Homework 1

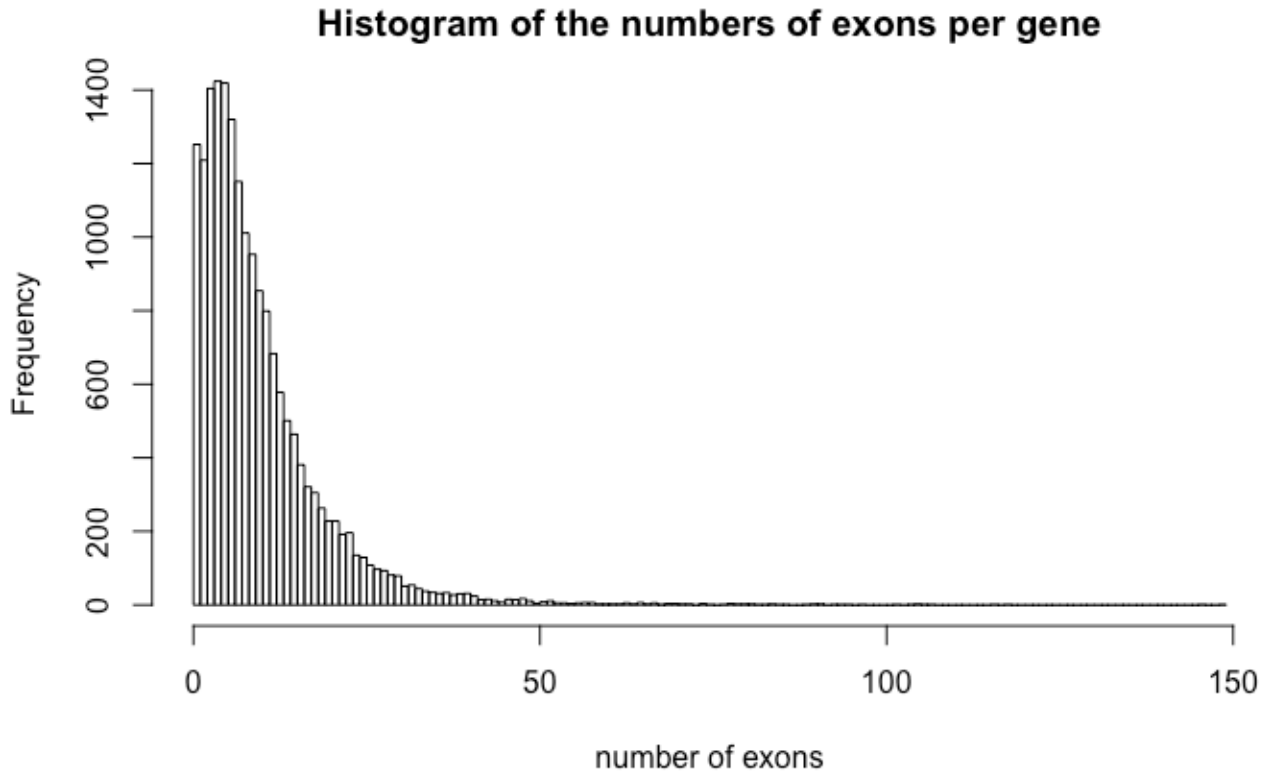
Marie Holm Abildgaard, Natthawut Adulyanukosol, Mathias Isager Fessler, Enrique Goñi Echeverria, Adrian Otamendi Laspiur, Katrine Harpelunde Poulsen

1. The most common number of exons

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.

The given file has a header row, so we need to specify this to load the dataset correctly.

```
genes_df <- read.table("gene_lengths_v2.txt", header=TRUE)

hist(genes_df$exon_count, breaks = c(0:max(genes_df$exon_count)), xlim = c(0,max(ge
nes_df$exon_count)), main = "Histogram of the numbers of exons per gene",
xlab = "number of exons")
```



The histogram shows that the most common number of exons per gene is 4 exons. The distribution of the number of exons is right-skewed, where almost none of the genes have more than 50 exons.

