than the total exon length.
- The length variation is much larger for the total intron length than for the total exon length.

4. Are the mRNA lengths significantly longer than the total intron lengths, or is it the other way around?

```
df %>%
  with(., wilcox.test(intron_length, mrna_length, paired = F))
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  intron_length and mrna_length
## W = 283380000, p-value < 2.2e-16
## alternative hypothesis: true location shift is not equal to 0
```

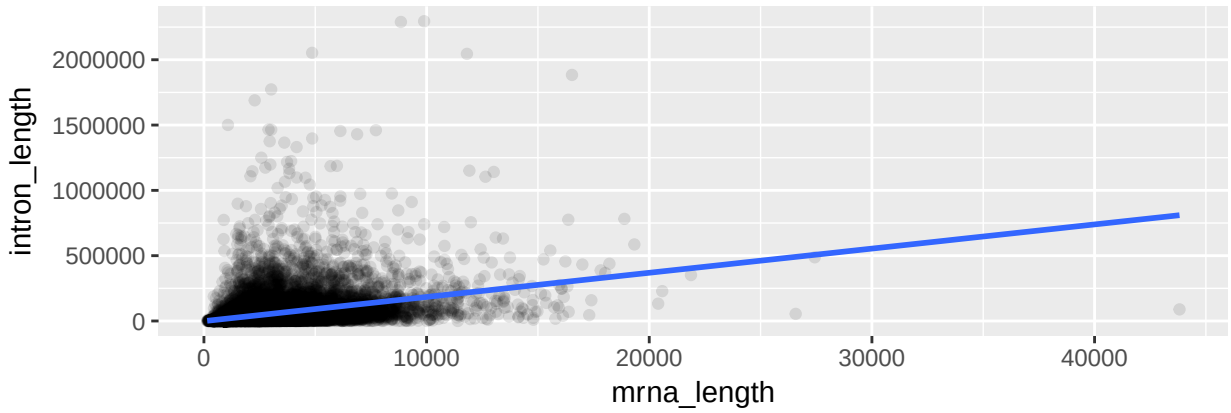
H0: Distribution of total intron/exon lengths is the same

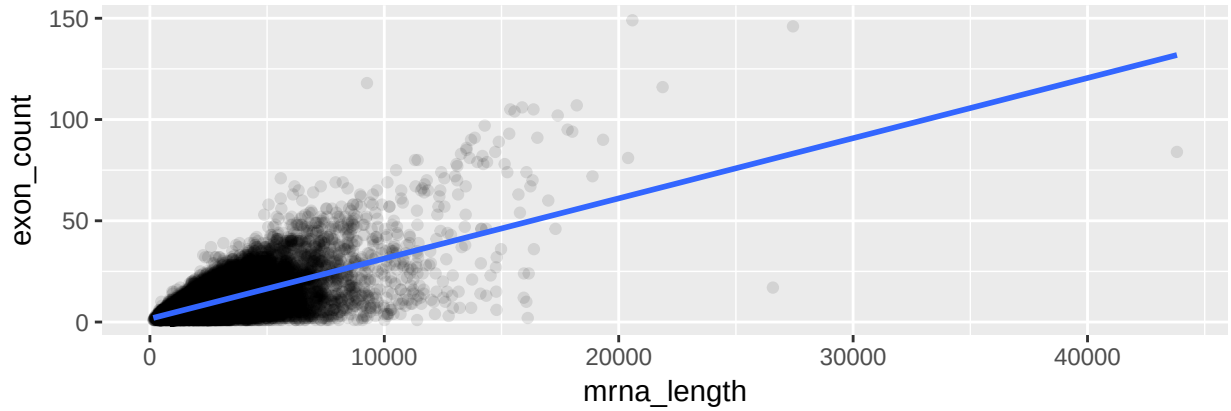
H1: Distribution of total intron/exon lengths is different.

Conclusion: using the non-parametric wilcox test, as the data is not normally distributed, we obtain a p-value of less than $2.2 \cdot 10^{-16}$. On a significance level of 95% this is highly significant, and we can reject the null hypothesis. The conclusion is that there is a significant difference in total length of exons and introns, and combined with the plots from the previous exercise, it is clear that total intron lengths are larger than total exon lengths.

5. Continuing on the same question: is the total exon length more correlated to the total intron length than the number of exons? Show this both with a plot and with correlation scores. Comment on the results:

```
p1 <- df %>%
  ggplot(aes(x = mrna_length, y = intron_length)) +
  geom_point(alpha = 0.1) +
  geom_smooth(method = "lm", se = F)

p2 <- df %>%
  ggplot(aes(x = mrna_length, y = exon_count)) +
  geom_point(alpha = 0.1) +
  geom_smooth(method = "lm", se = F)

grid.arrange(p1, p2)
```






```
with(df, cor(mrna_length, intron_length))
```

```
## [1] 0.3473037
```

```
with(df, cor(mrna_length, exon_count))
```

```
## [1] 0.6390378
```

Comments:

- From the correlation values, we can see that there is a greater correlation between total exon length and exon count, than between total exon length and total intron length.
- Correlation values range between -1 and 1 (-1 = perfect negative correlation, 1 = perfect positive correlation, 0 = no correlation). The calculated correlations are positive, but neither are good correlations.

6. What gene has the longest (total) exon length? How long is this mRNA and how many exons does it have? Do this in a single line of R.

```
filter(df, mrna_length == max(mrna_length))[c(1,2,4)]
```

```
##    name mrna_length exon_count
## 1 MUC16       43815         84
```

7. In genomics, we often want to fish out extreme examples like all very short genes, or all very long genes. It is often helpful to make a function to do these tasks it saves time in the long run.

Make a function called count_genes that takes three inputs:

```
count_genes <- function(vector, x1 = 0, x2 = max(vector)){
  n.genes <- length(subset(vector, vector > x1 & vector <= x2))
  out <- n.genes/length(vector)
  return(out)
}
```

Test the function

```
mrna_length <- df$mrna_length

count_genes(mrna_length)
```

```
## [1] 1
```

```
count_genes(mrna_length, x1 = 10000)
```

```
## [1] 0.01130402
```

```
count_genes(mrna_length, x1 = 1000, x2 = 10000)
```

```
## [1] 0.873276
```

```
count_genes(mrna_length, x1 = 0, x2 = 100)
```

```
## [1] 0
```