6. Çoklu lokus evrimi

Uygulama

Fisher (https://en.wikipedia.org/wiki/Ronald_Fisher) who was part of the modern synthesis (https://en.wikipedia.org/wiki/Modern_synthesis_(20th_century)) of population genetics and evolutionary theory.

The iris dataset

To get an understanding of how we apply an ANOVA, we will first use the `iris` dataset, which comes as part of the R distribution. You actually encountered this dataset for the first time back in the first R session (https://evolutionarygenetics.github.io/Introduction.html). We'll make it a `tibble` so it is easier to visualise.

```
iris <- as.tibble(iris)
```

```
## Warning: `as.tibble()` is deprecated, use `as_tibble()` (but mind the new semantics).
## This warning is displayed once per session.
```

Take a look at the `iris` dataset now. You can learn more about it with `?iris`. The dataset is actually a fitting one to describe the use of ANOVA in biology as it is one that Fisher often used to test his statistical models.

Anyway, what does `iris` contain? Well it is a set of various morphological measurements from 3 species of flowers. There are actually four measurements from 50 of each iris species. We can verify this using some tidyverse commands.

```
iris %>% group_by(Species) %>% tally()
```

Just to recap here - we used `group_by` to group the dataset by the `Species` column and then `tally` to count the number of rows.

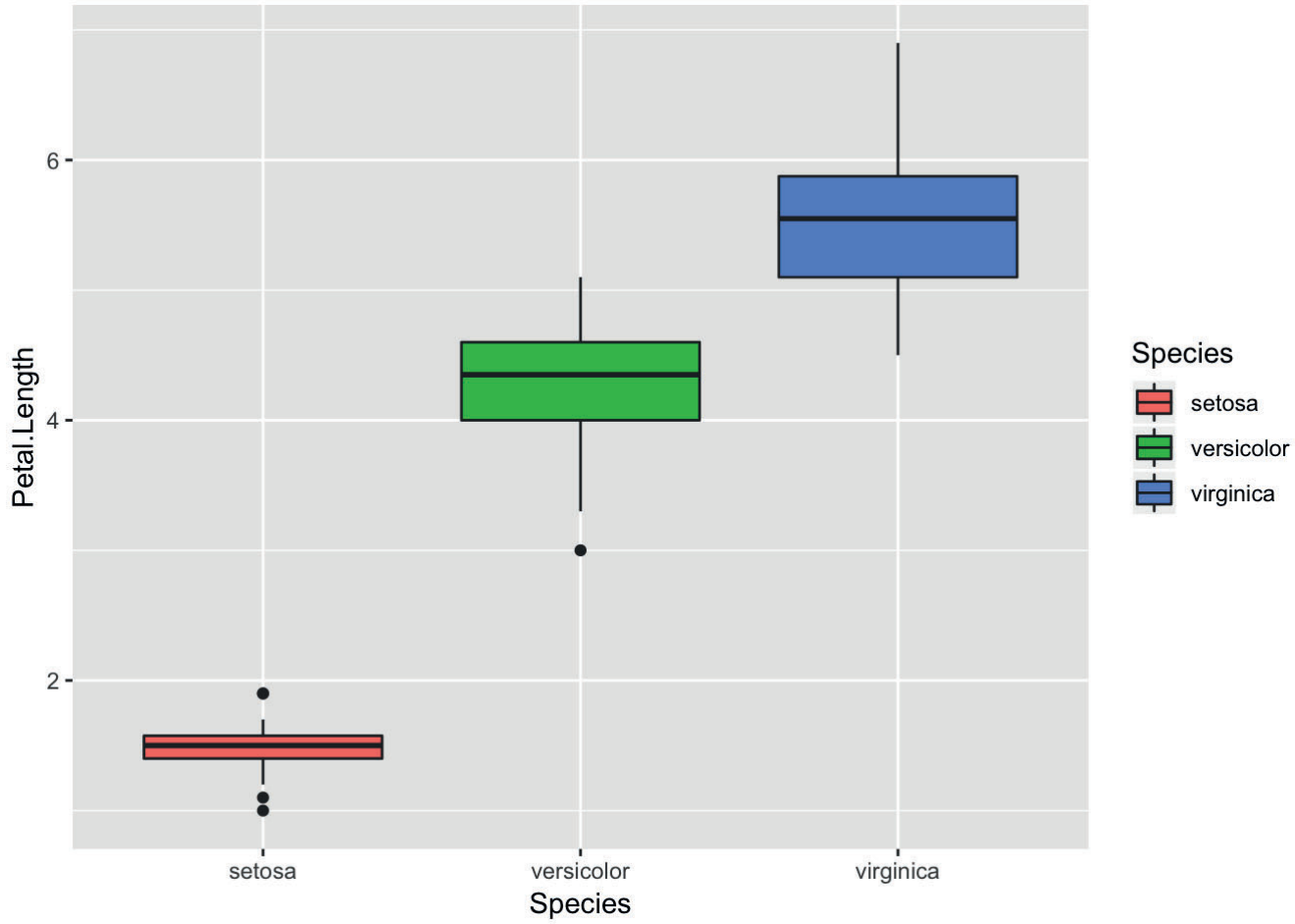
Take one more look at the `iris` dataset to note that the `tibble` contains the four measurements (`Sepal.Length`, `Sepal.Width`, `Petal.Length` and `Petal.Width`) and also a fifth column, `Species which unsurprisingly lets us know the species.

Examining trait differences between groups.

