Lab2 Tutorial: Linear Regression

Daniel Hocking

2018-09-13

Lab adapted from Linear Regression I by Dr. Nicholas Nagel at the University of Tennessee

Introduction

In this lab, we will get data and do a few few linear regressions using various datasets.

Another thing that you'll see in today's lab is how to merge two different data sets together. This is an extraordinarily valuable skill. Rarely will all of your data be in one dataset. I find that this is more so for geographers than for other disciplines.

There are two common ways to do a linear regression in R, depending on whether you just want the intercept and slope, or whether you want to plot it. The command to run a regression and get the intercept and slope is `lm`. The command to show a line in ggplot is `geom_smooth(method='lm')`. Before we get to doing regressions, we need to look into R packages including `ggplot2` with special funtions that make life easier.

R libraries

There must be tens of thousands of different functions in R now, covering over a hundred different fields. More than you could possibly ever use. To keep all this straight, there is "base" R, and then over a thousand different add-on packages. Most people will only ever use a few.

The package we will use the most is called `ggplot2.` Another package that we will use in this lab is called `lme4.`

To load packages, use the `library` function.

Below is an example of a code "chunk" in Rmarkdown. Try executing this chunk by either (1) clicking the *Run* button within the chunk (little green arrow on the right), (2) Selecting Run on the top menu, or (3) by placing your cursor inside the chunk and pressing *Cmd+Shift+Enter*.

```
library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 3.3.2
```

On a classroom comupter this should work because the packages are already installed. If you're on your personal computer, chances are high that you get the message, `Error in library(ggplot2) : there is no package called 'ggplot2'`.

This means that the package `ggplot2` is not in your computer's library. You may have to install it first. This is how to install it and lme4.

```
install.packages('lme4')
install.packages('ggplot2')
```

However, on a university computer, you likely will not have permission to install packages. Everything we need for this course should be already installed for you. After they are installed, you can go back and load the libraries with the command:

```
library(lme4)
```

```
## Loading required package: Matrix
## Warning: package 'Matrix' was built under R version 3.3.2
```

```
library(ggplot2)
library(dplyr)
```

```
## Warning: package 'dplyr' was built under R version 3.3.2
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

You only need to install a package once. But everytime you fire up R, you will have to use the library command to load the package into R so that you can use it. Think of R as a desk in your bedroom where you do your studying. Before you can study, you will need to go to your `library` and get the books you need (you have your own library at home, right?). And if the library doesn't have the books, then you will have to ask the library to go out and get the books (with `install.packages`).

Please note, while you can run R without internet access, you will need to be online when you run install.packages. **Pro Tip:** If you are taking a plane trip and expect to get work done, then always make sure that you have installed the packages you will need before you take off.

Getting the data

Data are usually obtained from a file that someone gives you, or that you make yourself, or that you download from the web. Some data come installed with base R and others come as example datasets with particular packages. Download the `Sparrows.txt` file from Canvas and put it in a single folder containing your R Project and Rmd file (if can be in a subfolder within that folder such as `Data/Sparrows.txt`). Use the following code to read in the text file and look at the file structure. You can also click it in your "Environment" to the right in to open it with the RStudio Viewer.

```
sparrows <- read.table("Data/Sparrows.txt", sep = "", header = TRUE, stringsAsFactors = FALSE)
str(sparrows)
```

```
## 'data.frame':    1281 obs. of  12 variables:
##  $ Speciescode: int  1 1 1 1 1 1 1 1 1 1 ...
##  $ Sex        : int  0 0 0 0 0 0 5 0 0 5 ...
##  $ SexNew     : int  1 1 1 1 1 1 6 1 1 6 ...
##  $ wingcrd    : num  59 55 53.5 55 52.5 57.5 53 55 55 55.5 ...
##  $ flatwing   : num  60 56 54.5 56 53.5 59 54 56 56 56.5 ...
##  $ tarsus     : num  22.3 19.7 20.8 20.3 20.8 21.5 20.6 21.5 20.8 20.5 ...
##  $ head       : num  31.2 30.4 30.6 30.3 30.3 30.8 32.5 31.2 31.6 31.4 ...
##  $ culmen     : num  12.3 12.1 12.8 11.9 12.6 12 13.5 12.3 13.2 13.2 ...
##  $ nalospi    : num  13 8.3 8.6 8.7 8.8 8.1 10.7 8.7 9.1 10.5 ...
##  $ wt         : num  9.5 13.8 14.8 15.2 15.5 15.6 15.6 15.7 15.7 15.7 ...
##  $ observer   : int  2 8 7 3 3 2 3 5 2 6 ...
##  $ Age        : int  2 2 2 1 1 2 1 1 1 1 ...
```

Let's also look at the first 10 rows of data

```
head(sparrows, 10)
```

```
##    Speciescode Sex SexNew wingcrd flatwing tarsus head culmen nalospi   wt
## 1            1   0      1    59.0     60.0   22.3 31.2   12.3    13.0  9.5
## 2            1   0      1    55.0     56.0   19.7 30.4   12.1     8.3 13.8
## 3            1   0      1    53.5     54.5   20.8 30.6   12.8     8.6 14.8
## 4            1   0      1    55.0     56.0   20.3 30.3   11.9     8.7 15.2
## 5            1   0      1    52.5     53.5   20.8 30.3   12.6     8.8 15.5
## 6            1   0      1    57.5     59.0   21.5 30.8   12.0     8.1 15.6
## 7            1   5      6    53.0     54.0   20.6 32.5   13.5    10.7 15.6
## 8            1   0      1    55.0     56.0   21.5 31.2   12.3     8.7 15.7
## 9            1   0      1    55.0     56.0   20.8 31.6   13.2     9.1 15.7
## 10           1   5      6    55.5     56.5   20.5 31.4   13.2    10.5 15.7
##    observer Age
## 1         2   2
## 2         8   2
## 3         7   2
## 4         3   1
## 5         3   1
## 6         2   2
## 7         3   1
## 8         5   1
## 9         2   1
## 10        6   1
```

```
# sparrows[1:10, ] # also prints the first 10 rows and each column
```

Or we can just look at the fist 12 rows of only the sex, wingcrd, and wt columns.

