Homework 1

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

0. Load the dataset

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)
```

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.

```
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")
```
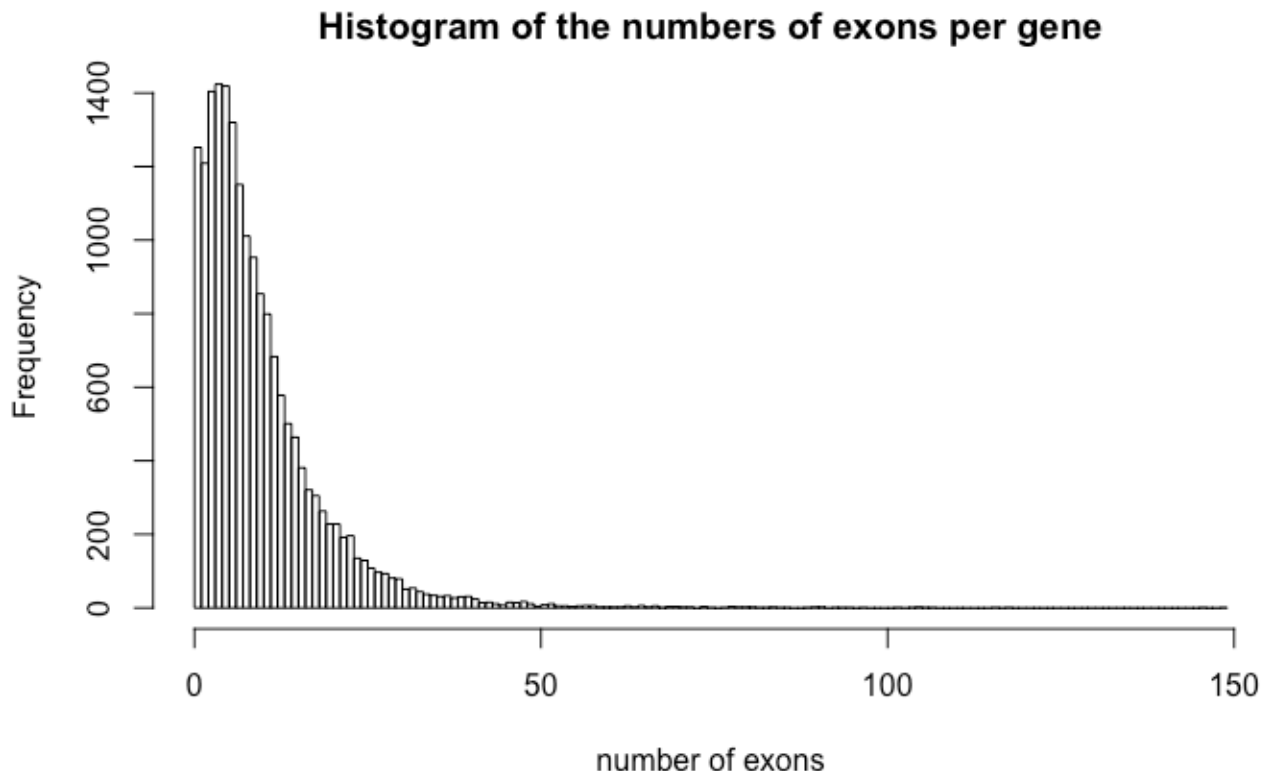

The histogram shows that the most common number of exons per gene is 4 exons. The distribution of the number of exons is right-skewed. Half of the genes have less than 10 exons (median).

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?

