Now let's try using these packages and the commands we've learned in an example:

Let's first import a file with variants (SNPs, indels, etc...) from chr1 of *Mus musculus*:

```
> dbsnp137 <- import("mm10_snp137_chr1_trunc.bed.gz")
```

We want to find all variants within exons on this mouse chromosome. Let's first collapse all overlapping exons in the mouse `TranscriptDb` object we created earlier and create an object with only exons from chr1:

```
> collapsed_exons <- reduce(exons(txdb), ignore.strand=TRUE)

> chr1_collapsed_exons <- collapsed_exons[seqnames(collapsed_exons) == "chr1"]
```

Before extracting variants in exons, let's first inspect our variant file:

```
> summary(width(dbsnp137))
```

If a variant has a width of 0, we cannot find its overlap with exon ranges, so we must adjust its width to do this:

```
> dbsnp137_resized <- dbsnp137

> zw_i <- width(dbsnp137_resized) == 0

> dbsnp137_resized[zw_i] <- resize(dbsnp137_resized[zw_i], width=1)
```

We can now pull out those variants that overlap exons on chromosome 1 by creating a `hits` object:

```
> hits <- findOverlaps(dbsnp137_resized, chr1_collapsed_exons,
        ignore.strand=TRUE)
```

and determine the number of variants and the proportion of variants that are exonic:

```
> length(unique(queryHits(hits)))

> length(unique(queryHits(hits)))/length(dbsnp137_resized)
```

We can also use the `countOverlaps()` function to find the number of variants per exon (note we have to reverse the order of the query since we're finding values per exon now)

```
> var_counts <- countOverlaps(chr1_collapsed_exons, dbsnp137_resized, ignore.stran
```

and we can append this to our `GRanges` object that includes exons:

```
> chr1_collapsed_exons$num_vars <- var_counts
```