As we're aware of by now, one of the strengths of R is its ability to visualize data. It really is a good idea to always do this before you perform a statistical test, just to get an idea of what the data is showing.

We are eventually going to use an ANOVA to test whether there is a difference between petal length between the species. But first let's plot this to get some intuition on what such a test will actually show. We will use `geom_boxplot` to make a boxplot.

```
ggplot(iris, aes(Species, Petal.Length, fill = Species)) + geom_boxplot()
```



Boxplots are quite useful in this case since they show the distribution of the data. You can see from the one we drew here that distributions of the three species are clearly different with respect to their petal length. The black bars in each of our boxplots are the **median** values of the measurement for each group - so again these are obviously different.

Before we turn to actually testing whether these differences are significant from one another, let's get some confirmatiion on the actual mean values of petal length for the three species. We can do this using a `tidyverse` approach.

```
iris %>% group_by(Species) %>%
  summarise(mean_petal_length = mean(Petal.Length))
```

This is very similar to what we did previously, we grouped our data by species using `group_by` but this time, instead of using `tally`, we used `summarise` to calculate the mean `Petal.Length` for each group. Note that the `mean_petal_length` part of the argument to `summarise` is just to specify the name of the column in the output. Clearly we can see that the **mean** values of these traits are different between the species.

Using ANOVA to test for differences between groups

Now that we can see there appears to be some difference between species in petal length based on our boxplots, we can actually formally test this. As we learned earlier, ANOVA is essentially a way to test whether the **means** of different groups are equal. We are testing whether the means of a **dependent variable**, petal length in this case, differs with our **independent variable**, which is species here.

To conduct an ANOVA in R, we will use the function `aov` like so to create an object we will call `model`.

```
model <- aov(Petal.Length ~ Species, data = iris)
```

What did we do? Let's break it down a little. The `aov` function just means we specified an analysis of variance.

We then used a formula as our first argument `Petal.Length ~ Species`. This is essentially like saying, “How does petal length vary with species”? A little confusingly in R, the dependent variable comes first and *then* the dependent variable. This part of our argument to `aov` specifies what we are actually testing. Finally we used `data = iris` to specify that we are using the `iris` data for this calculation.

Next we can examine the output of our model, to get an understanding of what it shows. To do this, we need to use a special function, `summary`:

```
summary(model)
```

By using `summary`, we return an ANOVA table. This can be a little confusing the first time you look at one, but it is actually quite straightforward. The first row of the table is for `Species`. The two last values in this row are an **F** statistic and a **p-value**. We will return to what the *F* statistic actually means in a short while.

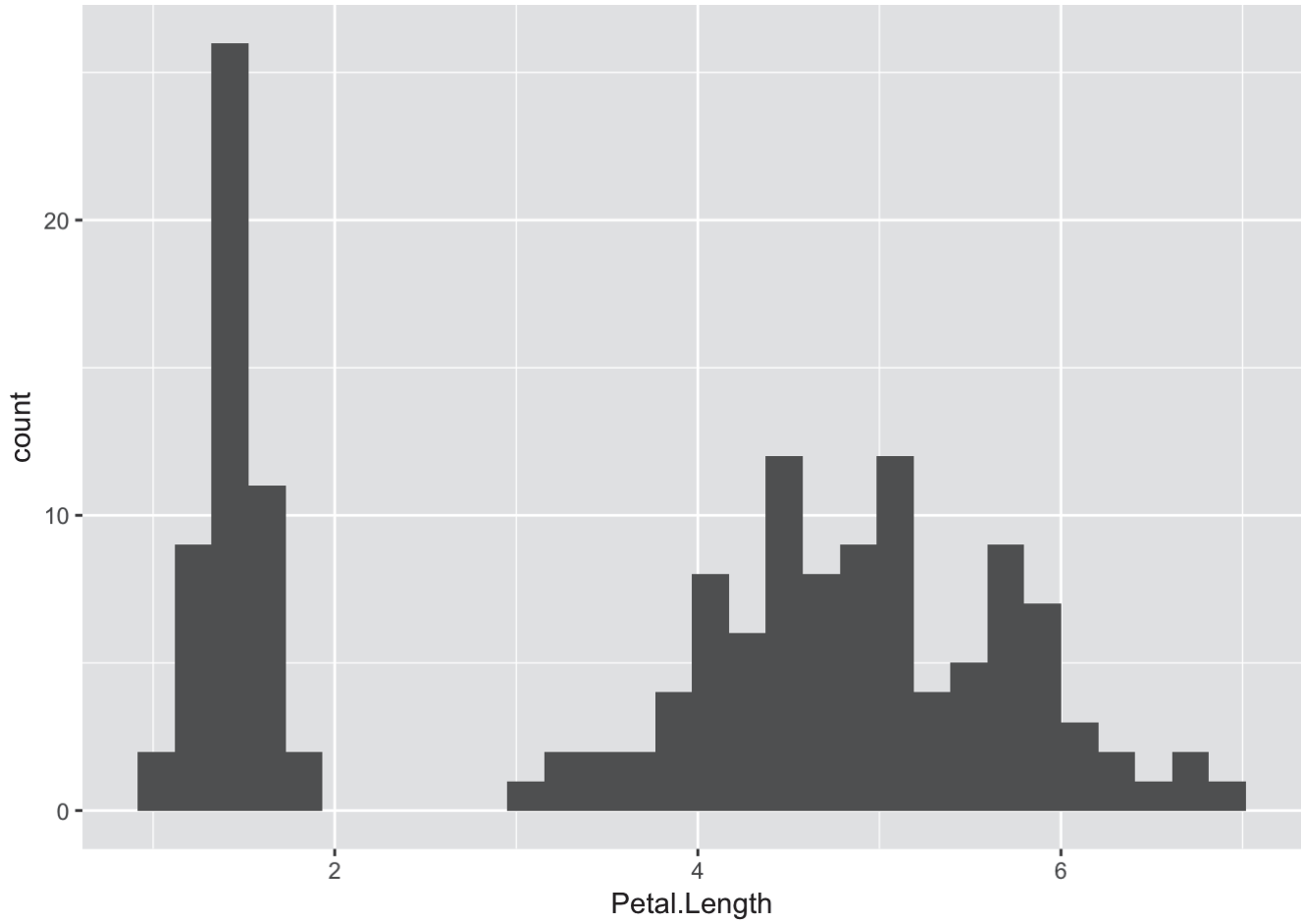
Let’s focus instead on the *p*-value. This is essentially a way of determining whether the results we see could be explained by chance. It is a test of **significance** - we did something similar with the Chi-squared test back in Chapter 3 (https://evolutionarygenetics.github.io/Chapter3.html). In the biological sciences, we have an arbitrary cutoff for a *p*-value, set at 0.05. If our *p*-value is lower than this, then we can say that our dependent variable is significantly different among the groups.