```
sparrows[1:12, c("Sex", "wingcrd", "wt")]
```

```
##    Sex wingcrd   wt
## 1    0    59.0  9.5
## 2    0    55.0 13.8
## 3    0    53.5 14.8
## 4    0    55.0 15.2
## 5    0    52.5 15.5
## 6    0    57.5 15.6
## 7    5    53.0 15.6
## 8    0    55.0 15.7
## 9    0    55.0 15.7
## 10   5    55.5 15.7
## 11   0    57.0 15.8
## 12   0    55.0 15.8
```

or all rows of just the wingcrd using the $ to call that column

```
sparrows$wingcrd
```

```
##   [1] 59.0 55.0 53.5 55.0 52.5 57.5 53.0 55.0 55.0 55.5 57.0 55.0 54.5
##  [14] 53.0 54.0 56.5 55.5 55.5 55.5 53.5 56.5 57.0 56.5 56.5 54.0 56.0
##  [27] 51.5 54.5 54.5 53.0 58.0 57.0 57.0 56.5 57.0 57.0 53.0 52.5 56.5
##  [40] 56.5 55.5 56.0 54.0 55.0 55.0 57.0 56.0 56.5 55.0 56.5 54.5 58.5
##  [53] 55.0 54.5 56.0 57.0 54.5 55.0 54.5 54.5 54.0 55.0 55.0 56.0 54.0
##  [66] 55.0 54.0 56.0 56.5 54.5 57.0 51.5 55.0 55.0 55.0 55.0 54.0 54.0
##  [79] 57.5 57.5 58.0 54.0 58.0 58.0 56.0 55.0 56.5 54.0 55.0 53.0 58.5
##  [92] 56.0 56.5 55.0 54.0 55.5 56.0 54.0 54.0 55.0 54.0 56.0 54.0 57.0
## [105] 54.0 54.5 57.0 57.5 55.0 54.0 56.5 57.0 54.0 56.0 55.0 57.0 56.0
## [118] 55.5 56.5 56.5 55.0 56.0 60.0 54.0 55.0 56.0 53.0 56.5 57.0 53.0
##  [131] 55.5 51.0 55.0 58.0 55.0 55.0 54.5 57.0 55.0 56.0 56.5 53.0 55.0
##  [144] 59.0 56.0 62.0 56.0 54.5 55.0 59.5 58.0 58.0 56.5 57.5 54.5 54.0
##  [157] 54.0 54.5 54.5 55.0 56.0 57.0 54.0 56.0 55.0 54.0 55.0 55.0 57.0
##  [170] 56.5 55.0 56.0 56.0 59.0 54.0 56.0 54.0 54.0 56.0 58.0 54.0 55.0
##  [183] 55.0 53.5 54.5 54.5 55.0 56.5 56.0 55.0 58.0 56.5 56.0 55.0 53.5
##  [196] 56.5 56.5 56.0 54.0 55.0 56.5 58.0 55.0 58.0 56.5 57.0 53.0 55.0
##  [209] 55.0 59.0 56.0 56.5 55.0 55.0 53.5 54.0 55.5 58.5 58.0 54.0 55.0
##  [222] 58.0 58.0 57.0 57.0 57.0 55.5 57.0 56.0 55.0 54.5 55.0 56.5 56.5
##  [235] 55.0 54.0 56.0 54.5 60.0 58.0 58.0 58.0 58.0 55.0 57.0 57.5 59.0
##  [248] 59.0 58.0 54.5 56.0 56.0 55.0 55.0 54.0 56.5 56.0 59.5 58.0 57.0
##  [261] 55.5 58.5 55.0 57.0 58.0 59.0 60.0 59.0 57.0 55.5 55.0 58.0 57.0
##  [274] 52.5 54.0 54.0 55.0 55.0 56.5 54.5 55.0 57.0 55.0 55.5 55.0 59.5
##  [287] 57.0 59.0 57.0 56.0 58.5 56.0 55.0 54.0 54.0 55.0 54.5 54.5 58.0
##  [300] 55.5 59.0 58.0 58.5 57.0 56.5 59.0 58.5 58.0 56.0 55.0 56.0 59.0
##  [313] 59.5 58.0 55.0 57.0 56.0 54.0 55.0 59.0 54.0 57.0 55.5 55.0 53.5
##  [326] 61.0 59.0 58.0 57.0 56.0 56.0 58.5 58.0 58.0 56.0 54.0 55.0 55.0
##  [339] 56.0 58.0 56.5 55.0 55.0 55.0 58.5 58.0 58.0 59.0 58.0 56.5 60.0
##  [352] 56.0 58.0 54.0 54.5 55.5 58.0 58.0 59.0 60.0 56.0 59.0 60.0 58.5
##  [365] 58.0 59.0 56.0 54.5 58.0 56.5 59.0 58.0 55.0 56.0 54.0 56.0 56.0
##  [378] 57.0 53.0 53.5 54.0 54.0 56.0 55.0 57.0 56.0 59.0 59.5 59.0 56.0
##  [391] 58.0 58.5 60.0 56.0 57.0 58.5 58.5 58.0 57.5 58.0 57.0 56.0 55.0
##  [404] 55.0 59.5 55.5 58.5 60.0 54.0 55.0 56.0 57.0 56.0 53.0 54.0 57.0
##  [417] 57.0 54.0 57.0 58.0 57.0 57.0 57.0 56.0 58.0 58.5 58.0 54.0 55.0
##  [430] 54.5 55.0 55.0 60.0 58.5 58.0 59.0 55.0 56.5 57.0 59.0 56.0 60.0
##  [443] 56.0 59.0 56.0 58.0 60.0 58.0 58.0 57.0 57.0 58.0 57.0 57.0 59.0
##  [456] 57.0 57.0 56.5 57.0 55.0 54.5 56.0 56.0 56.0 56.0 55.0 52.0 56.5
##  [469] 55.5 57.0 55.0 54.5 54.5 56.0 58.0 56.0 59.0 60.0 57.0 60.0 58.0
##  [482] 57.5 56.0 56.0 57.0 55.0 56.0 57.5 56.0 58.0 53.0 59.0 56.0 59.0
##  [495] 56.0 56.0 59.0 59.5 58.0 58.5 59.0 57.0 57.0 58.0 57.0 57.0 55.0
##  [508] 57.0 55.5 55.0 57.0 58.0 57.0 59.0 59.0 57.0 57.5 57.5 57.0 56.5
##  [521] 58.0 57.5 57.0 68.0 57.0 59.0 60.0 59.0 57.5 56.0 57.0 56.0 57.0
##  [534] 58.5 58.0 56.5 57.0 57.5 60.0 57.0 58.0 57.0 56.5 59.0 61.0 58.0
##  [547] 57.0 56.0 54.5 56.0 56.0 55.0 55.0 57.0 59.0 58.0 59.0 57.0 58.0
##  [560] 57.5 56.5 58.0 57.0 59.0 59.0 59.0 56.5 57.5 57.0 59.5 60.0 58.0
##  [573] 58.0 58.0 57.0 55.0 55.5 59.0 60.0 56.0 59.0 57.5 56.0 58.0 58.0
##  [586] 58.0 57.5 56.0 58.5 57.5 60.0 57.5 59.0 58.5 55.5 56.5 60.0 59.0
##  [599] 57.0 57.0 55.0 54.5 59.0 54.0 59.0 59.0 59.5 59.0 59.0 59.0 58.0
##  [612] 60.0 58.0 58.0 58.0 59.0 61.0 58.0 