

FISH 102

Tasks

In this practical you will learn how to do the following in R:

- Length distributions
- Age-length tables
- Age-length keys
- Age-length distributions
- Proportion at age
- Mean length at age
- Weight-length relationship
- Catch in numbers at age
- Summary plots

Length and age data

With length and age data it is important to be able to know how to create length distributions, age-length distributions etc It is also important that you practise looking at the data – don't just type in the commands, look at what they do. Also, don't just do the plots I ask for, try other plots as well.

The most important R commands used here aggregate data and use the dimension names of objects. There are examples of all of these in the Introduction to R.

For some commands I have given you an example with a small dataset to try first. When you are using a new command, or trying to do something complex it can be helpful to first use a small dataset where you know what the answer should be or where it is easy to see exactly what has been done.

Some of this may look quite complicated – it just takes longer to explain than do. It's important, however, that you use a text editor to keep a record of your commands so you can easily replicate the process later.

The data are available at:

www.hafro.is/~lorna/age
www.hafro.is/~lorna/length

- Copy the data into a text editor – remember it is easier if you run R in the same directory as the file you are reading in.
- Read the data into R using `read.table` (eg `adata <- read.table("age", header=T)`) which creates a `data.frame`. If R is not running in the same directory you will need to provide the path of the file.

The columns are:

station	a unique number for each haul/tow
rectangle	position: each rectangle is one degree longitude and half a degree latitude
species	code 1 is cod
le	length in cm
age	age in years
wgt	weight in grams
number	the number at that length OR the number at that age and length

The age data are a subset of the length data, so all lengths in the age table should also be in the length table.

Length distributions – from a vector of length data

With the age data we have only one fish for each row. With data in a format like this a length distribution can be created using `table`.

- Example: Given a vector of length data: `x = sample(5:12, 20, replace=T)`
`sort(x)`
`table(x)`
`hist(x)`
- Create a vector containing the length data from the age file. (eg `le <- age$le`)
- Plot it and tabulate it (as for the example). `table` calculates the length distribution.

As these data are sparse – only zero or one fish for most lengths – it is maybe better to aggregate.

- Group into 5cm length classes. `le5 <- floor(le/5)*5+2.5` What does the 2.5 at the end do?
- How would you do this for 10cm groups?
- Plot histograms to look at the data.
- To create a length distribution you need to count the number of fish in each length group. Create the length distribution.

As the length distribution should be in even groups we have some missing. With a large dataset this is less likely to be a problem but there still might be gaps at the ends of the distribution.

The gaps in a vector can be filled in as follows:

- Create a vector the length the length distribution should be and name the cells of the vector. eg
`rnames <- seq(min(le5), max(le5), 5)`
`ldist <- rep(0, length(rnames))`
`names(ldist) <- paste(rnames)`
- Then, calculate the length distribution from the data and put it into the empty vector. This substitutes the elements of `tmp` into `ldist` where `names(tmp) = names(ldist)`.
`tmp <- table(le5)`
`id <- names(tmp)`
`ldist[id] <- tmp`
- This can be done in fewer lines by combining commands:
`rnames <- seq(min(le5), max(le5), 5)`
`ldist <- rep(0, length(rnames))`
`names(ldist) <- paste(rnames)`
`tmp <- table(le5)`
`ldist[names(tmp)] <- tmp`
- Plot the length distribution.

```
Line plot: plot(as.numeric(names(ldist)), ldist, type="l")
```

```
Barplot: barplot(ldist)
```

- Calculate and plot the length distribution as proportions – the number in each length class divided by the total number. (Use `sum` to calculate the total.)
- Practise calculating length distributions with different levels of aggregation.

Length distributions – from a matrix of length and count data

With the data in the length file we can have more than one fish per row which means we cannot use `table` and use `tapply` instead.

- Example: Create a small dataset `temp` and use `tapply`. Then plot the length distribution.

```
le <- c(seq(20,40, 5), 25, 35)
num <- c(4,7,9,6,2, 5,3)
temp <- data.frame(le = le, num = num)
temp.tab <- tapply(temp$num, temp$le, sum)
```
- To create a length distribution from the `~lorna/length` data you need to sum the number of fish in each length group. (eg `l.dist <- tapply(ldata$number, ldata$le, sum)`)
- Plot the data (`barplot`). Do these data need to be aggregated into wider length groups?
- If you want to aggregate the length data you can store them as another column in the data.frame. eg `ldata$le5 <- floor(ldata$le/5)*5+2.5`
- Do you have the same problem with empty length cells? If so, fix the length distribution.
- Plot the length distribution as before.
- Calculate and plot the length distribution as proportions.

