Elementary Statistical Modeling for Applied Biostatistics

Jeffrey A. Walker

2018-10-02

Contents

1 Statistical Modeling **5**
- 1.1 Statistical modeling with linear models . . . 5
- 1.2 Model fitting . . . 8
- 1.3 Multilevel models . . . 10
- 1.4 Linear models versus non-linear models . . . 10

2 Organization – R Projects and R Notebooks **11**
- 2.1 Importing Packages . . . 11
- 2.2 Create an R Studio Project for this Class . . . 11
- 2.3 R Notebooks . . . 11

3 Data – Reading, Writing, and Fake **15**
- 3.1 Create new notebook for this chapter . . . 15
- 3.2 Importing Data . . . 15
- 3.3 Creating Fake Data . . . 23
- 3.4 Saving Data . . . 25
- 3.5 Problems . . . 26

4 Variability and Uncertainty (Standard Deviations and Standard Errors) **27**
- 4.1 The sample standard deviation vs. the standard error of the mean . . . 27
- 4.2 Using Google Sheets to generate fake data to explore uncertainty . . . 29
- 4.3 Using R to generate fake data to explore uncertainty . . . 31
- 4.4 Bootstrapped standard errors . . . 33

5 A linear model with a single, continous *X* **35**
- 5.1 A linear model with a single, continous X is classical "regression" . . . 35
- 5.2 Working in R . . . 38
- 5.3 Problems . . . 43

6 Least Squares Estimation **45**
- 6.1 OLS regression . . . 45
- 6.2 How well does the model fit the data? R^2 and "variance explained" . . . 46

7 A linear model with a single, categorical *X* **51**
- 7.1 A linear model with a single, categorical X is the engine behind a single factor (one-way) ANOVA and a t-test is a special case of this model. . . . 51
- 7.2 Working in R . . . 54

8 P-values **59**
- 8.1 p-values . . . 59
- 8.2 Creating a null distribution. . . . 60
- 8.3 Statistical modeling instead of hypothesis testing . . . 64
- 8.4 frequentist probability and the interpretation of p-values . . . 65

8.5 Problems 71

Appendix 1: Getting Started with R **73**
8.6 Get your computer ready 73
8.7 Start learning 73
8.8 Getting Data into R 74
8.9 Additional R learning resources 74
8.10 Packages used extensively in this text 74

Appendix 2: Online Resources for Getting Started with Linear Modeling in R **75**

Chapter 1

Statistical Modeling

More cynically, one could also well ask "Why has medicine not adopted frequentist inference, even though everyone presents P-values and hypothesis tests?" My answer is: Because frequentist inference, like Bayesian inference, is not taught. Instead everyone gets taught a misleading pseudo-frequentism: a set of rituals and misinterpretations caricaturing frequentist inference, leading to all kinds of misunderstandings. – Sander Greenland

We use statistics to learn from data with uncertainty. Traditional introductory textbooks in biostatistics implicitly or explicitly train students and researchers to "discover by p-value" using hypothesis tests (appendix xxx). Over the course of many chapters, the student is trained to use something like a dichotomous key to choose the correct "test" for the data at hand, compute a test statistic for their data, compute a p-value based on the test statistic, and compares the p-value to 0.05. Textbooks typically give very little guidance about what can be concluded if $p < 0.05$ or if $p > 0.05$, but many researchers conclude (incorrectly) they have "discovered" something if $p < 0.05$ but found "no effect" if $p > 0.05$.

Researchers learn almost nothing useful from a hypothesis test. True, a p-value is evidence against the null, and thus, a tool to dampen our tendency to being fooled by randomness. But if we are investigating the effects of an increasingly acidified ocean on coral growth, $p = 0.002$ may be evidence of an effect of the experimental intervention, but, from everything we know about pH and cell biology, it would be absurd to conclude from any data that pH does not affect growth. Instead, we want to know the magnitude of the effect and our uncertainty in estimating this magnitude. We can use this magnitude and uncertainty to make predictions about the future of coral reefs, under different scenarios of ocean acidification. We can use the estimated effects and uncertainty to model the consquences of the effects of acidification on coral growth on fish production or carbon cycling.

The "discovery by p-value" strategy, or Null-Hypothesis Significance Testing (NHST), has been criticized by statisticians for many, many decades. Nevertheless, introductory biostatistics textbooks written by both biologists and statisticians continue to organize textbooks around a collection of hypothesis tests, with little emphasis on estimation and uncertainty.