Here we can cleary see that our *p*-value is actually extremely small - less than 2×10^{-16} which tells us that petal length is **signficiantly different** between species. The asterisks after the *p* value are also there to tell us this - `***` simply means that our *p*-value is close to zero.

ANOVA and the proportion of variance explained

So far, we have learned about ANOVA in it’s simplest form - i.e. to test for differences among means. But this is not the only way we can think about it and the clue is in the name - it analyses **variance**. What do we mean by this? To illustrate it, let’s have a look at the **total** distribution of `Petal.Length` from the `iris` data. We can do this using a histogram.

```
ggplot(iris, aes(Petal.Length)) + geom_histogram()
```



So, this time we are visualising petal length for all species grouped as one. Imagine for a moment you **didn't know** that we have three species in our dataset. It is pretty obvious from this plot that there is a lot variation in petal length - there appear to be at least two and maybe three peaks in the data.

What might explain why there is so much variation? Well of course, we know there are actually three species included in these measurements. So, if we split our dataset into different species, can that **explain the variance we observe**? This is another way to think of ANOVA - as way of measuring the variance explained by how we split up the dataset.

So this brings us back to the **F** statistic. It is essentially a ratio of the variance explained by our grouping to the variance that occurs within each of them. One way to think of this is that the higher the value of *F*, the more likely our grouping explains a significant proportion of the variance in our data.

What if we want to actually measure how much variance our grouping (species here) actually explains_ Luckily, this is very straightforward. We can look at a slightly different output from the `model` object we created earlier.

```
summary.lm(model)
```

There is a lot of information here! However, you should focus on the last two lines of the output. On the very last line, you will see the *F* statistic and *p*-value from when we called `summary(model)` . Above that are two values of another statistic called the **R-squared** or R^2. Focus on the adjusted R^2.

Our adjusted R^2 is 0.94 which essentially means that by grouping our data by species, we explain 94% of the variance in petal length. So clearly, species has a very important role in explaining the morphological differences in our data. One biological explanation for this is that genetic basis for petal length might be different between the species. We will see in the next section how we can apply a more advanced version of ANOVA to actually determine the genetic basis of a trait.

Performing a QTL analysis in bedbugs

Bedbugs, pesticide resistance and study design

We are going to use the `qtl` package in R to perform a basic QTL analysis on pesticide resistance on an **F2 intercross** in bedbugs, *Cimex lectularius*. Note that all the data we use here come from Fountain et al (2016) (http://www.g3journal.org/content/6/12/4059).

Bed bugs are an human ectoparasite that have experienced a population boom in the last two decades. They are nasty things, so a lot of research has focused on using pesticides to control them. However, some bedbug populations have evolved pesticide resistance, particularly to the most commonly used pyrethroid insecticide.

In their study, Fountain et al crossed two strains of bedbugs. one resistant to the pesticide and the other susceptible. Using a resistant female and susceptible male, they created an F1 generation and then crossed two randomly selected F1 offspring, ultimately producing 90 F2 individuals. They then exposed the F2s to pytheroid insecticide and scored their resistance to it. The pesticide disrupts motor function, so F2s were scored either as susceptible (unable to right themselves if turned over), partially resistant (able to right themselves but walk with some difficulty) and resistant (walk normally, no apparent effect on motor control).

The two grandparents, the two F1 parents and the 90 F2s were then genotyped using RAD-sequencing. RAD-sequences were mapped to a draft bedbug genome and then SNPs were called from 12 962 RAD tags. During their analysis, the authors discarded some individuals and markers so the dataset we are working is a subset of the original data.

Reading in the bedbug data

First we need to download the data, which is here (https://evolutionarygenetics.github.io/bedbugs_cross_data.csv)

Since we loaded the `qtl` package at the start of the tutorial, we can now easily read in the data using the following command.

```
bedbugs <- read.cross(format = "csv", dir = "",
                      file = "./bedbugs_cross_data.csv",
                      genotypes = c("AA", "AB", "BB"),
                      estimate.map = FALSE)
```

When you run this command, you will see that we read in the data from 71 individuals and 334 markers. Strangely, there are only two phenotypes here. This isn't right, there should be three, so we will need to correct this later.