61.0 57.0 58.0 58.0 55.0 56.0
##  [625] 57.0 57.0 55.5 58.5 57.0 58.0 56.0 60.0 59.0 59.5 59.0 59.0 59.0
##  [638] 58.5 57.5 59.0 58.5 57.0 59.0 58.5 59.0 58.5 57.5 60.0 58.0 58.0
##  [651] 59.0 59.0 56.0 56.0 55.0 59.0 57.0 58.0 59.5 59.5 57.5 57.0 58.5
##  [664] 56.5 56.0 59.0 58.0 60.0 60.0 59.5 59.0 58.0 57.5 59.0 59.5 57.5
##  [677] 55.5 57.0 58.0 60.0 60.0 59.0 58.0 57.0 60.0 58.0 58.5 58.0 59.5
##  [690] 59.0 57.0 58.5 59.0 60.0 59.5 60.0 54.0 56.0 55.5 58.0 60.0 58.0
##  [703] 59.0 58.0 57.0 60.5 58.0 60.0 59.0 59.0 61.0 57.0 59.5 59.5 57.0
##  [716] 57.5 59.0 59.0 58.0 57.0 57.5 55.0 57.0 57.0 59.0 58.0 61.0 60.0
##  [729] 59.0 57.0 58.0 57.0 60.0 60.0 60.0 59.0 57.0 57.0 60.0 60.0 59.0
##  [742] 57.0 58.0 59.0 57.0 55.0 58.5 58.0 58.0 58.0 60.0 59.5 58.0 58.0
##  [755] 60.0 59.5 57.0 59.0 60.0 58.0 58.5 59.0 54.0 54.0 57.0 55.5 55.0
##  [768] 57.0 56.5 59.0 58.0 60.0 59.0 59.0 59.5 60.0 59.5 61.0 58.0 58.0
##  [781] 58.0 59.0 58.5 58.0 58.0 59.0 59.5 59.5 58.0 55.0 60.5 59.5 58.0
##  [794] 58.0 59.0 59.5 58.0 58.0 59.0 60.5 60.5 58.5 59.0 60.0 57.0 58.0
##  [807] 59.0 59.0 59.0 56.5 59.0 56.0 56.0 58.0 57.0 58.0 57.5 55.5 57.0
##  [820] 58.0 57.0 60.0 59.0 58.0 57.0 58.0 58.5 59.0 58.0 58.0 58.0 56.5
##  [833] 60.0 58.0 59.0 59.0 59.0 59.0 57.0 60.0 60.5 57.0 58.0 57.0 58.0
##  [846] 59.5 57.0 59.0 57.0 59.0 58.0 56.5 57.0 59.0 58.5 59.0 58.0 56.0
##  [859] 59.0 58.5 57.0 58.0 58.0 59.0 59.0 59.5 54.5 54.0 53.0 56.0 59.0
##  [872] 57.0 61.0 59.0 61.0 59.5 59.0 60.0 59.0 60.0 61.0 59.0 59.0 60.5
##  [885] 58.5 56.0 59.0 59.0 59.5 57.5 56.0 55.0 59.0 56.5 59.0 58.0 61.5
##  [898] 58.0 61.0 60.0 58.0 58.0 55.5 60.0 61.0 57.0 59.0 58.0 59.5 60.0
##  [911] 59.0 59.0 60.0 58.0 57.0 59.0 59.0 56.5 60.5 57.0 57.0 60.0 61.0
##  [924] 57.5 60.0 54.0 55.0 56.0 60.5 60.0 59.0 60.0 58.5 57.0 58.0 59.0
##  [937] 58.0 61.5 59.0 59.0 58.5 60.5 58.5 60.0 59.0 59.0 59.0 58.0 58.5
##  [950] 57.5 58.0 56.0 58.0 58.0 58.5 55.0 55.0 55.0 59.5 59.5 59.0 59.5
##  [963] 60.5 57.5 59.5 59.5 59.5 60.5 59.0 54.0 59.0 56.5 58.0 59.0 57.5
##  [976] 59.0 60.5 58.5 59.0 58.0 59.5 57.0 58.5 57.0 61.5 61.0 58.5 57.5
##  [989] 61.5 59.0 59.5 57.5 61.0 59.0 60.0 59.0 58.0 59.0 57.0 59.5 59.0
## [1002] 57.0 55.5 55.0 58.5 58.5 58.0 57.0 59.0 59.0 61.0 59.0 59.5 61.0
## [1015] 61.0 57.0 55.5 62.0 64.0 59.0 60.0 60.5 59.5 59.0 58.5 57.0 56.0
## [1028] 60.0 59.0 59.0 60.0 59.0 59.0 57.0 59.0 59.0 60.0 60.0 61.0 60.0
## [1041] 59.5 58.5 59.0 60.0 58.5 56.0 57.0 56.5 55.0 59.5 59.5 58.5 58.5
## [1054] 58.0 61.0 59.0 58.0 59.0 61.0 60.0 60.0 59.5 61.0 56.0 55.5 58.0
## [1067] 59.0 60.0 59.5 61.0 59.0 59.0 58.0 59.0 60.0 60.5 60.0 55.0 55.5
## [1080] 60.5 60.0 56.5 60.5 59.5 59.5 57.0 57.5 59.5 58.5 57.5 61.0 59.0
## [1093] 59.5 61.0 57.5 59.5 57.5 57.0 58.0 58.0 61.0 62.0 55.5 56.5 55.0
## [1106] 55.0 56.0 58.0 57.0 59.5 56.5 61.0 55.5 60.5 59.5 58.0 60.0 58.0
## [1119] 54.5 56.0 62.0 60.0 59.0 57.0 61.0 59.5 57.0 59.0 61.0 58.0 56.5
## [1132] 58.5 61.0 57.0 58.0 55.0 59.0 59.5 58.5 54.5 59.5 60.0 59.0 58.0
## [1145] 59.0 59.5 63.0 56.0 59.5 59.0 59.0 59.0 60.0 60.0 59.0 60.0 60.0
## [1158] 61.5 59.0 55.5 54.0 53.5 57.0 57.5 64.0 59.5 58.0 58.5 58.5 61.0
## [1171] 59.0 60.0 58.0 62.0 58.0 60.0 60.0 61.0 66.0 59.0 57.0 59.5 56.0
## [1184] 61.0 58.0 60.0 60.0 58.0 64.0 62.0 63.5 63.5 61.5 58.8 60.0 58.0
## [1197] 63.0 62.0 61.5 62.5 57.0 61.0 59.5 60.0 59.5 57.0 57.0 62.0 61.5
## [1210] 60.5 61.0 63.0 61.0 62.0 60.0 62.5 62.0 61.0 55.5 57.0 61.0 62.0
## [1223] 59.0 64.0 62.0 63.0 62.5 61.5 56.5 60.0 60.0 58.0 64.5 63.0 62.0
## [1236] 57.0 61.0 65.0 66.0 63.0 61.0 60.5 58.5 63.0 64.0 60.0 62.0 63.5
## [1249] 61.0 60.0 65.0 59.0 59.5 62.0 62.5 65.0 59.0 61.0 59.5 63.5 60.0
## [1262] 62.0 63.0 60.0 60.0 67.0 57.5 63.0 63.0 64.0 54.5 60.0 58.0 63.5
## [1275] 62.0 63.0 61.0 64.5 63.0 63.0 57.0
```

If you save a MS Excel file as a comma separated csv file, you will use a similar command but with a comma for the separator. You can try this with the `Salamander_Demographics.csv` file.