2. Total length of introns

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

```
genes_df$intron_length <- genes_df$genome_length - genes_df$mrna_length
```

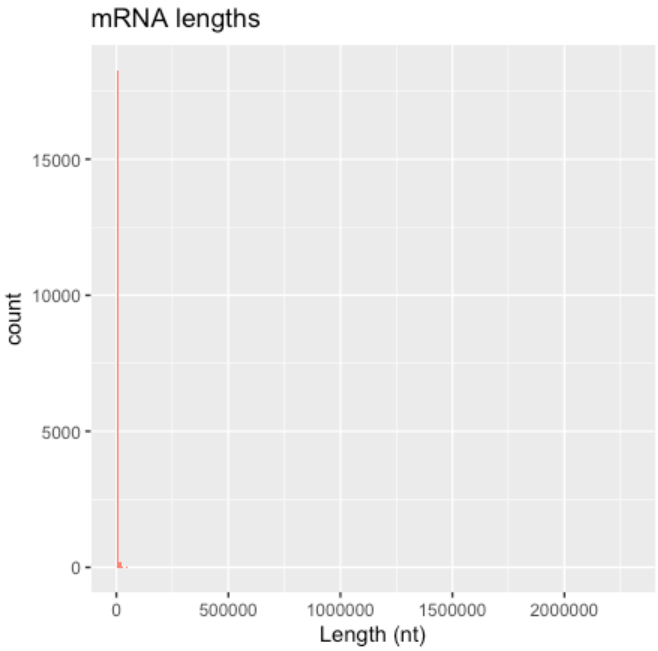
3. The length of total exon and total intron

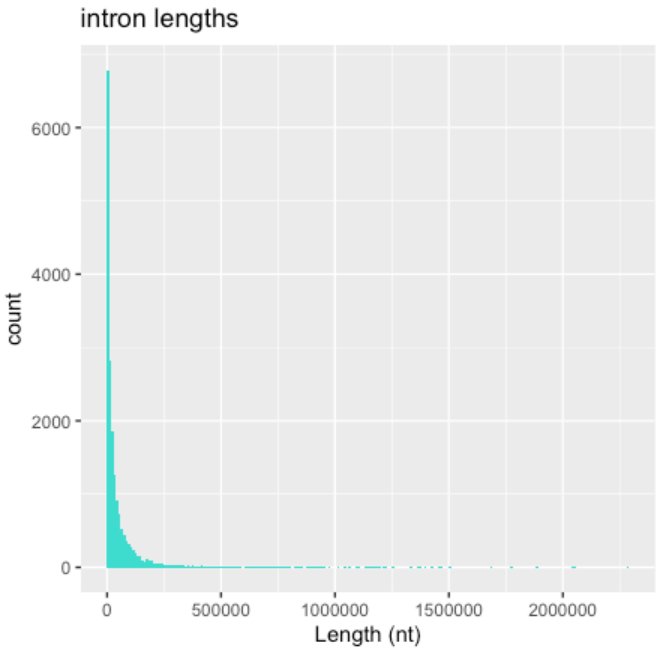
Make histograms and boxplots showing the distribution of total exon and total intron lengths, all as subplots in the same larger plot, where each dataset have a different color. On the histograms, the number of bins should be exactly the same, and the x-axis should have the same scale. Comment the plot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

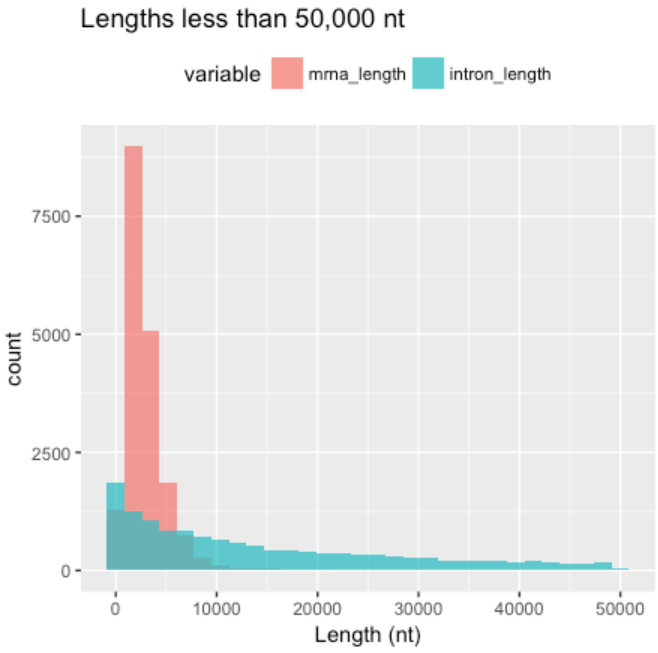
```
library("ggplot2")
library("gridExtra")
library("reshape2")

# use the maximum value of the intron lengths and 0 as x-axis limits
plot1 <- ggplot(data=genes_df, aes(mrna_length))  +
  geom_histogram(breaks=seq(0, max(genes_df$intron_length), by=10000),
                 fill="salmon") +
  labs(title="mRNA lengths", x = "Length (nt)")

plot2 <- ggplot(data=genes_df, aes(intron_length))  +
  geom_histogram(breaks=seq(0, max(genes_df$intron_length), by=10000),
                 fill="turquoise") +
  labs(title="intron lengths", x = "Length (nt)")

# reshape the dataframe
genes_df2 <- melt(genes_df, id.vars = "name")
genes_df2 <- genes_df2[(genes_df2$variable == "mrna_length" |
                        genes_df2$variable == "intron_length"), ]
# filter only length <= 50000
genes_df3 <- genes_df2[genes_df2$value <= 50000, ]

plot3 <- ggplot(genes_df3, aes(x=value, fill=variable)) +
  geom_histogram(alpha=0.7, position="identity") +
  theme(legend.position="top") +
  labs(title="Lengths less than 50,000 nt", x = "Length (nt)")

plot4 <- ggplot(genes_df2, aes(x=variable, y=value, fill=variable)) +
  geom_boxplot() +
  theme(legend.position="none") +
  labs(title="Lengths of mRNA and intron", y = "Length (nt)")

grid.arrange(plot1, plot2, plot3, plot4, ncol=2)
```

