Lab 5: Linear Regression

Entomology 5126 - Spring 2026

Jeremy Hemberger and Brian Aukema

February 20, 2026

Table of contents

1 Introduction

Today's lab focuses on simple linear regression. ANOVA is used to examine factors; regression is used for continuous variables. Once you have both techniques in your toolbox, you will be able to do almost anything! This includes combining both factors and continuous variables in the same model (Analysis of Covariance, or ANCOVA), and changing assumptions about the distribution of the data (e.g., generalized linear models). This basic ANOVA and regression framework will serve us well as we move into more advanced topics in spatial and temporal analyses through the month of March.

2 Regression

The basic formula for a regression model is:

$$y_i = \beta_0 + \beta_1 x_i + \epsilon_i$$

The β_0 is the intercept, or where the line crosses the y-axis. The β_1 is the slope, or the increase per unit change in x. The ϵ_i refers to the "errors" about the line. We assume that the errors are normally distributed N(0, σ^2). If the model was perfect (never the case), there would be no errors and your data would lie along a perfectly straight line.
There are four assumptions for a linear regression model:

1. The model is correct
2. The observations are independent
3. The variances are equal
4. The errors are normally distributed

As you might expect, you can check the assumptions of equal variances and normally distributed errors in your residual plot. Actually, in regression, residual plots can also help detect when your model might not be correct (such as if you need a polynomial model or another technique to fit a curve). Although today's data is *very* close to needing curve-fitting techniques, as you will see from the residual plots, the data are actually statistically best fit with a straight line. We will focus on simple linear regression for today's assignment. The goal for today is to become comfortable with fitting and evaluating simple linear regression models.

2.1 Useful commands for fitting regression models

- The linear model command, `lm()`. Here are the steps:

  1. Decide what to call your model. E.g.,
     `> fm1 <-`
  2. Specify the `lm()` command, and determine what `y` variable you are analyzing. It should be something that is, or can be transformed to be, normally distributed.
     `> fm1 <- lm(y~`
  3. List the continuous variable(s) that you are analyzing. For example, if you examining the effect of precipitation on a response variable `y`, you might write
     `> fm1 <- lm(y~precipitation`
  4. Specify the dataset you are using.
     `> fm1 <- lm(y~precipitation, data=mylabdata)`

  Hey! This looks very similar to fitting an ANOVA (which is also a linear model)!

- The `summary()` function. Examine the F-statistic of the overall model, which you can find at the bottom of the summary. This tells you whether there is *any* signficant relationship between your response variable and your predictor. (Alternatively, you can also recover the same F-statistic from the `anova()` command, because even a regression can be analysed using an ANOVA framework. There just aren't as many replicates in each "group"). If the p-value is greater than 0.05, then there is nothing significant. Go have lunch and design another experiment.
- Remember, while the p-value from the overall model is good to examine, you should also simultaneously examine how well the model fits and if a transformation might be necessary:
  `> plot(residuals(fm1)~fitted.values(fm1))`
- Once you have a good model, to get detailed information on model coefficients:
  `> summary(fm1)`
- Remember, a residual plot should show roughly equal variance around the horizonal zero line across the plot. If the spread is like a fan, for example, you may need to apply a transformation. You can do this right in your model fitting command:
  `> fm1 <- lm(sqrt(y)~precipitation, data=mylabdata)`

3 Today's data

Male crickets chirp by raising their wing covers to 45 degrees and rubbing the front area of one wing cover (scraper) against the rough area on the other wing (file). The faster this stridulation, the higher the chirping pitch that is produced. Also, it is thought that stridulation rate increases with air temperature. You may even remember equations from elementary school providing instruction on how to tell the ambient air temperature on a summer evening from the number of chirps over a one minute period.
Today's data set, `Lab5data.xls`, comes from *The Song of Insects*, written in 1948 by Harvard physics professor George W. Pierce. It is a very small dataset: 15 observations of striped ground crickets (*Allonemobius fasciatus*) chirping at different ambient temperatures. There are only two variables. `Pitch` is the average number of vibrations per second. (I *think* that there are about 9-10 vibrations in an auditory chirp). `Temperature` is the ambient air temperature in degrees Farenheit.

4 Today's Assignment

1. You know what to do for question one. For regression, we want to work with explanatory variables that are *not* factors.

2. Check the data graphically with a scatterplot matrix (by now, of course, this is second nature). What sort of correlation is displayed (positive or negative)? Although this is not always done, you can determine the linear correlation using the `cor()` function, specifying the two variables separated by a comma. If you were to fit a regression, would you expect a postive or negative slope?

3. Fit a linear regression of pitch as a function of ambient air temperature. Examine and comment on the residual plot. In residual plots from ANOVAs, there was a vertical line for each group. Why are there no vertical lines for regression models?

4. If you were going to publish this data, would you use a transformation of the `y` data? Why or why not?

5. For the rest of the assignment, proceed without transforming your data. Examine the `summary()` of your model. What is the relationship between the correlation above in question 2 and the `Multiple R-Squared` in the `summary()` output?

6. Continue examining the `summary()` of your model. The `(Intercept)` is just that: where the line crosses the y-axis, or what the y-value would be at $x = 0$. What does the t-test associated with the `(Intercept)` test? What does the p value tell us in this case (statistically, and ecologically)?

7. The `temperature` estimate in the `summary()` output is the slope coefficient. You will also see, of course, an estimate of the standard error and a t-value. What does this t-test test?

8. Of course, we could analyse this entire experiment by partitioning the variation by sums of squares, as we would do in an ANOVA, and derive an F-test that would test whether temperature has any explanatory utility. What is the relationship between the t-value and the overall F-ratio in the `summary()` output?

9. I found that the term for temperature is significant in explaining the average number of vibrations per second (pitch). For each one degree increase in temperature, the number of vibrations increases 0.21193. Needless to say, this is a bit awkward (e.g., for every 5 degree increase in air temperature, the number of vibrations increases by 1.05965). It might make more sense to round this number to exactly 0.2, so we could say that for every increase of 5 degrees, the number of vibrations increases by 1.
   Test whether the slope is significantly different than 0.2. The numbers from the summary output may be helpful in constructing the test. Another useful command is the `pt()` function, which generates the probability from a t distribution. The syntax is `pt(t-statistic value, df, lower.tail=T)`.

Note: The **second** argument in the `pt()` command is `df` or degrees of freedom associated with the test. While we learned *n-1* in class for a one sample *t*-test, here (and in the future), we need to reduce the degrees of freedom to the amount of information left accounting for any estimated parameters (i.e., one slope uses one degree of freedom). This just means that by convention *we use the degrees of freedom of the residual or error term for the hypothesis test.* The **third** argument in the `pt()` command returns either the upper or lower tail of the *t* distribution. I can never remember whether `T` (true) or `F` (false) is the upper or lower tail. But, you should be able to figure that out simply from the probability returned. If you are unsure, scratch a diagram of the critical *t* statistic distribution on scrap paper.

10. How many vibrations per second would I expect at 86F?
11. Over what temperature range would you feel comfortable using this model?