1.1 Statistical modeling with linear models

This book is an introduction to the analysis of biological data using a statistical modeling approach. As an introduction, the focus will be linear models and extensions of the linear models including linear mixed models and generalized linear models. Here, I refer to all of these as "linear models" because all are a function of a linear predictor. Linear models are the engine behind many hypothesis tests but the emphasis in statistical modeling is estimation and uncertainty instead of test statistics and p-values. A modeling view of statistics is also more coherent than a dichotomous key strategy.

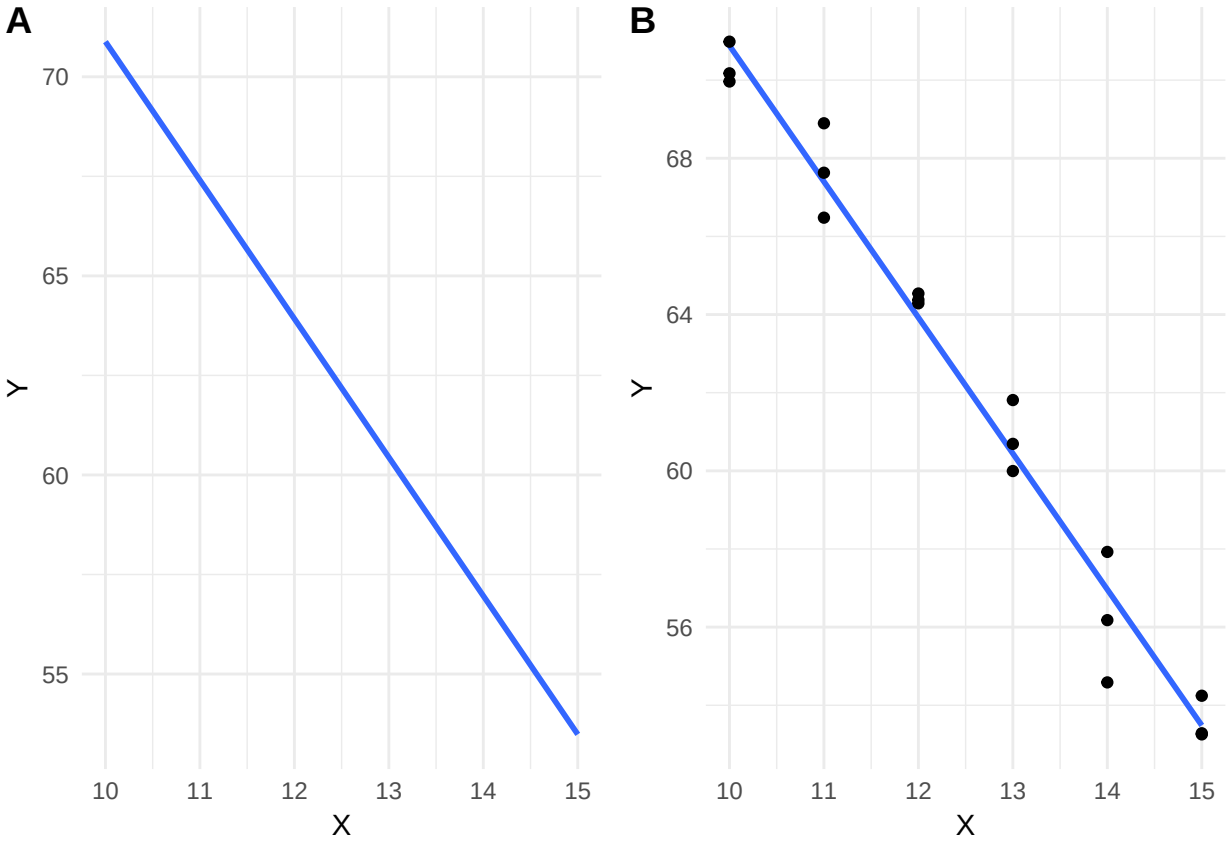


Figure 1.1: A line vs. a linear model. (A) the line $y=-3.48X + 105.7 is drawn. (B) A linear model fit to the data. The model coefficients are numerically equal to the slope and intercept of the line in A.

All students are familiar with the idea of a linear model from learning the equation of a line, which is

$$Y = mX + b \tag{1.1}$$

where m is the slope of the line and b is the Y-intercept. It is useful to think of equation (1.1) as a function that maps values of X to values of Y. Using this function, if we input some value of X, we always get the same value of Y as the output.