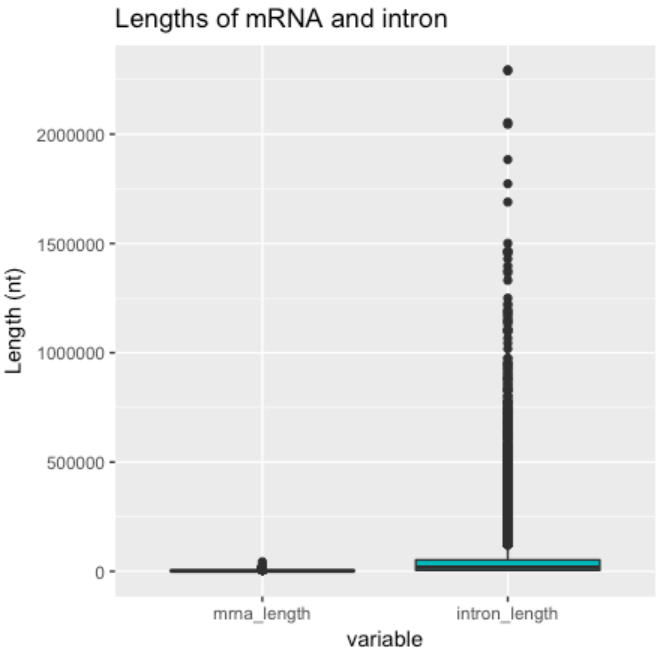
To get a better visualization of the distributions of short exons and introns, we plotted overlaying histograms of those less than 50,000 nt in length.

Histograms and boxplots show that introns are mostly longer than exons. The boxplots show that the intron lengths span a larger range.









4. Hypothesis testing

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

As illustrated by the histograms and other normality tests (not shown here), the mRNA and intron lengths are not normally distributed, thus we cannot use a parametric test. Therefore, we performed the non-parametric Wilcoxon test.

The samples are two separate sets of measurements. Hence, we used an unpaired test.

From the plots earlier, we observe that mRNA lengths are mostly shorter than intron lengths. We would like to prove this observation statistically by testing the difference in one direction. Thus, a one-sided was used with the following hypotheses.

H_0: The mRNA lengths are equal to or longer than the intron lengths.
H_a: The mRNA lengths are significantly shorter than the intron lengths.

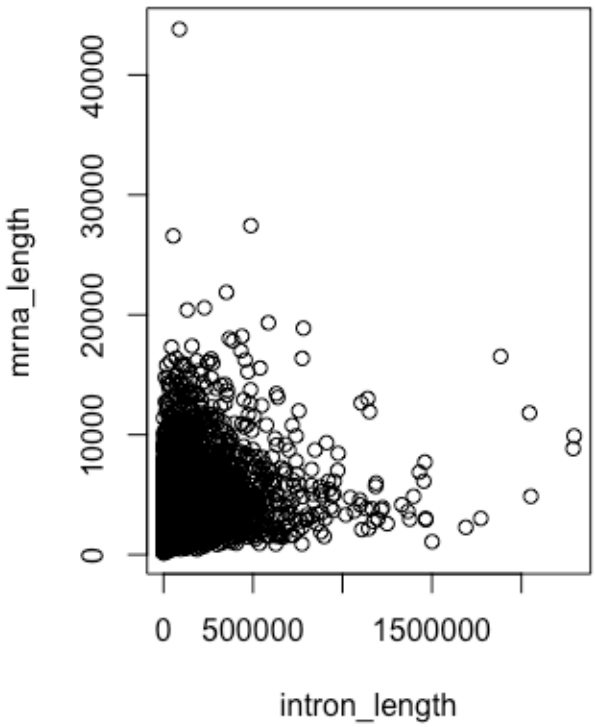
```
wilcox.test(genes_df$mrna_length, genes_df$intron_length, alternative = "less",
            paired = FALSE)

##
##  Wilcoxon rank sum test with continuity correction
##
## data:  genes_df$mrna_length and genes_df$intron_length
## W = 58458000, p-value < 2.2e-16
## alternative hypothesis: true location shift is less than 0
```