```
sally <- read.table("Data/Salamander_Demographics.csv", sep = ",", header = TRUE, stringsAsFactors = FA
str(sally)
```

```
## 'data.frame':    3382 obs. of  20 variables:
##  $ line : int  1861 1115 360 2897 1432 372 231 2739 2236 543 ...
##  $ page : int  60 36 12 92 46 12 8 87 72 17 ...
##  $ dates : chr  "4/21/09" "9/9/08" "5/31/08" "5/7/11" ...
##  $ month : int  4 9 5 5 10 5 5 10 5 6 ...
##  $ day   : int  21 9 31 7 16 31 27 24 14 5 ...
##  $ year  : int  2009 2008 2008 2011 2008 2008 2008 2009 2009 2008 ...
##  $ time  : chr  "N" "N" "N" "N" ...
##  $ plot  : chr  "5" NA "3" "7" ...
##  $ mass  : num  0.427 0.633 0.639 0.921 0.943 ...
##  $ svl   : int  33 37 42 43 45 46 47 48 NA NA ...
##  $ tl    : int  63 68 63 79 74 NA 75 89 87 NA ...
##  $ sex   : chr  NA NA NA NA ...
##  $ gravid: chr  "N" "N" "N" "N" ...
##  $ group : chr  NA NA NA NA ...
##  $ clutch: int  NA NA NA NA NA NA NA NA NA NA ...
##  $ color : chr  "R" "R" "R" "R" ...
##  $ recap : chr  NA NA NA "N" ...
##  $ mark  : chr  NA NA NA NA ...
##  $ id    : int  1371 NA 187 2154 1042 198 74 2036 1564 351 ...
##  $ damage: chr  "N" "N" "Y" "N" ...
```

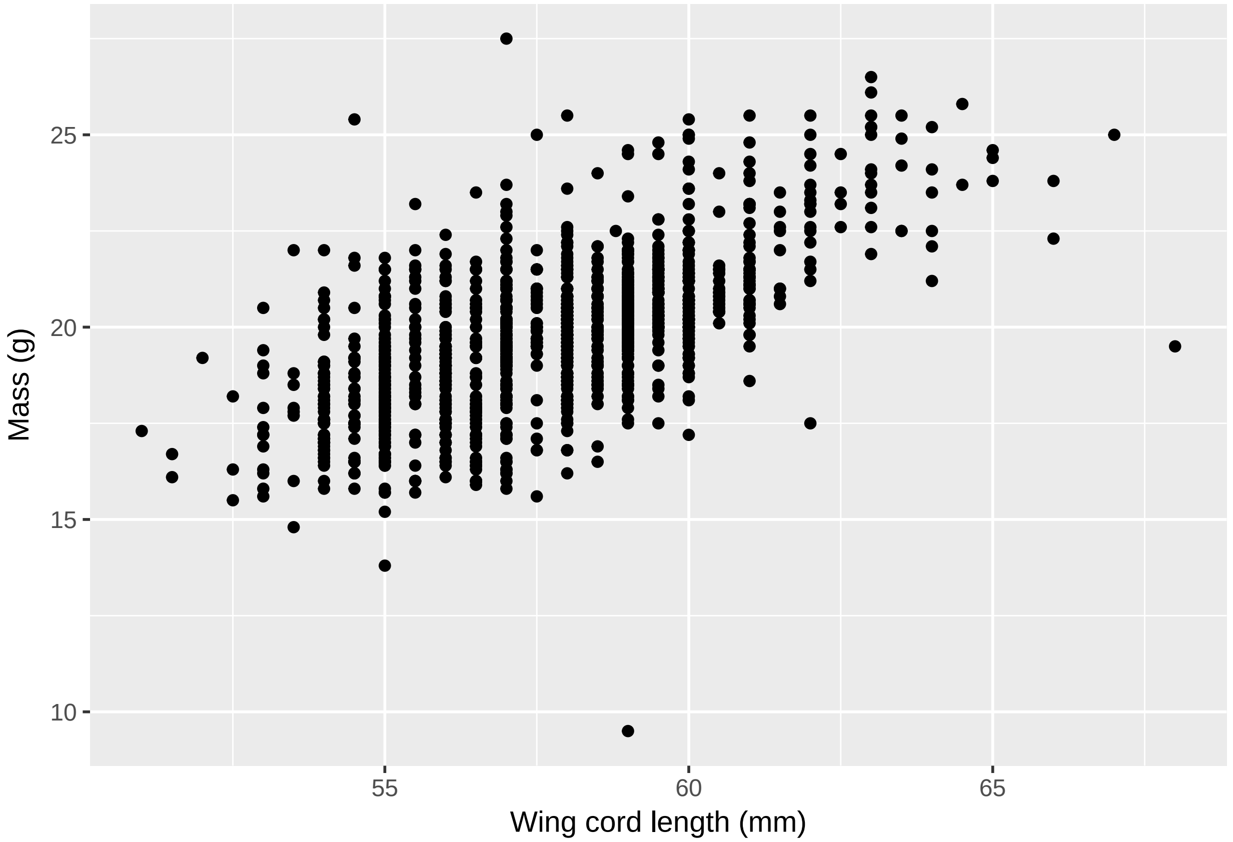
Regression with the sparrow data

Let's use the sparrow data because we are familiar with it. Let's pick a length measurement, wingcrd, and estimate the relationship with mass (unfortunately named `wt` for weight).

Plotting

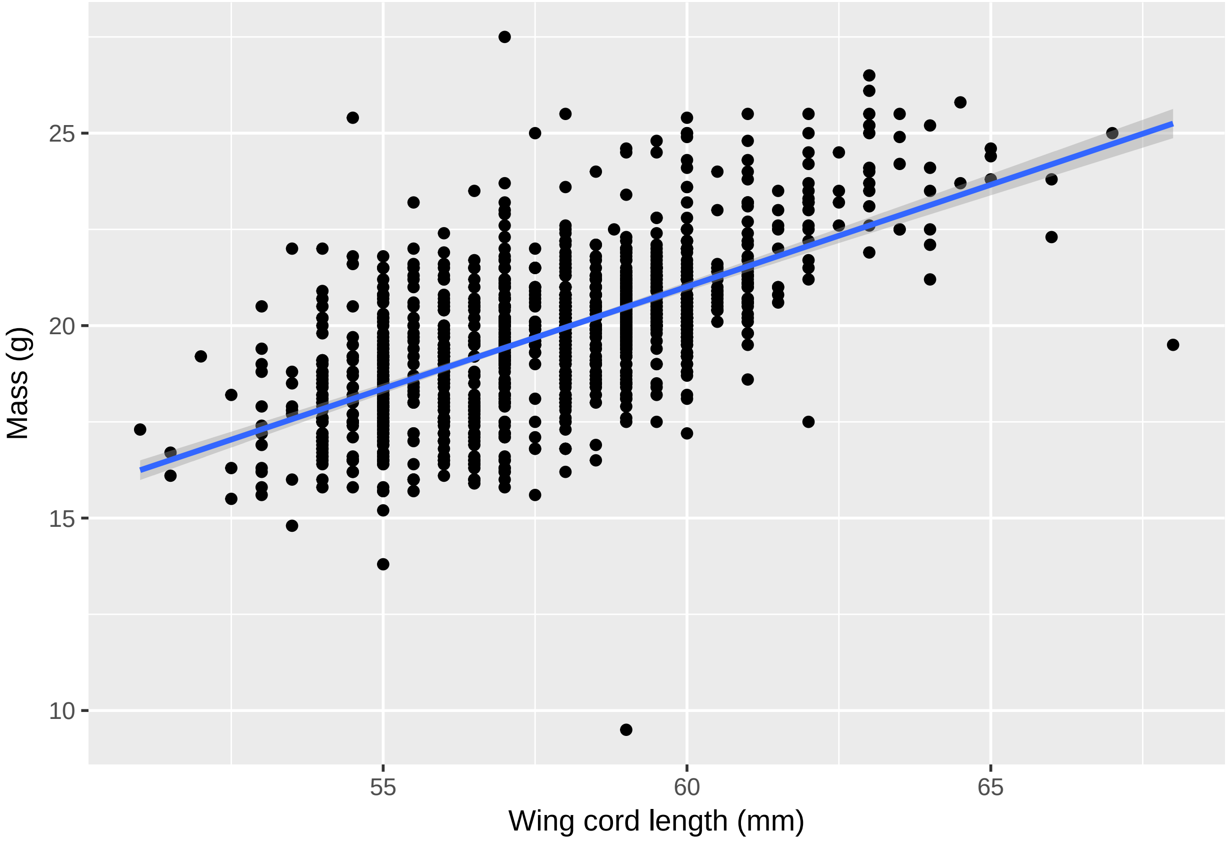
You should always plot your data first. The x-axis should be `wingcrd`, and the y-axis should be `wt`. We usually use a scatter plot for plotting two continuous variables, and so let's use a point geometry in the ggplot2 package to show each observation.