A linear model is a function, like that in equation (1.1), that is fit to a set of data, often to model a process that generated the data or something like the data. The line in Figure 1.1A is just that, a line, but the line in Figure 1.1B is a model of the data in Figure 1.1B. The basic structure of a linear model is

$$Y = \beta_0 + \beta_1 X + \varepsilon \tag{1.2}$$

A linear model has two parts: the "linear predictor" ($Y = \beta_0 + \beta_1 X$) and the "error" (ε). The linear predictor part looks like the equation for a line except that I've used β_0 for the intercept and β_1 for the slope and I've put the intercept term first. This re-labeling and re-arrangement make the notation for a linear model more flexible for more complicated linear models. For example $Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \varepsilon$ is a model where Y is a function of two X variables.

As with the equation for a line, the linear predictor part of a linear model is a function that maps a value of X to a specific value of Y. This mapped value is the **expected value** given a specific input value of X. This is often written as $\mathrm{E}[Y|X]$. The error part of a linear model is a random variable that adds some random value to this expected value. Nothing about the model part of a linear model can predict its value.

The inputs to a linear model (the X variables) have many names including "independent variables," "predictor variables,", "explanatory variables," "treatment variables," and "covariates". The output of a linear model (the Y variable or variables if the model is multivariate) is the "dependent variable," "response," or "outcome." The β in the linear model are model **parameters** There can be additional parameters in more sophisticated models. The coefficients of the X in a linear model (β_1 in model (1.2)) are often called "the effects" (so β_1 is the effect of X_1).

Although a linear model is a model of a data-generating process, linear models are not typically used to actually generate any data. Instead, when we use a linear model to understand something about a real dataset, we think of our data as one realization of a process that generates data like ours. A linear model is a model of that process. That said, it is incredibly useful to use linear models to create fake datasets for at least two reasons: to probe our understanding of statistical modeling generally and, more specifically, to check that a model actually creates data like that in the real dataset that we are analyzing.

1.1.1 Linear models are used for prediction, explanation, and description

Researchers typically use linear models to understand relationships between one or more Y variables and one or more X variables. These relationships include

1. Descriptive modeling. Sometimes a researcher merely wants to describe the relationship between Y and a set of X variables, perhaps to discover patterns. For example, the arrival of a spring migrant bird (Y) as a function of sex (X_1) and age (X_2) might show that males and younger individuals arrive earlier. Importantly, if another X variable is added to the model (or one dropped), the coefficients, and therefore, the precise description, will change. That is, the interpretation of a coefficient as a descriptor is *conditional* on the other covariates (X variables) in the model. In a descriptive model, there is no implication of causal effects and the goal is not prediction. Nevertheless, it is very hard for humans to discuss a descriptive model without using causal language, which probably means that it is hard for us to think of these models as *mere description*. Like natural history, descriptive models are useful as patterns in want of an explanation, using more explicit causal models including experiments.

2. Predictive modeling. Predictive modeling is very common in applied research. For example, fisheries researchers might model the relationship between population density and habitat variables to predict which subset of ponds in a region are most suitable for brook trout (*Salvelinus fontinalis*) reintroduction. The goal is to build a model with minimal prediction error, which is the error between predicted and actual values for a future sample. In predictive modeling, the X ("predictor") variables are largely instrumental – how these are related to Y is not a goal of the modeling, although sometimes an investigator may be interested in the relative importance among the X for predicting Y (for example, collecting the data may be time consuming, or expensive, or enviromentally destructive, so know which subset of X are most important for predicting Y is a useful strategy).

3. Explanatory (causal) modeling. Very often, researchers are explicitly interested in *how* the X variables are causally related to Y. The fisheries researchers that want to reintroduce trout may want to develop and manage a set of ponds to maintain healthy trout populations. This active management requires intervention to change habitat traits in a direction, and with a magnitude, to cause the desired response. This model is predictive – a specific change in X predicts a specific response in Y – because the coefficients of the model provide knowledge on how the system functions – how changes in the inputs *cause* change in the output. Causal interpretation of model coefficients requires a set of strong assumptions about the X variables in the model. These assumptions are typically met in **experimental designs** but not **observational designs**.

With observational designs, biologists are often not very explicit about which of these is the goal of the modeling and use a combination of descriptive, predictive, and causal language to describe and discuss results. Many papers read as if the researchers intend explanatory inference but because of norms within the biology community, mask this intention with "predictive" language. Here, I advocate embracing explicit, explanatory modeling by being very transparent about the model's goal and assumptions.

1.2 Model fitting

In order to use a linear model to describe, predict, or explain, we need to fit a model to data in order to estimate the parameters. If we fit model (1.3) to some data, the estimated parameters are the coefficients (b_0 and b_1) of the fit model

$$\mathrm{E}[Y|X] = b_0 + b_1 X \tag{1.3}$$

The left-hand side of equation (1.3) is the **conditional expectation** and is read as "the expectation of Y given X" or "the expected value of Y given X". Throughout this book, I use the greek β to refer to a theoretical, data-generating parameter and the roman "b" to refer its estimate.