```
par(mfrow=c(1,2))
hist(genes_df$mrna_length, col="blue", breaks=seq(0,2500000,by=10000),
     main="exon", xlab="length (nt)")
hist(genes_df$intron_length, col="red", breaks=seq(0,2500000,by=10000),
     main="intron", xlab="length (nt)")
```
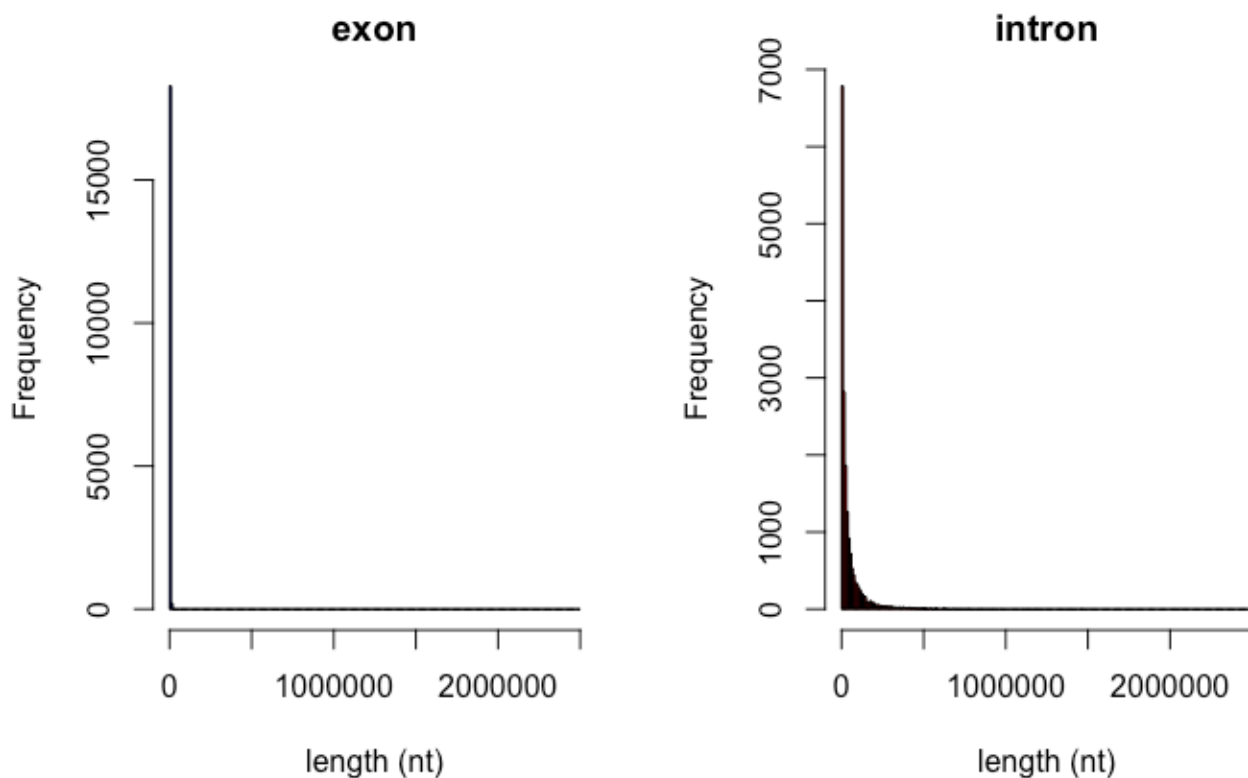

To get a better visualization of the distributions of short exons and introns, we plotted overlaying histograms of those less than 50,000 nt in length.

```
par(mfrow=c(1,2))
hist(genes_df$mrna_length[genes_df$mrna_length<50000], col=rgb(0,0,1,0.5),
     breaks=seq(0,50000,by=1000), main="lengths less than 50,000 nt",
```

```
      xlab="length (nt)")
hist(genes_df$intron_length[genes_df$intron_length<50000], col=rgb(1,0,0,0.5),
      breaks=seq(0,50000,by=1000), add=TRUE)