```
ggplot(sparrows, aes(x = wingcrd, y = wt)) +
  geom_point() +
  labs(x='Wing cord length (mm)', y='Mass (g)')
```



It looks like a line fits that fairly well. So first, let's go ahead and add a line to the plot using `geom_smooth`.

```
ggplot(sparrows, aes(x = wingcrd, y = wt)) +
  geom_point() +
  labs(x='Wing cord length (mm)', y='Mass (g)') +
  geom_smooth(method = "lm")
```



You see the regression line in blue, but a grey band around the regression that represents our confidence in the regression line. We'll get that soon enough, but for now just recognize that we estimated this based on a sample, and so there is some uncertainty in our data, and thus, there is some uncertainty in our line. The grey band is an estimate of where the regression could be given this uncertainty.

Visually, this looks like a decent regression.

The linear model function and interpretation

Now, let's fit the regression using the function `lm`. To use `lm`, we need to specify the y and x variables. R has a type of "short hand" for linear models that looks like this : `y ~ x`. R will interpret that $y = b_0 + b_1 x$.

```
my_lm <- lm(wt ~ wingcrd, sparrows)
```

Right now, R has finished the regression, and we just need to know how to print it out.

```
summary(my_lm)
```

```
##
## Call:
## lm(formula = wt ~ wingcrd, data = sparrows)
##
## Residuals:
```