Let's break down what we did here - we used `qtl`'s `read.cross` function to read in our cross data. We specified the format as a comma-separated variable file, we specified the directory the data is in (left blank here because it is in the same directory we are working in) and also the path to the file.

We also specified how our genotypes are encoded using the `genotypes` argument and importantly, we specified `estimate.map = FALSE` to ensure that we are only reading in the data and not creating a linkage map at this stage. It is worth noting at this point that `qtl` is an extremely powerful and complex package with a lot of options and functions. At the end of the tutorial, we will point you towards other resources that can help you learn more about it, but for now we can ignore these options. However if you are interested in learning a little more, you can learn more about these arguments by looking at the help like so: `?read.cross`.

One last point, try typing in `bedbugs` to see what you get in the R console. You'll see a lot of summary information on the cross object we created when we red in the data. Again, it says there are only 2 phenotypes…

Correcting the phenotypes

So, we need to correct the phenotypes? Let's first take a look at what is stored so far.

```
# look at the phenotypes
bedbugs$pheno
```

Aha, the reason R says there are 2 phenotypes is because there are two columns in the phenotype data.frame - one called `id` and the other called `res`. This second vairable is the resistance (i.e. S = susceptible, PR = partially resistant and R = resistant). So we can easily assign this as an environmental variable for now.

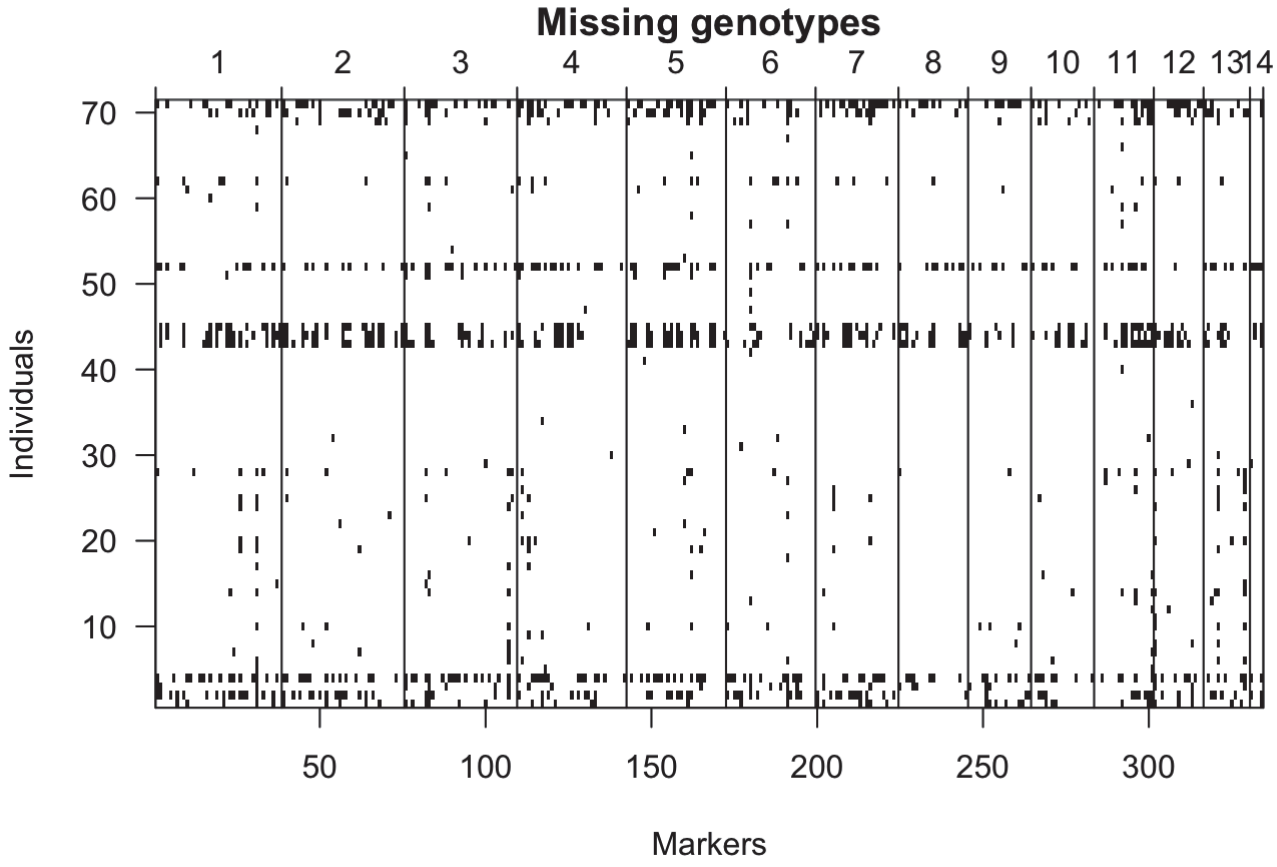
```
# Assign resistance as an environmental variable
res <- bedbugs$pheno$res
```

Incidentally, how did we know to use `$pheno` to access the phenotypic data? Using the `objects` function on the `bedbugs` object - i.e. `objects(bedbugs)` will give us a rundown of what we can access inside.