The goal of descriptive and explanatory modeling is the estimate of the coefficients of the X variables and their uncertainty. The goal of predictive modeling is the estimate of predicted values, and their uncertainty, given specific values of X. These predicted values are the conditional expectations.

For the model fit to the data in Figure 1.1B, the coefficient of X is the slope of the line. Perhaps surprisingly, we can fit a model like equation (1.2) to data in which the X variable is categorical. A simple example is the experiment of antioxidants (vitamins C and E) on lifespan in Voles (Fig. 1.2). In this experiment, the X variable is categorical, with three **levels**: "Control", "Vitamin_E" and "Vitamin_C". Categorical X variables are often called **factors**. The trick to using a linear model with categorical X is to recode the factor levels into numbers – how this is done is explained in Chapter xxx. When the X variable is categorical, the coefficients of the X are *differences in group means*. The linear model fit to the vole data has two coefficients, one for Vitamin E and one for vitamin C. The estimate and uncertainty of the these two coefficients are shown in the top part of Figure 1.2. The bottom part shows the raw data, as well as the group (factor level) means and the uncertainty in the estimate of these means.

The simplest possible model that can be fit to the data is

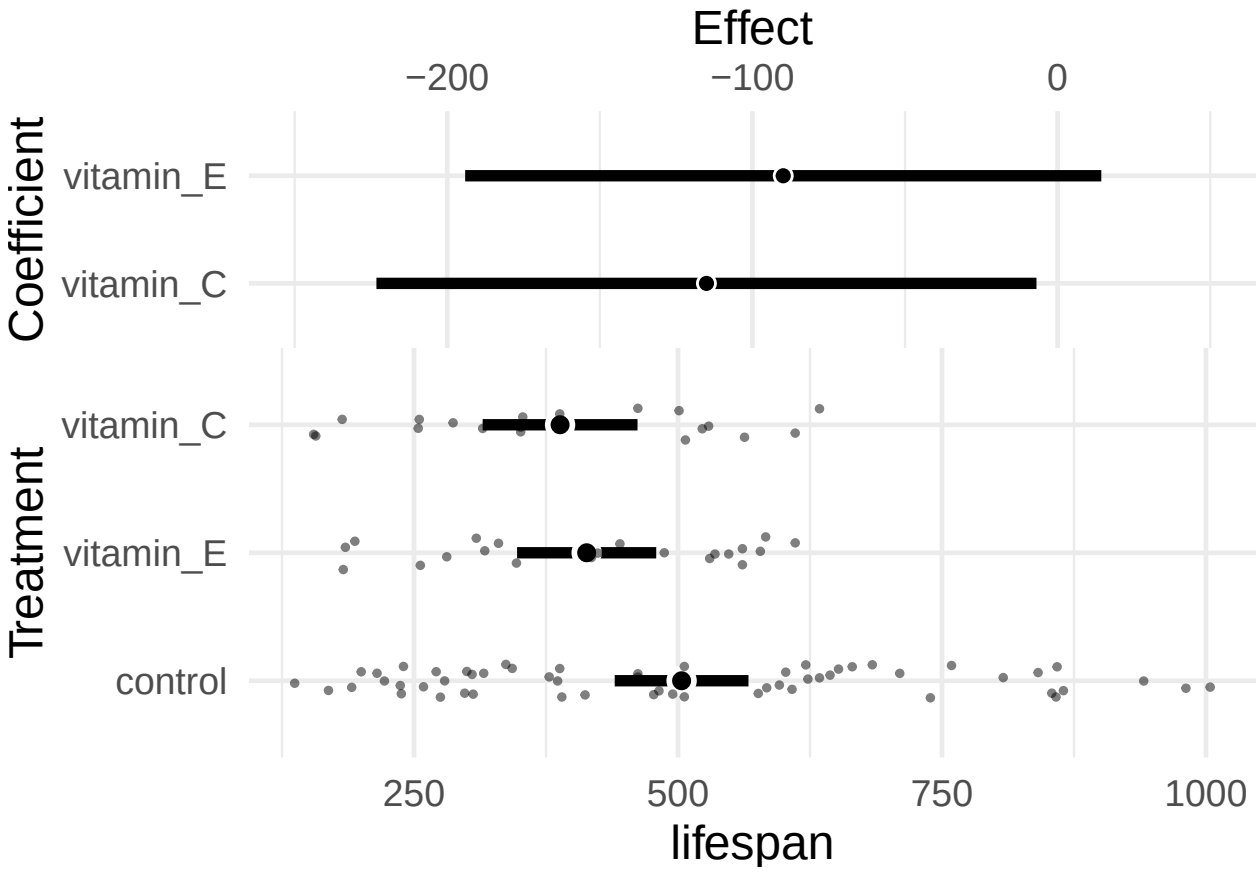


Figure 1.2: HarrellPlot of vole data.

$$\mathrm{E}[Y] = b_0 \tag{1.4}$$

which is simply the mean of Y, or, more specifically, the **unconditional mean** of Y, since its value is not conditional on any value of X.