```
##      Min       1Q   Median       3Q      Max
## -10.9821  -0.8766   0.0017   0.7828   8.0774
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -10.77121    1.05992  -10.16   <2e-16 ***
## wingcrd       0.52972    0.01835   28.86   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.506 on 1279 degrees of freedom
## Multiple R-squared:  0.3945, Adjusted R-squared:  0.394
## F-statistic: 833.1 on 1 and 1279 DF,  p-value: < 2.2e-16
```

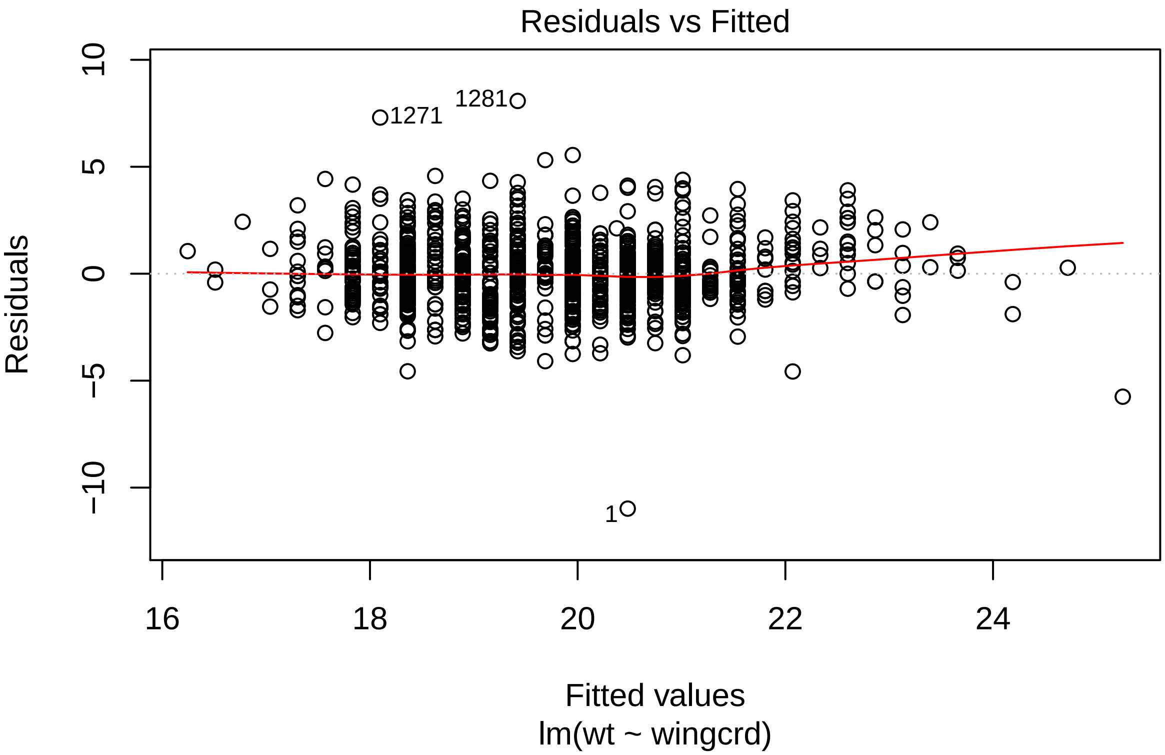
That's it. Pretty easy, right?

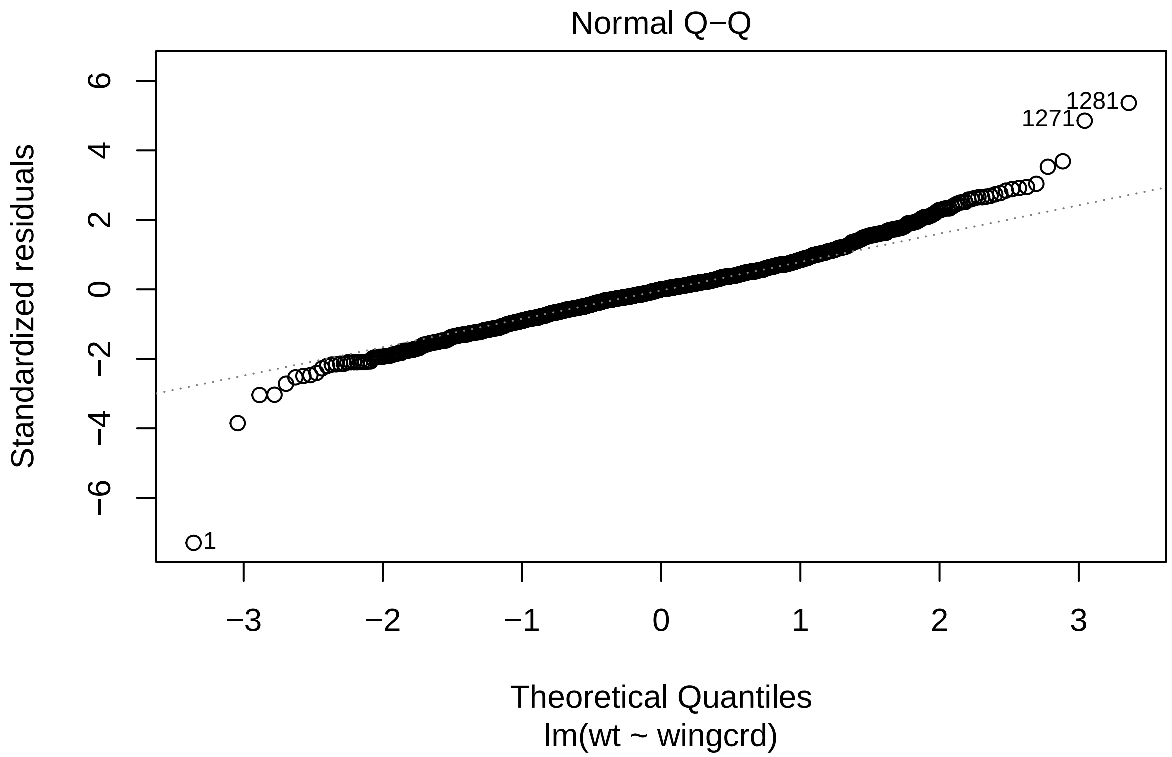
What does the line predict for the mass for a bird with a wing cord of 70 mm? It predicts $-10.77121+0.52972*70$ or 26.3 g. This might be reasonable.

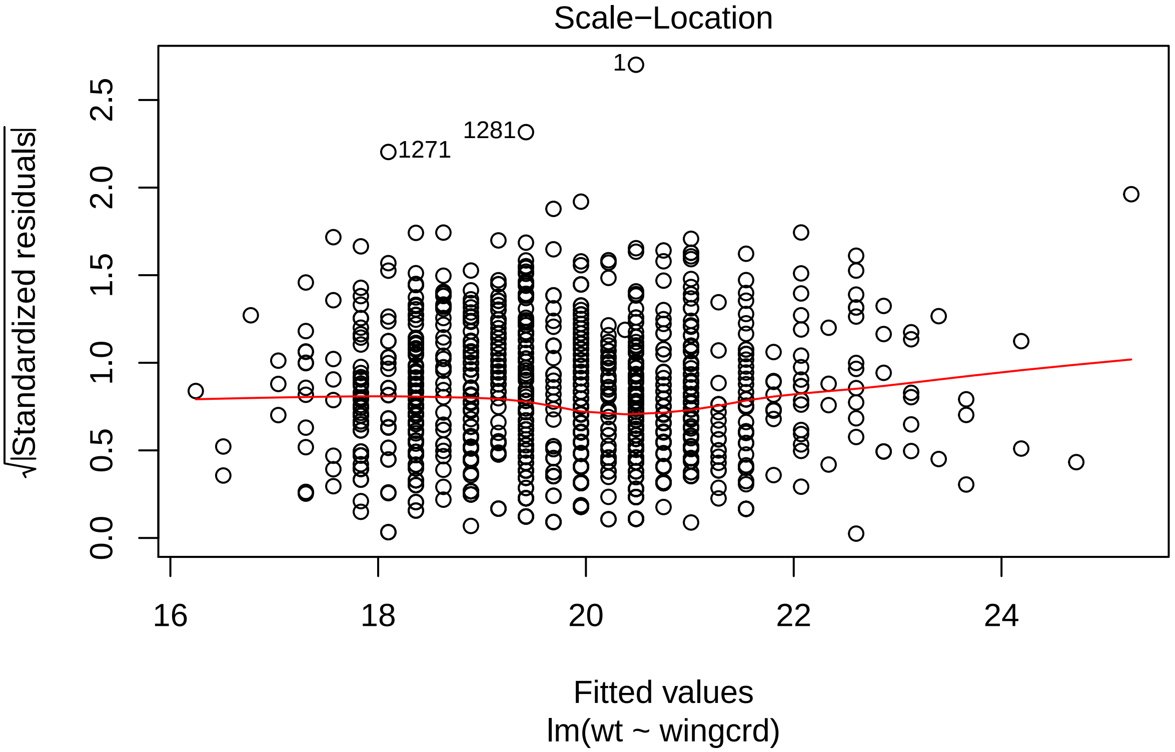
What does the line predict for the mass for a bird with a wing cord of 200 mm? It predicts $-10.77121 + 0.52972 * 200$ or 95.1 g. This would be a big bird and is totally unrealistic for a sparrow. Be careful when making predictions. In general, don't make predictions that aren't supported by your data.

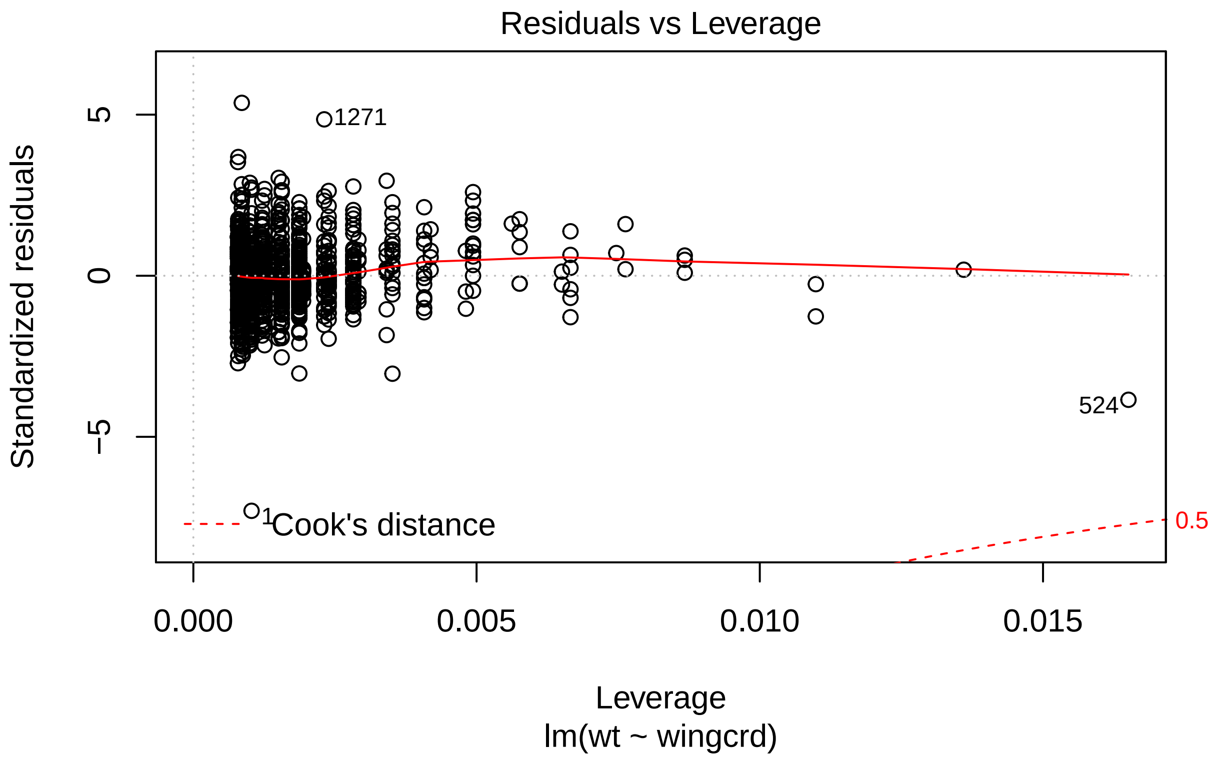
We always want to check the to see if our model and data matched the assumptions of the linear model (refer back to your lecture notes). We can do this by visualizing the residuals. It's easy with the plot function on your fitted model object.