Exploring the data

Now our data is read into the R environment and we have everything setup, a good first start is to get an idea of what is going on with our data. We have learned already how good it is to plot data, and luckily for us, `qtl` has a number of functions to make this straightforward. Let's use the `plotMissing` function

```
plotMissing(bedbugs)
```



Let's break this plot down. Since this data is the finalised data from Fountain et al, we already have 14 linkage groups assembled - hence the 14 vertical 'blocks' across the plot. When there is a black tile, we have missing data - and the markers are plotted along the x-axis.

It takes a bit of practice to get used to reading these sort of plots, but essentially you can see that there are several individuals which are lacking genotypes across multiple markers.

Examining the linkage map

Now we have explored the data, our next step is to actually look at the linkage map. A linkage map is the position of markers in the genome in terms of their recombination distance from one another - i.e. how many recombination events occur between them. Lets take a look at a summary of the linkage map. Note that sometimes, a linkage map is also referred to as a **genetic map**.

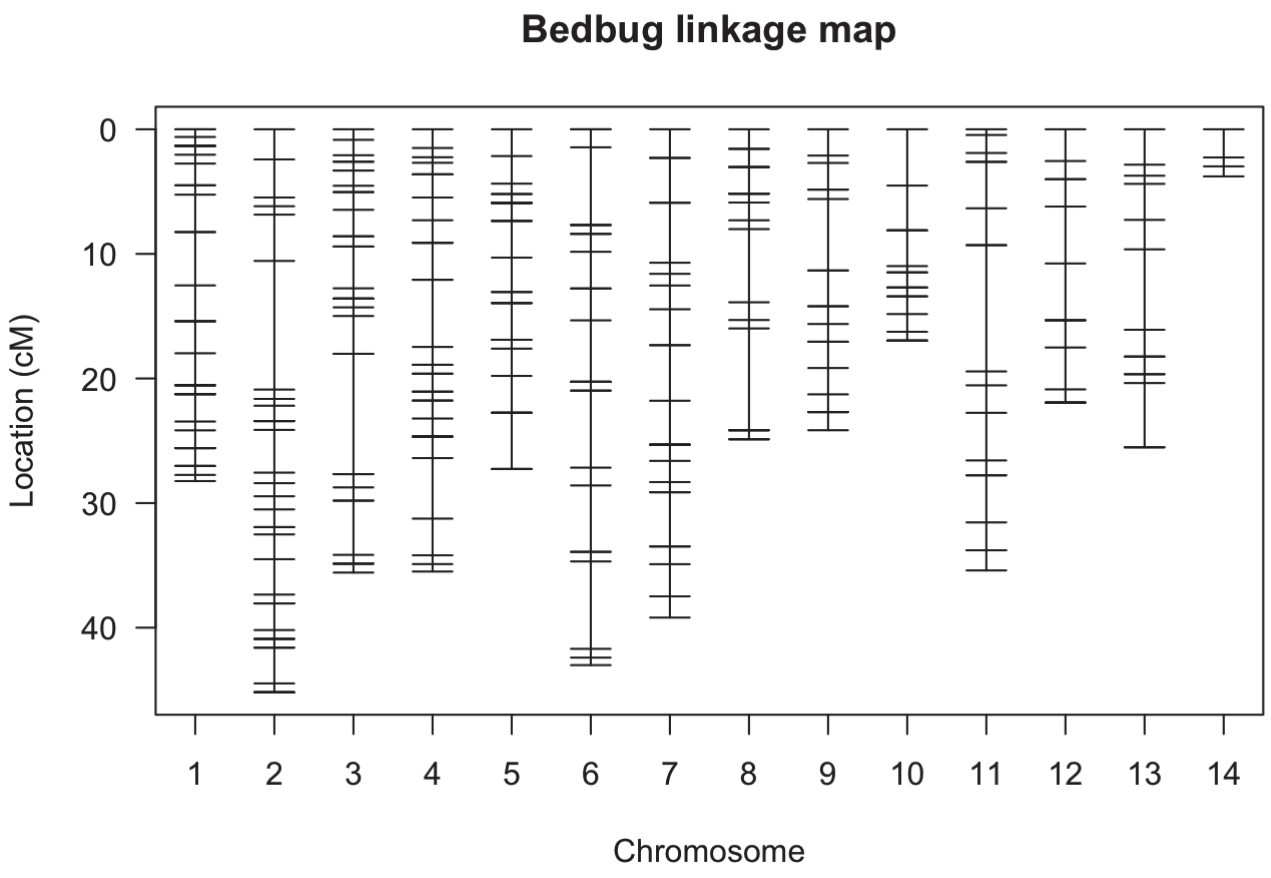
```
# examing the bedbug linkage map
summaryMap(bedbugs)
```

This returns a `data.frame` where each row is a linkage group and the four columns are: *`n.mar` - the number of markers* `length` - the length of the linkage group in centimorgans (see below) *`ave.spacing` - the average spacing between markers (in centimorgans)* `max.spacing` - the maximum spacing between markers (in centimorgans)

We can see that that the 334 markers are spread across 14 linkage groups. Spacing of markers here is shown in **centimorgans** and the total length of the linkage map is 407 cM. Recall that a single centimorgan represents the probability that of 0.01 that a recombination event in one generation. Alternatively you could think of it as one recombination event every 100 generations.