Age distributions

- Create a barplot of the age distribution using the methods you have learned for length distributions.
- Calculate the proportion at age.

Age-length tables

Using the age data we can tabulate using 2 columns to create an age-length table (ALT) a 2 dimensional table.

- Tabulate the data (eg `alt <- table(age$le2, age$age)`)
- Try with with different length aggregations.
- If you have gaps in the length cells how can you create an even table?
- This is done in a similar way as for the length distribution – but with a matrix. As there are 2 dimensions `dimnames` is used instead of `names`.
First create a matrix of zeros with `dimnames` of all possible values of age and length.

```
age.vec <- min(age$age):max(age$age)
```

```
le.vec <- seq(min(age$le5),max(age$le5),5)
alt <- matrix(rep(0, length(age.vec)*length(le.vec)), ncol=length(age.vec))
dimnames(alt) <- list(le.vec, age.vec)
```

Then calculate the age length table and substitute the age length table into the empty matrix.

```
alt.tmp <- table(age$le5, age$age)
alt[dimnames(alt.tmp)[[1]], dimnames(alt.tmp)[[2]]] <- alt.tmp
```

Example: This small example may be clearer. Here, `tmp2` is a larger matrix of zeros and the values of `tmp` are substituted into it.

```
tmp <- matrix(c(2,4,2,4,2,4), byrow=T, ncol=2)
dimnames(tmp) <- list(c(2,4,6), 1:2)
```

```
tmp2 <- matrix(rep(0,10),byrow=F, ncol=2)
dimnames(tmp2) <- list(2:6, 1:2)
```

```
tmp2[dimnames(tmp)[[1]],dimnames(tmp)[[2]]] <- tmp
```

Operations on the rows and columns of a table

The length and age distributions of the age data can be calculated directly from the age-length table.

To sum the columns (ages)
`apply(alt, 2, sum)`

To sum the rows (lengths)
`apply(alt, 1, sum)`

- Compare these with the number at length and at age you calculated from the original data.

Functions other than `sum` can be used in `apply` eg `mean`, `median`.

Age-length distributions and mean length at age

Mean length at age can be calculated directly from the age data using `tapply`.

- Calculate mean length at age from the age data.
- Calculate the corresponding variance.

In length stratified sampling schemes age samples are taken from a fixed number of fish by length group. This can cause problems in calculating mean length at age directly from the age data. Even without such a stratification scheme using the length data (which is more abundant) along with the age data provides a better estimate of mean length at age.

The ALT is used to assign an age to fish with only length data. To do this, the ALT is transformed into an age length key (ALK) i.e. each row represents the proportion of fish at each age for a length group eg the age distribution of fish of between 50 and 54 cm. To do this

each row of the ALT is divided by the total number of fish in that row ie each column of the ALT is divided by the total length distribution of the table.

The most reliable way to do this is to create a matrix of the same dimensions as the ALT.

- Calculate the length distribution of the ALT
eg `ld.tmp <- apply(alt, 1, sum)`
- Create a matrix where each column is the length distribution.
eg `ld.mat <- matrix(rep(ld.tmp, ncol(alt)), byrow=F, ncol=ncol(alt))`
- Create the ALK by ALT/LD – `round` will help if you want to look at it to fewer decimal places.
eg `alk <- alt/ld.tmp`
- As the ALT contains many zeros, these are now missing values (NA). In the ALK these should be changed back to zero. To substitute 0 for every NA:
eg `alk[is.na(alk)] <- 0.`

The ALK is then multiplied by the length distribution from the length samples. Again you need to create a matrix from the length samples. The lengths of the length distribution must match those of the ALK. To do this the `names` of the length distribution must match the length `dimnames` of the age-length key, eg `ld <- ld[dimnames(alk)[[1]]]`.

Age distribution from the ALD

- Replicate the length distribution as before to create a matrix with the correct number of columns (and rows) called here LD – the lengths of LD must match those of the ALK.
- Calculate the ALD by `ALK*LD`.
- Calculate the number at age from the ALD. (`apply`)
- Calculate the proportion at age.
- How does the proportion at age from the ALD compare with that from the age data alone?