```
plot(my_lm)
```




Normal Q−Q
Standardized residuals
6
4
2
0
−2
−4
−6
1281
1271
1
−3
−2
−1
0
1
2
3
Theoretical Quantiles
lm(wt ~ wingcrd)


Scale−Location
√|Standardized residuals|
2.5
2.0
1.5
1.0
0.5
0.0
1
1281
1271
16
18
20
22
24
Fitted values
lm(wt ~ wingcrd)

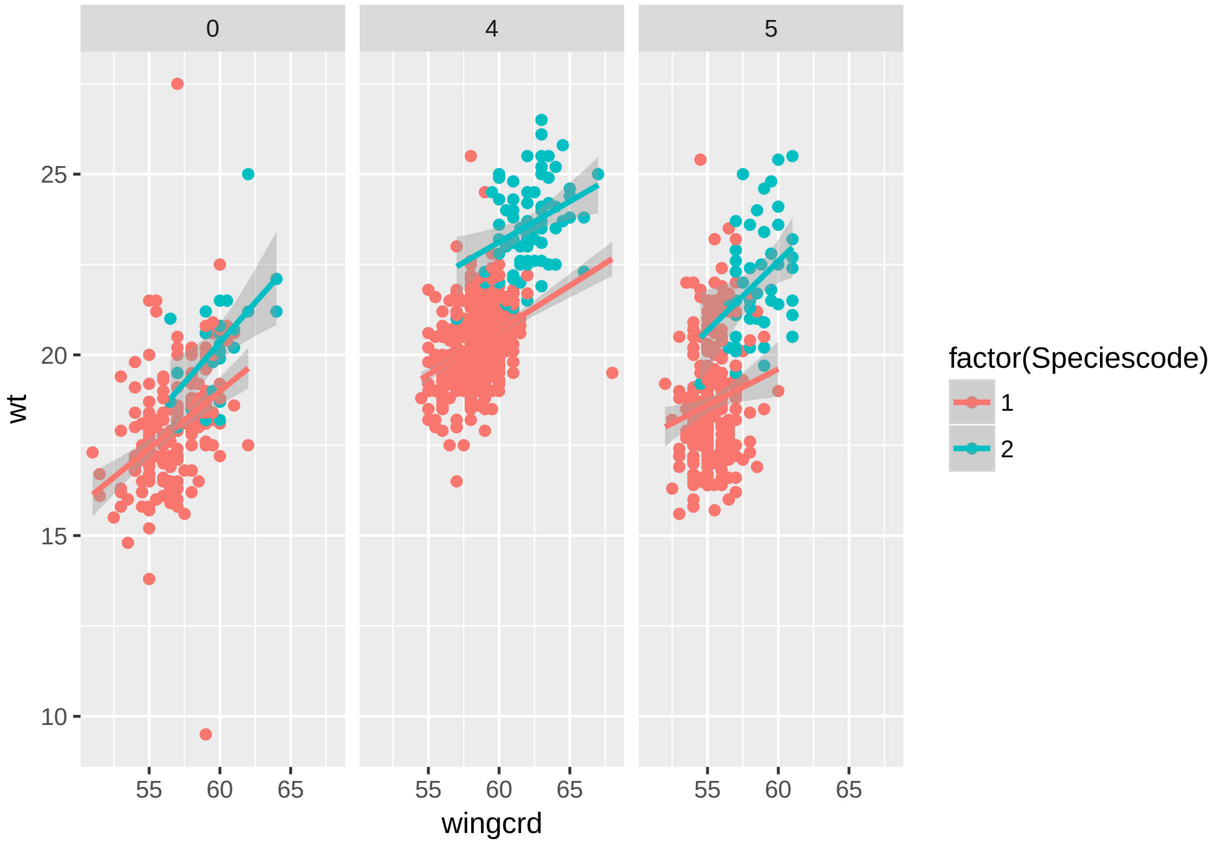


Calculate average response for each Species.

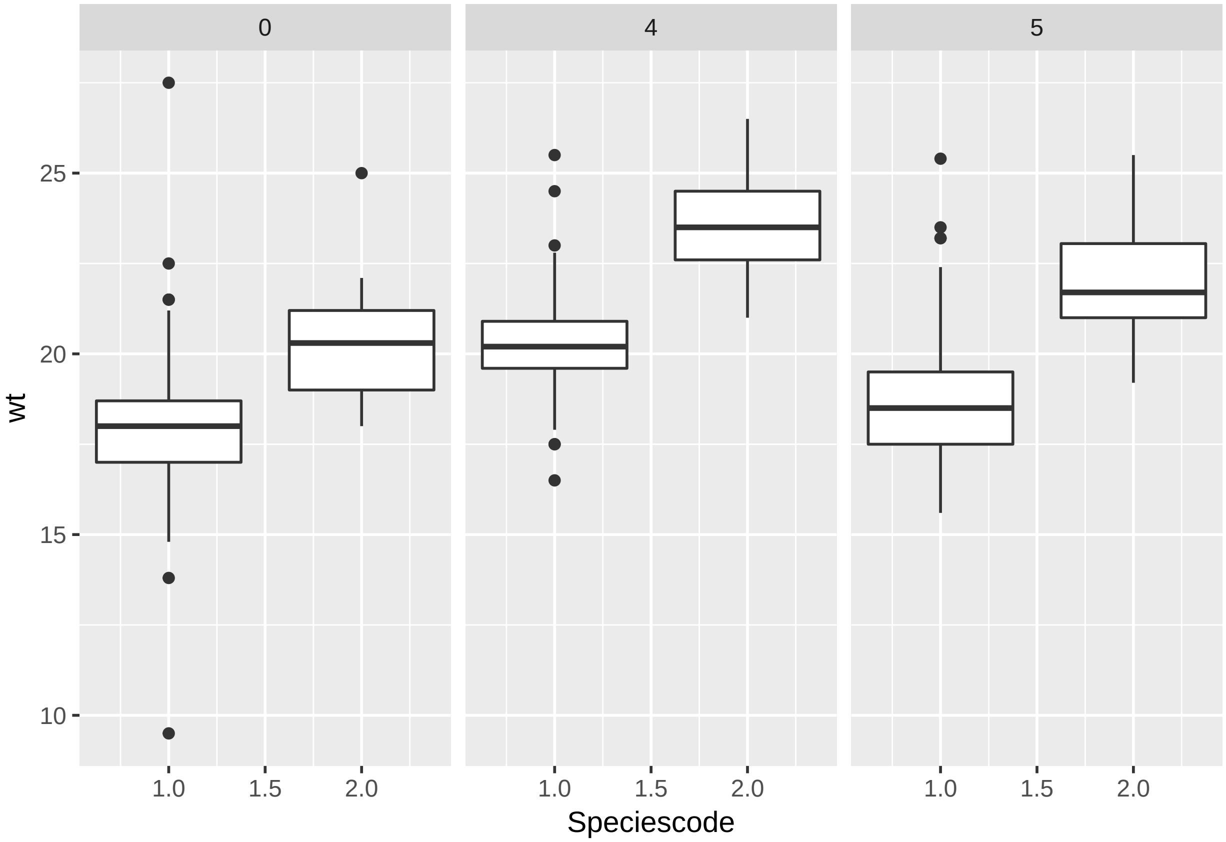
We will calculate the average reponse per species

I will draw bar charts for each country. If there are dramatic breaks between adjacent groups, then that may suggest that that differences between adjacent categories is very nonlinear.