Next we can take a look at the linkage map itself.

```
plotMap(bedbugs, show.marker.names = FALSE, main = "Bedbug linkage map")
```

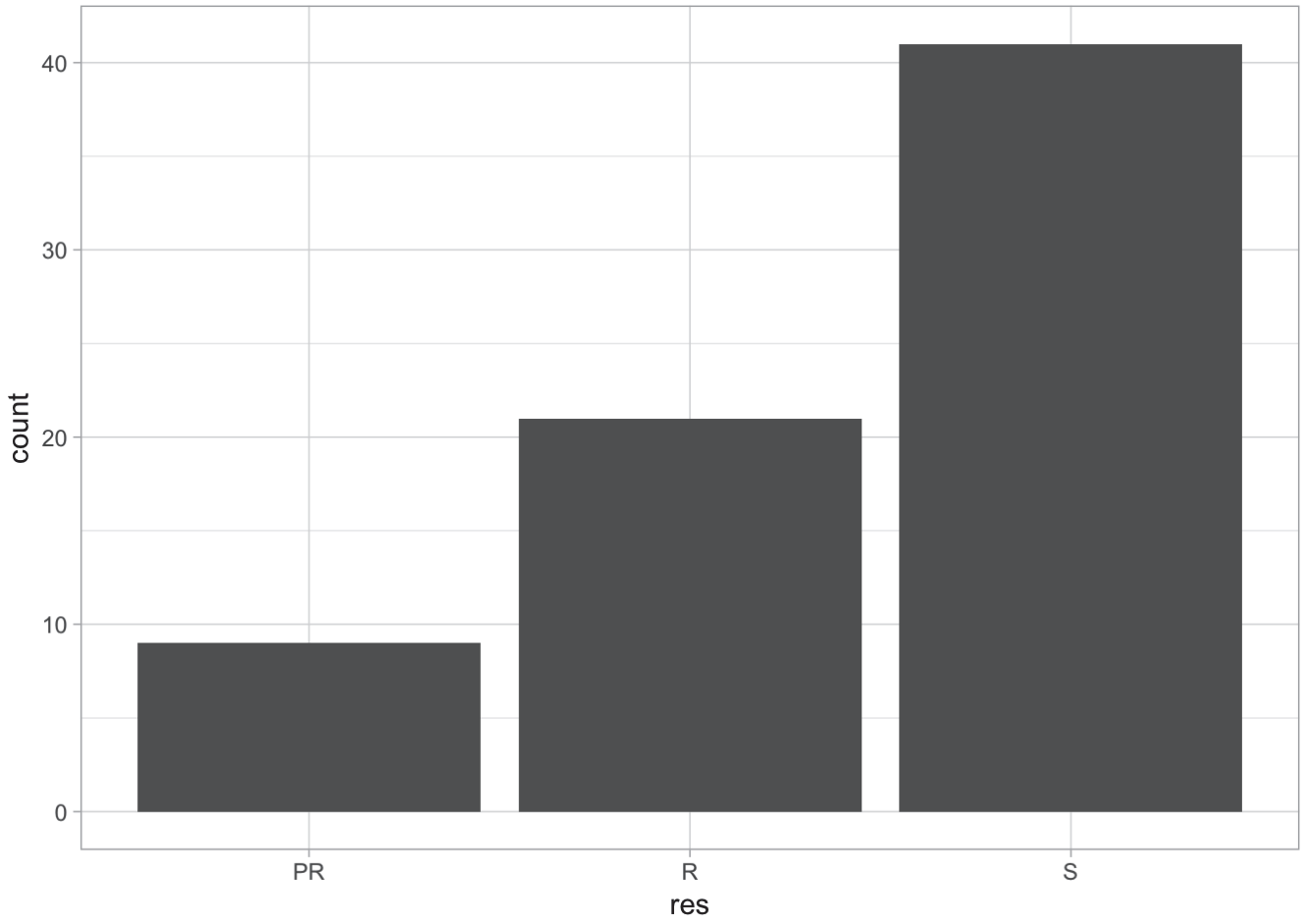


This is a visualisation of the data from the summary table - i.e. it is the spacing and distribution of markers. There are 14 groups and we see the markers laid out on each of them. The further apart the markers are, the greater the probability that there will be a recombination event between them.

Performing a QTL analysis

With the linkage map in place, we can now attempt to actually map out pesticide resistance. Let's take a look at the distribution of phenotypes.

```
# make a data.frame of the phenotype data
pheno <- as.tibble(data.frame(res))
ggplot(pheno, aes(res)) + geom_bar() + theme_light()
```



You can see from this that majority of the F2 generation were susceptible to pyrethroid and that only a small number were partially resistant. When we perform a QTL mapping experiment, we are essentially looking for the genome region that can describe the greatest percentage of the variance in this phenotypic distribution.

Before beginning mapping, we need to perform one quick fiddle with our code. `qtl` requires that the phenotype is encoded as a number, so we use the following command to change our resistance coding to numbers.

```
# convert to numeric data
bedbugs$pheno$res <- as.numeric(res)
# examine the results
bedbugs$pheno$res
```

We're now good to go! To perform the analysis, we will use the `scanone` function like so:

```
bedbugs_scan <- scanone(bedbugs, pheno.col = 2)
```