boxplot(genes_df$mrna_length, genes_df$intron_length, col=c('blue', 'red'),
      names=c("exon length", "intron length"), ylab="length (nt)")
```
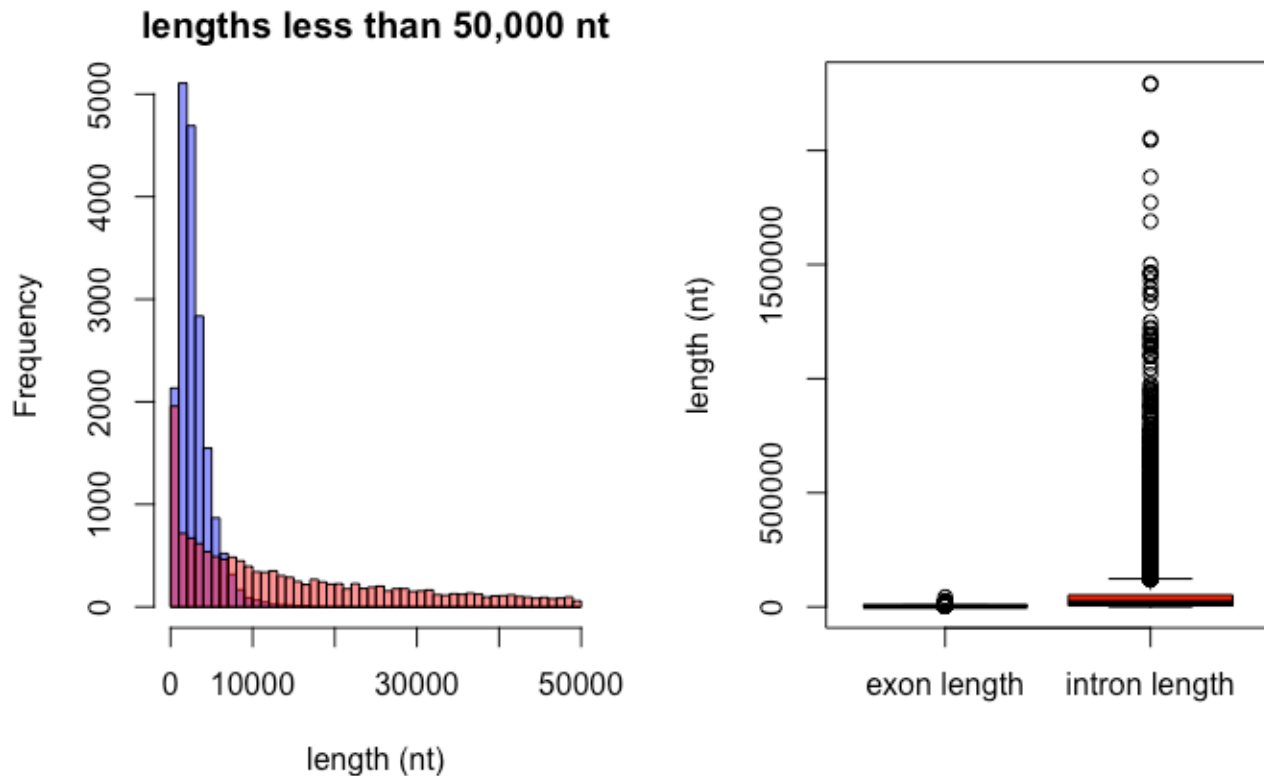

The length of total exon (blue) and total intron (red).

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?

We cannot use parametric t-test, because from normality tests, the mRNA lengths and intron lengths are not normally distributed. Two-sample Wilcoxon test, as known as Mann-Whitney test, is used instead.

H0: The mRNA lengths are equal to or longer than the intron lengths.
H1: The mRNA lengths are significantly shorter than the intron lengths.

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

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)
exonL_intronL = lm(mrna_length ~ intron_length, data = genes_df)
abline(exonL_intronL, col="red")

plot(mrna_length ~ exon_count, data = genes_df)
exonL_exonN = lm(mrna_length ~ exon_count, data = genes_df)
abline(exonL_exonN, col="red")
```
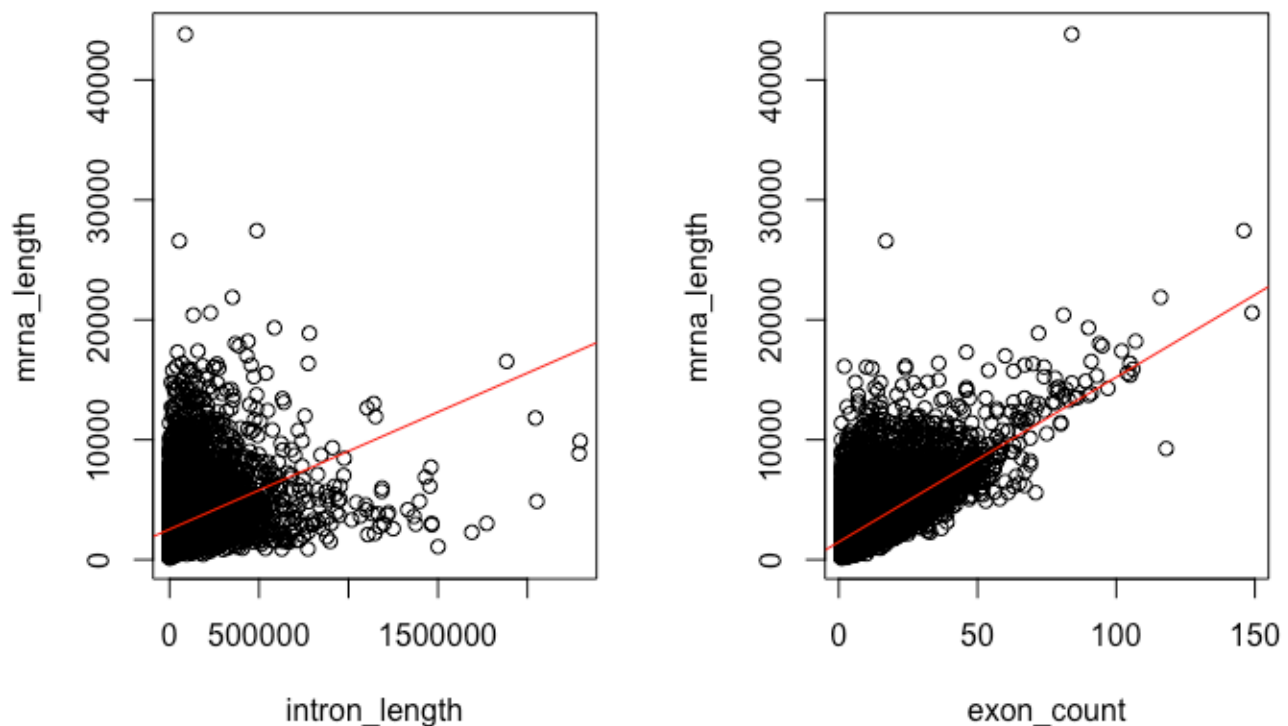

As all parameters (mRNA lenghts, the number of exons per gene, and intron lengths) are not normally distributed, we use non-parametric Spearman's correlation.

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

The correlation values are highly similar. However, from the plots, the mRNA lenghts appear to be more correlated to the number of exons than intron length, but this is not represented by the correlation values.

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
```