```
ggplot(data=sparrows, aes(wingcrd, wt, colour = factor(Speciescode))) + geom_point() + geom_smooth(meth
```



```
ggplot(data=sparrows, aes(Speciescode, wt, group = Speciescode)) + geom_boxplot() + facet_wrap(~Sex)
```



To calculate species averages, we will need to (1) `group_by` species, and then (2) summarize by `mean`. I've been doing this for awhile, so I know that `mean` won't like the NAs (missing values) in the data. So we will have to tell `mean` that it can remove NAs if it needs to.

```
species_data <- sparrows %>%
  group_by(Speciescode) %>%
  summarise(wingcrd = mean(wingcrd, na.rm=TRUE),
            wt = mean(wt, na.rm=TRUE)) %>%
  ungroup()

species_data
```

```
## # A tibble: 2 x 3
##   Speciescode  wingcrd       wt
##         <int>    <dbl>    <dbl>
## 1           1 57.32921 19.43989
## 2           2 60.62027 22.54324
```

Export Data

Now imagine you want to save some data or results you worked up in R. To save a set of R objects for later use in R, you can use the `save` function.

```
save(my_lm, species_data, file = "Output/lab2_tutorial.RData")
```

If you have a table you want to be able to open in a spreadsheet like MS Excel you can use the `write.csv` function.

```r
write.csv(species_data, file = "Output/species_data.csv", row.names = FALSE)
```

Commands learned in this lab

Base R

- `$`
- `mean( , na.rm=TRUE)`
- `lm()`
- `save()`
- `write.csv()`

tidyverse (dplyr)

- `%>%`
- `group_by()`
- `summarise()`
- `mean()`

ggplot

- `geom_smooth(, method='lm')`