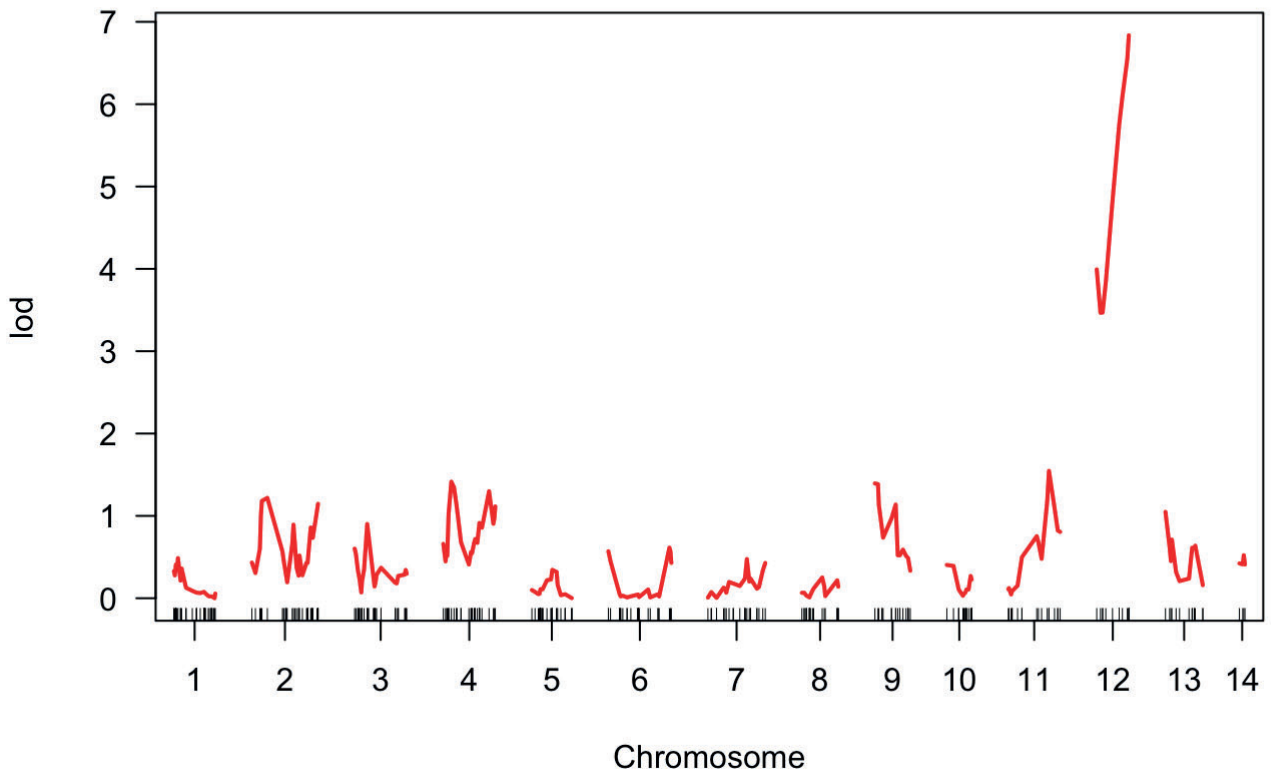
You might see a warning here, but don't worry too much about that since we are only looking at a simple example. So what exactly have we just done? Using the `scanone` function, we ran a QTL analysis across the entire set of markers and looked for an association between the genotypes at a marker and the phenotype. This is more or less an ANOVA, but we will return to that shortly. For now, let's take a closer look at the results:

```
summary(bedbugs_scan, threshold = 3)
```

Calling `summary` on the scan object shows us a single marker on linkage group 12 (referred to a chromsome here) at 21.9 cM may be a QTL. It also returns something called a `lod` - this is a LOD score and for this marker it is 6.84.

Let's plot the LOD distribution to get a better idea of what is going on.

```
plot(bedbugs_scan, col = "red")
```



You can see quite clearly that whaterver the LOD is, it is much higher on linkage group 12 and this peak focuses right where our marker is. This suggests a strong association between genotypes here and the phenotype in question.

What do we mean by a LOD score? It is beyond the scope of the tutorial to go in to too much detail about this, but LOD stands for logarithim of the odds ratio. It is essentially the ratio between a model where a QTL exists at a marker and one where there is no QTL at all. So, if a LOD is 0 or close to 0, then there is essentially no evidence a QLT is present. However, if a QTL is present and explains variance in the phenotype then the LOD score is expected to be higher, as it is at this marker.

