Tutorial #1 - Introduction to Linear Modelling

Guy F. Sutton

05/05/2020

What is a linear regression?

A linear regression model is one of the most basic models used to analyse ecological data. But, what makes something a linear regression model?

1. Linear - Our predictor (independent) variable shows a linear relationship with our response (dependent) variable.
2. Regression - We are measuring the response between one (or more) predictor variables and a response variable.

When we model one numeric predictor and a numeric response variable, we have a *simple linear regression (SLR)*. When we have more than one numeric predictors and a numeric response variable, we have a *multiple regression*. Today, we are going to focus on simple linear regressions in R.

So, when we model Y as a linear function of X, we are performing an SLR.

What are the assumptions of a SLR?

There is a nice acronym which makes this easy to remember: **LINE**.

1. Linear - The relationship between your predictor and response is linear.
2. Independent - The errors (residuals) are independent (no autocorrelation, pseudo-replication, ect…).
3. Normal - The errors (residuals) are normally distributed. *NB! - SLR does not assume your raw data are normally distributed, just the errors*
4. Equality of variance (homogeneity) - At each value of your predictor variable, the variance in your response variable is equal.

How do we run a simple linear regression in R?

R has a built-in function called `lm` that is the workhorse used to fit linear models. You can find more information on this function by typing `?lm` into your console.

The general formula for running a linear regression in R is:

```
model_name <- lm(response ~ predictor, data = data_frame_name)
```

Let's break this down:

1. We are telling `R` to perform a linear regression by using the `lm` function.
2. We are going to store/save our linear regression in a variable called `model_name` using the <- (assign) key.
3. We would like to model our `response` variable as a linear function of the `predictor` variable.
4. The `data = ...` argument tells `R` where to look for your data.

Running your first linear regression in R

In this next section, we are going to run our first simple linear regression in R. We will need to load in our data, check the data imported correctly, and then we can proceed.

I have made up some fake data simulating an experiment where we reared female insects at 4 different temperatures, and measured the number of larvae they produced (fecundity). We also measured the body mass of the females (in grams) at the start of the experiment, because we suspect that bigger females may produce more larvae than smaller females.

```
# Check data imported properly
head(data)
```

```
## # A tibble: 6 x 3
##    temp adult_mass larvae
##   <dbl>      <dbl>  <dbl>
## 1    15          3      0
## 2    15          5     17
## 3    15          4      6
## 4    15          6     11
## 5    15          3     11
## 6    15          4      5
```

Our dataset called `data` has 3 columns.

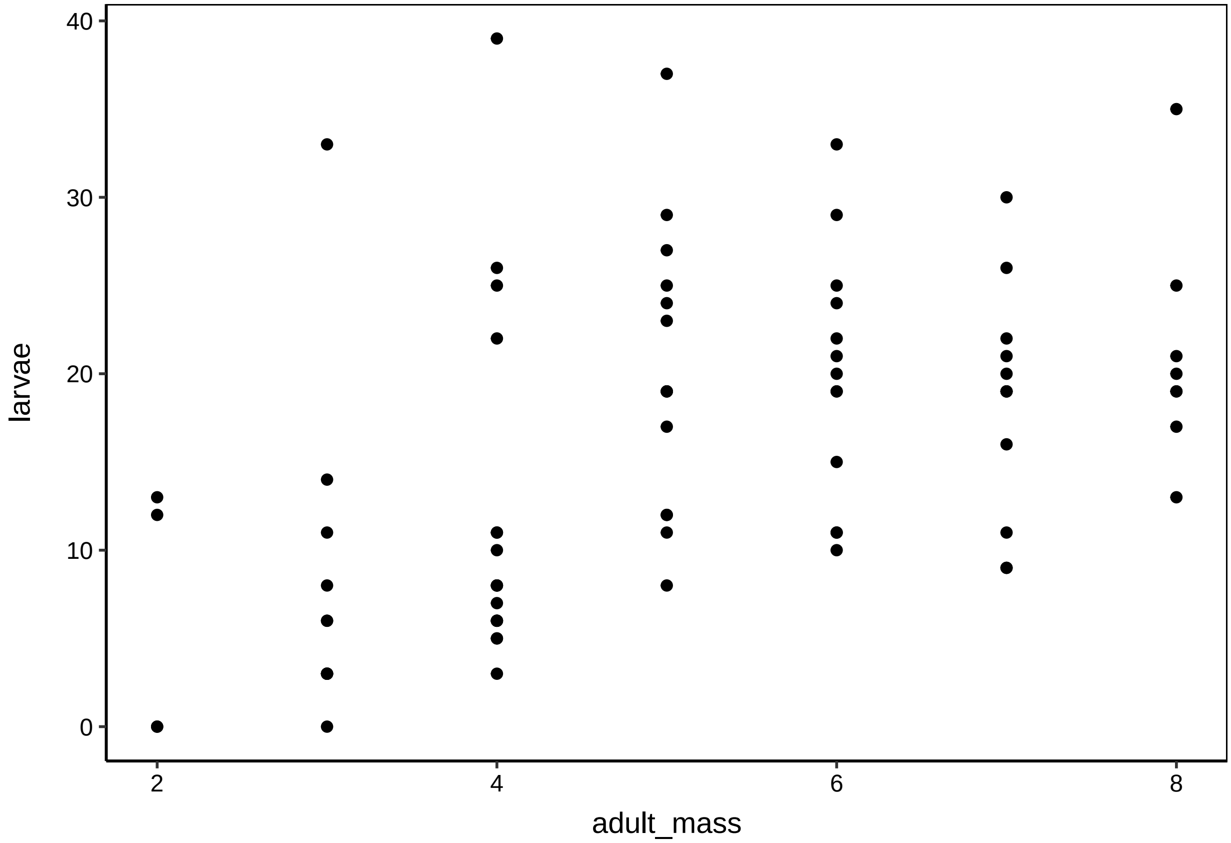
1. `temp` = temperature in degrees celcius.
2. `adult_mass` = adult body mass at the start of the experiment (in grams).
3. `larvae` = no. of larvae she produces during experiment.