Mean length at age from the ALD

To calculate mean length at age from the ALD. The ALD is a matrix of the number of fish in each age-length cell.

- Multiply the ALD by the length of each cell: the lengths are `as.numeric(dimnames(ald)[[1]])` by:
 replicate the lengths to create a matrix.
 multiply ALD by the length matrix to create a new matrix ALD2
- Sum the columns of ALD2 to calculate the total length of all fish by age. Divide this by the total number at age.

How do the calculations of mean length at age directly from the age data and from the ALD compare?

Sourcing a file

Write a file to do all the necessary steps and run the file in R using eg `source("calc.r")`.

The steps are:

```
Read in length data
Read in age data
Calculate the length distribution from the length data.
Calculate the age-length table from the age data.
Calculate the age-length key.
Calculate the age-length distribution.
Calculate mean length at age.
```

To write to the screen from a sourced file it is necessary to use `print(objectname)` . Normally in R you have just typed `objectname`.

Changing the range of age and length data

In the method described above, the length distribution was reduced to the dimensions of the ALK. An alternative method is to calculate the length distribution only from lengths in the same range as the age data. As the age data are a subset of the length data there will often be smaller or large fish in the length than age data.

The length data can be modified by deleting values outside the range of the age data or by changing the lengths to the smallest/largest in the age sample. The command `ifelse` can be used in either case.

Some examples of `ifelse` are:

```
x <- 1:10
ifelse(x<5, 0, x)
if x < 5 then x = 0, else x = x ifelse(x==5, 0, x)
```

To use `ifelse` to delete certain values the values outside the length range are changed to `NA` which means missing and then deleted, eg:

```
mina <- min(adata$le)
maxa <- min(adata$le)
ldata$le <- ifelse(ldata$le<mina, NA, ldata$le)
ldata$le <- ifelse(ldata$le>maxa, NA, ldata$le)
ldata <- ldata[is.na(ldata$le),]
```

Alternatively, `ifelse` can be used to change the values to the minimum or maximum in the age sample, eg

```
mina <- min(adata$le)
maxa <- min(adata$le)
ldata$le <- ifelse(ldata$le<mina, mina, ldata$le)
ldata$le <- ifelse(ldata$le>maxa, maxa, ldata$le)
```

Length–weight relationship

The age data in www.hafro.is/~lorna/age contains length and weight data.

The length–weight relationship is typically written as:

$$W = a \times L^b$$

which can be rewritten as:

$$\ln W = \ln a + b \times \ln L$$

- Plot weight against length.
- Do any of these data look wrong? Outlying points can be removed as follows:

```
plot(data$x, data$y)
identify(data$x,data$y)
```

click on the ‘bad’ point with the left mouse button - it will be numbered

if you want to identify more points continue to click with the left and when you want to finish click with the right.

the number of all the identified plots will then be in your R window. You could also have done:

```
id <- identify(data$x,data$y)
```

- To remove the rows of data containing the outlying data points and create a new dataset:

```
data2 <- data[-id,]
```

If you didn’t put the points into an object you can still do this using the information printed into the R window, e.g

```
data2 <- data[-c(3,6),]
```
- Plot the log transformed data (natural logarithm). Do you have any outliers? Relabel the x and y axes with appropriate text.
- Fit a linear regression model to the log transformed data. (eg

```
fit <- lm(log(y) ~ log(x))
```

)
- Using `fit$coef` what are the values of a and b in the first equation above?
- Use `summary(fit)` to look more at the fitted model.
- Plot model diagnostic plots using:

```
par(mfrow=c(2,2))
plot(fit)
```

Catch at age

Catch at age can be calculated using a combination of length data, the weight-length relationship, proportion at age and total catch.

- Using the length data from the length file (the age data are only a subset of it) and the weight-length relationship, what is the mean weight of fish in the dataset? eg

```
mean(a*le^b)
```

The fish were weighed in grammes so convert this into kg.

- If the total catch is 1000 kg how many fish were caught?
- Using the calculated proportion at age, what is the catch in numbers at age? (eg $\text{agep} \times \text{num}$)

Plotting

Use boxplots to summarise the datasets. eg

Compare length distribution by rectangle.

Compare length distribution by age.

Compare number of fish by rectangle.

```
boxplot(le ~ rectangle, data = ldata)
```

```
boxplot(le ~ age, data = adata) boxplot(num ~ rectangle, data = ldata)
```

Try some other plots.