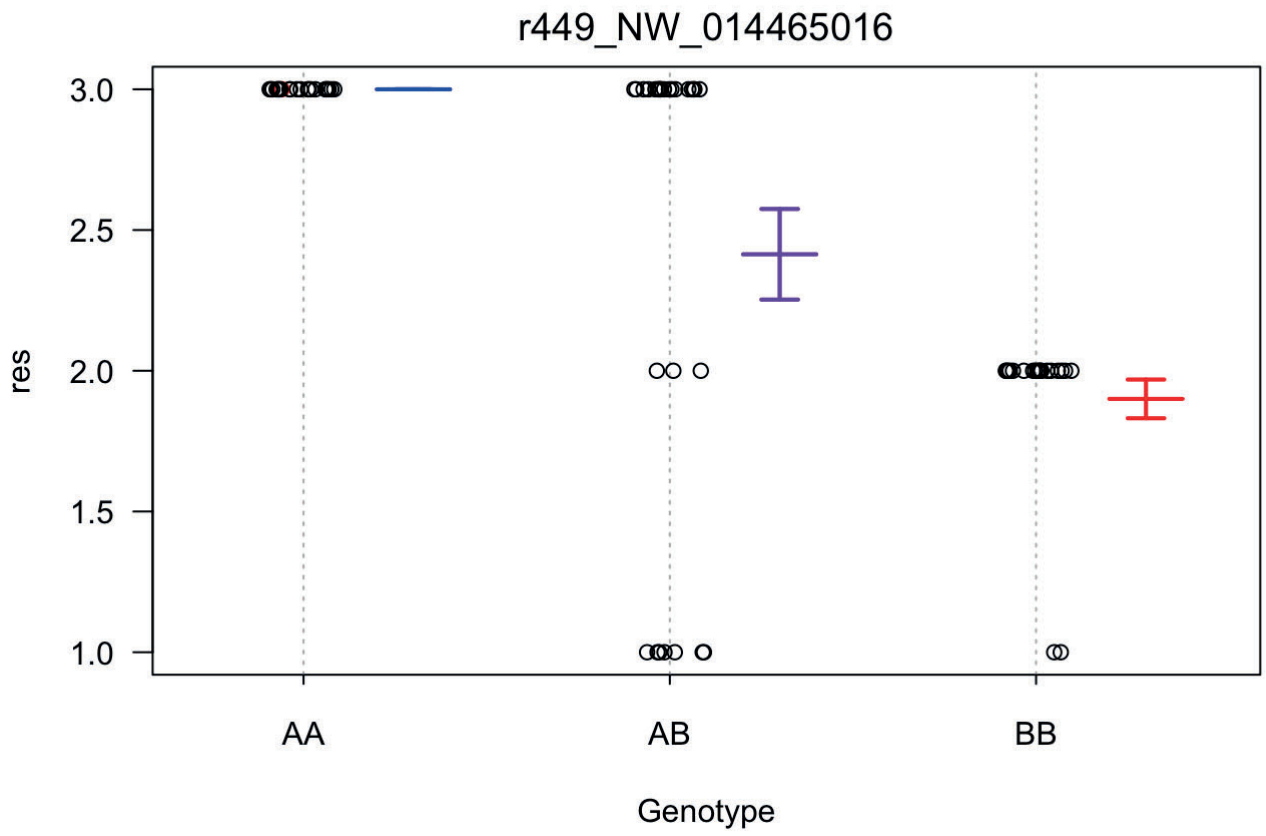
Linking ANOVA to our QTL analysis

Now that we have performed a QTL analysis and learned about ANOVA, we will try to bring the two together to demonstrate how they are closely related… as well as demonstrating the importance of statistics in evolutionary genetics!

Exploring the phenotype-genotype association

Remember earlier we learned that it is important to always visualise your data? This is true for QTL analysis too! We can do this quite esily using a function called `plotPXG` from `qtl` - i.e. where **PXG** stands for phenotype times genotype.

```
pheno_geno <- plotPXG(bedbugs, pheno.col = 2, "r449_NW_014465016")
```



Note that all we did here is specify the dataset `bedbugs` , the column the phenotype is in and the name of the marker we are interested in. However, not that this plot is slightly misleading because remember our phenotypic data is scored numerically (1 = partially resistant, 2 = resistant, 3 = susceptible). This plot is also based on the assumption the resistance trait we mapped is continous (instead of categorical) so it plots the trait means (coloured crosses).

A better way to summarise the data is to see it in a table. The following code looks complex but all it does is get the information the way we need it to be.

```
# pheno - 1 is partially resistant, 2 is resistant, 3 is susceptible
phenotype <- factor(pheno_geno$pheno, labels = c("partial resistance", "resistant", "suscepti
ble"))
# geno - 1 is AA, 2 is AB, 3 is BB
qtl_marker <- factor(pheno_geno$r449_NW_014465016, labels = c("AA", "AB", "BB"))
# make into a tibble
qtl_df <- as.tibble(data.frame(phenotype, qtl_marker))
```

All we did here is extract the phenotype information and also the marker information for the QTL and used the function `factor` to turn them in to `factor` variables - where there is a label for each category of the data (see here for a reminder of what a factor is (https://evolutionarygenetics.github.io/Introduction.html)). We then made everything into a `tibble` for later.

So we want to make a table where we can see the numbers of each resistance phenotype for each genotype. That's very easy!

```
# summary table
table(qtl_df)
```

It is fairly clear from this table and the figure if a bedbug has an A allele at this locus, it is more likely to be susceptible to the pesticide. In contrast, most BB individuals are resistant.

Testing the same genotype phenotype association with ANOVA

Last but not least. We can test the same association with ANOVA. To do this, we are going to read in some data we prepared for you earlier (https://evolutionarygenetics.github.io/qtl_df.tsv).

```
qtl_df2 <- as.tibble(read.delim("./qtl_df.tsv"))
```

Feel free to take a look at this - it is essentially the same data as before, but this time there is also the genotype data for a marker which is not a QTL.

Let's use ANOVA to test whether variation in genotype can explain the variation in phenotype.

```
model1 <- aov(as.numeric(phenotype) ~ qtl_marker, data = qtl_df2)
```

This is exactly same as the ANOVA we ran earlier with one exception. Like with `scanone` when we did our QTL analysis, `aov` requires that the phenotype data is stored as a `numeric` . So to deal with this we need to add the `as.numeric` function.

Finally, let's call `summary` on our model output to see whether there is a significant effect of the QTL marker on our phenotype.

```
summary(model1)
```

Quite clearly there is a significant difference between genotypes at this marker in terms of phenotype. What about at the non-qtl marker data we read in alongside this? Well for that, you can see the study questions…

Study questions

For study questions on this tutorial, download the `Chapter6_R_questions.R` from Canvas or find it here (https://evolutionarygenetics.github.io/Chapter6_R_questions.R).