Step 1: Define your research question or hypothesis

Let's say we were testing: do larger female produce more larvae? * Thus, `larvae` will be our response variable, and `adult_mass` will be our predictor variable.

Step 2: Visualise your data and their relationships

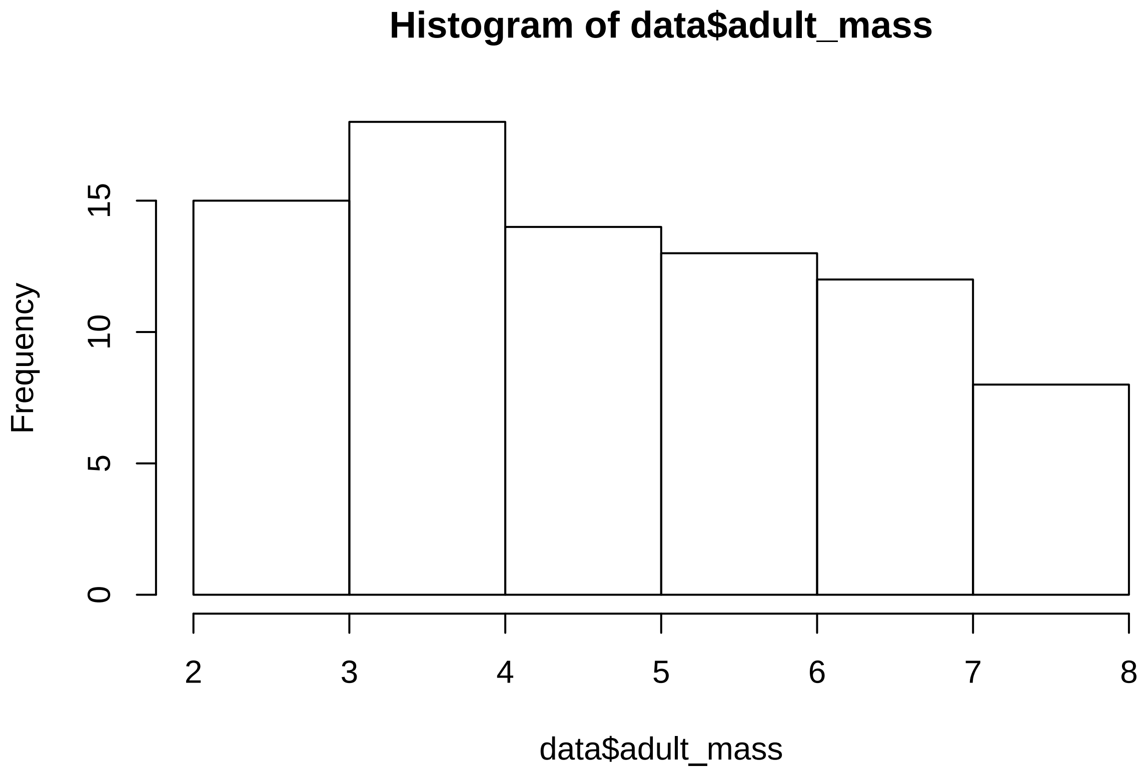
```
# Firstly, visualise relationship between predictor and response
ggplot(data = data, aes(x = adult_mass,
                        y = larvae)) +
  geom_point()
```



Remember, to run a simple linear regression, one of our primary assumptions is that the relationship between our predictor and our response is linear. Looking at the graph, we can see what looks like a linear relationship.

HINT: This is a good time for you to record any noticable features about your data. For example, are there any outliers? Is the relationship positive or negative? How strong is the relationship? Is there a lot of variation in the data? Now, we should take a closer look at potential irregularitites in the response and predictor variables.