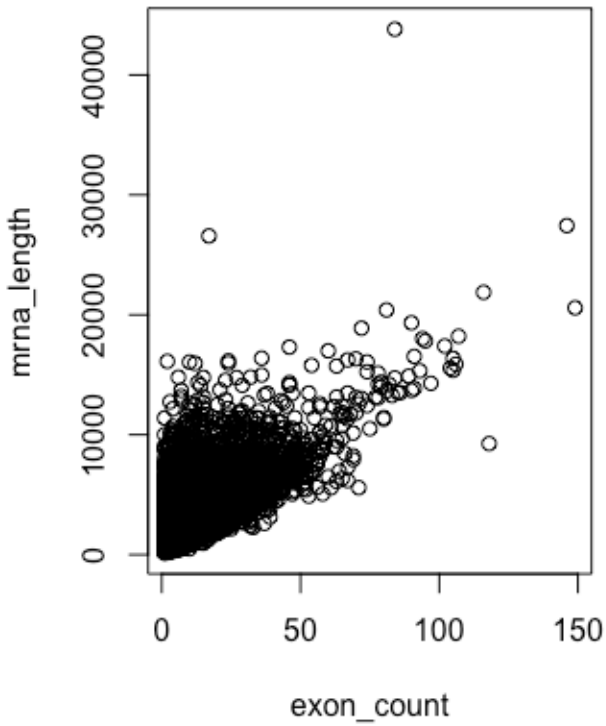
As the p-value is $2.2*10^{-16}$, which is less than 0.05, the null hypothesis is rejected.
Therefore, the mRNA lengths are significantly shorter than the intron lengths.

5. Correlation

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 your result.

```
par(mfrow=c(1,2))
plot(mrna_length ~ intron_length, data = genes_df)
plot(mrna_length ~ exon_count, data = genes_df)
```





As all parameters (mRNA lenghts, the number of exons per gene, and intron lengths) are not normally distributed, we use non-parametric Spearman's correlation.

```
cor(genes_df$mrna_length, genes_df$intron_length, method="spearman")
## [1] 0.5367173
cor(genes_df$mrna_length, genes_df$exon_count, method="spearman")
## [1] 0.5407801
```

The correlation coefficient between mRNA lenghts and intron lengths is 0.5367173.
The correlation coefficient between mRNA lenghts and the number of exons per gene is 0.5407801.

The correlation values are highly similar. Ranking-wise, the mRNA lengths are slightly more correlated to the number of exons than to intron lengths. However, the correlation is rank-based. Therefore, we cannot predict the length of mRNA given either the absolute number of exons or the absolute length of intron.

6. Longest total exon length

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 (without using ";").

```
genes_df[genes_df$mrna_length == max(genes_df$mrna_length), c(1,2,4)]
```

```
##       name mrna_length exon_count
## 8385 MUC16       43815         84
```

7. Extremes removal

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 two inputs: a. A vector with mRNA lengths b. A cutoff x1 which by default should be set to 0 c. A cutoff x2 which by default should be set to the longest (total) mrna length of the input vector, as you did in "6)". d. Then, the function should count the number of mRNAs that are no less than (<=) x2 but larger than (>) x1; and finally return the fraction of this count over the total count of mRNAs.

```
count_genes <- function(mrna_length_vec, x1=0, x2= max(mrna_length_vec)){
  return(length(mrna_length_vec[(x1 < mrna_length_vec) & (mrna_length_vec <= x2)])
         /length(mrna_length_vec))
}
```

Test this function with the mRNA lengths using the the five settings below: i) Using the default of x1 and x2; ii) Using the default of x2 and set x1=10000; iii) x1=1000 and x2=10000; iv) x1=100 and x2=1000; v) x1=0 and x2=100.

```
count_genes(genes_df$mrna_length)
## [1] 1
count_genes(genes_df$mrna_length, x1=10000)
## [1] 0.01130402
count_genes(genes_df$mrna_length, x1=1000, x2=10000)
## [1] 0.873276
count_genes(genes_df$mrna_length, x1=100, x2=1000)
## [1] 0.11542
count_genes(genes_df$mrna_length, x1=0, x2=100)
## [1] 0
```