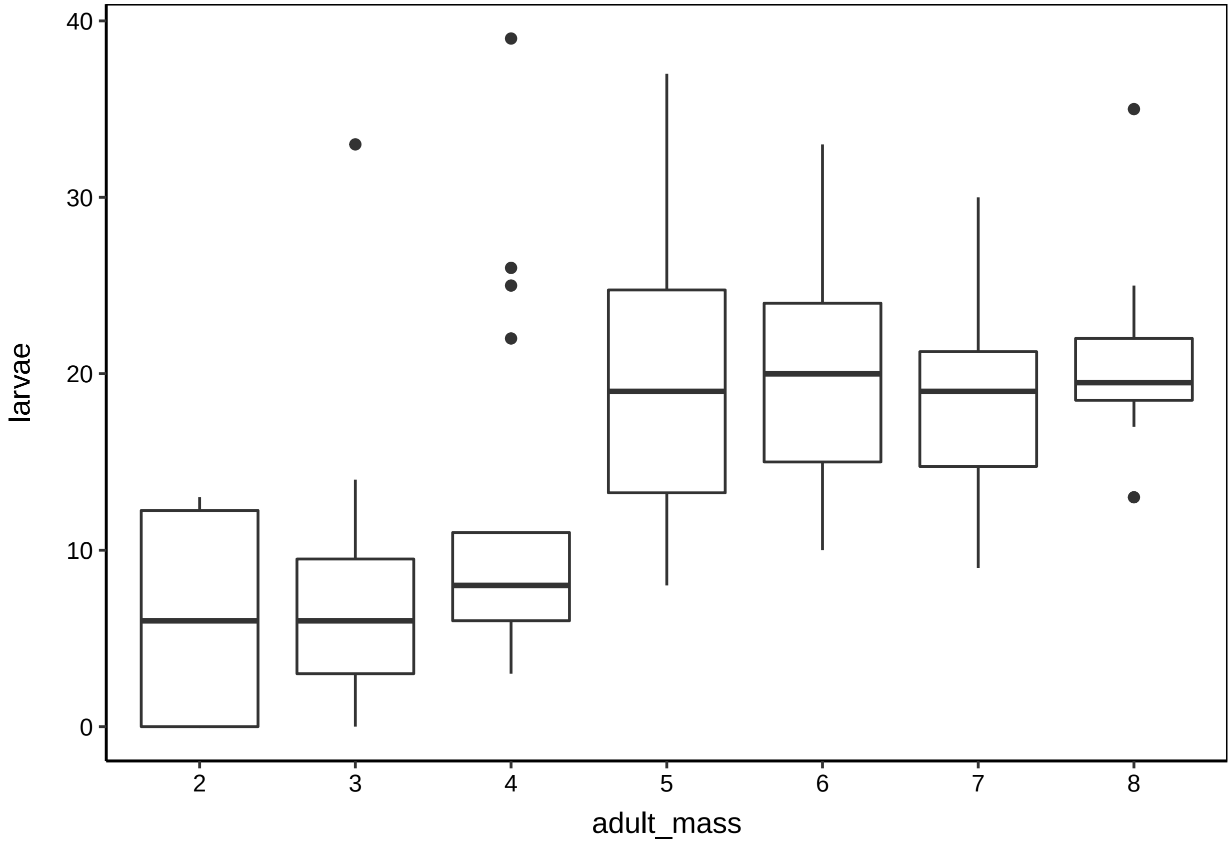
```
# What is the distribution of our predictor variable?
hist(data$adult_mass)
```



We see a pretty flat, normal distribution of female body weights. Make sure there are no gaps in the bars (that would indicate a potential outlier), or any bars that are significantly taller than others.

```
# Are there outliers in the response variable?
# Is there more variation in your response variable at some x-values than others?
# Firstly, we must make the x-variable into a factor, just to make this graph.
raw_data <- data %>%
  mutate(adult_mass = as.factor(adult_mass))

# Make the plot
ggplot(data = raw_data, aes(x = adult_mass,
                            y = larvae)) +
  geom_boxplot()
```



Here, we can see quite a few outliers in the response variable (`larvae`), as indicated by the 7 black, filled circles. That is a little concerning.

What should be even more worrying is that the variation in the number of larvae produced for females of certain `adult_mass` values (e.g. 3 and 4g) is much more than at other values (e.g. 2, 6 and 7g). This is a good indication that a your data will not meet the requirements of homogeneity of variance, but more on that later.

Step 3: Run your linear model

Now, we are going to finally run our first linear model.

Here, we specify our response variable `larvae` to the left of the `~` and our predictor variable `adult_mass` to the right of the `~`. We also have to remember to tell `R` to look for this data in the dataframe called `data`. We will store our linear model in a variable called `mod1`. Later, we will look at the output of our linear model by telling `R` to show us what is stored in `mod1`.

```
# Specify a linear model
# - We use the 'lm' function to tell R to run a linear model
# - We then specificy our response var (larvae),
# - We then specificy our predictor var (adult_mass),
#   (predictors go to the right of ~ sign).
# - We also have to specify where R must look for this data
#   using the 'data = ...' argument
mod1 <- lm(larvae ~ adult_mass,
           data = data)
```