1.2.1 "Statistical model" not "regression model"

Statistical modeling terminology can be confusing. The X variables in a statistical model may be quantitative (continuous or integers) or categorical (names or qualitative amounts) or some mix of the two. Linear models with all quantitative independent variables are often called "regression models." Linear models with all categorical independent variables are often called "ANOVA models." Linear models with a mix of quantitative and categorical variables are often called "ANCOVA models" if the focus is on one of the categorical X or "regression models" if there tend to be many independent variables. Other patterns occur. For example "ANCOVA models" often include interaction effects but "regression models" rarely do. To avoid thinking of statistical analysis as "regression vs. ANOVA", I will most often use the term "statistical model" for general usage, and use a more specific term only to emphasize something about the model in that particluar context.

1.3 Multilevel models

1.4 Linear models versus non-linear models

In this text, I use "linear model" for any model that is linear in the parameters, which means that the different components of the model are added together. Or, using the language of matrix algebra, the predictor is a simple dot product of the model matrix and the coefficients. For example, a cubic polynomial model

$$Y = \beta_0 + \beta_1 X + \beta_2 X^2 + \beta_3 X^3 + \varepsilon \tag{1.5}$$

is a linear model, even though the function is non-linear, because the different components are added (or, using matrix algebra, the predictor is $\mathbf{X}\boldsymbol{\beta}$).

A generalized linear model (GLM) has the form $g(\mu_i) = \eta_i$ where η (the greek letter eta) is the linear predictor, which is linear in the parameters.

$$\eta = \mathbf{X}\boldsymbol{\beta} \tag{1.6}$$

Many sources do not consider a GLM to be a "linear model" but an "extension" of a linear model. Regardless, a GLM is linear in the parameters and here, I include GLMs under the "linear model" umbrella.

Non-linear models, in conrast to a GLM or classical linear model, are not linear in the parameters (the predictor is not a simple dot product of the model matrix and a vector of parameters). For example, the Michaelis-Menten model is a nonlinear model

$$Y = \frac{\beta_1 X}{\beta_2 + X} + \varepsilon \tag{1.7}$$

Chapter 2

Organization – R Projects and R Notebooks

2.1 Importing Packages

The R scripts you write will include functions in packages that are not included in Base R. These packages need to be downloaded from an internet server to your computer. You only need to do this once. But, each time you start a new R session, you will need to load a package using the `library()` function. Now is a good time to import packages that we will use

1. Open R Studio and choose the menu item "Tools" > "Install Packages"
2. In the "packages" input box, insert "ggplot2, data.table, emmeans, lme4, reshape2". Make sure that "install dependencies" is clicked before you click "Install"

Again, once these are installed, you don't need to do this again. You simply need to use the `library()` function at the start of a script.

2.2 Create an R Studio Project for this Class

1. Create a folder named "BIO_413"
2. Within this folder, create new folders named
    1. "notebooks" – this is where your R notebooks are stored
    2. "R" – this is where R scripts are stored
    3. "data" – this is where data that we download from public archives are stored
    4. "output" – this is where you will store fake data generated in this class
    5. "images" – this is where image files are stored
3. Open R Studio and click the menu item File > New Project…
4. Choose "Existing Directory" and navigate to your BIO_413 folder
5. Choose "Create Project"
6. Check that a file named "BIO_413.Rproj" is in your BIO_413 folder

2.3 R Notebooks

A typical statistical modeling project will consist of:

1. reading data from Excel or text (.csv or .txt) files

2. cleaning data
3. analysis
4. generating plots
5. generating tables
6. writing text to describe the project, the methods, the analysis, and the interpretation of the results (plots and tables)

The best practice for reproducible research is to have all six of these steps in your R Notebook. Too many research projects are not reproducible because the data were cleaned in Excel, and then different parts of the data were separately imported into a GUI statistics software for analysis, and then output from the statistics software was transcribed to Excel to make a table. And other parts of the analysis are used to create a plot in some plotting software. And then the tables and plots are pasted into Microsoft Word to create a report. Any change at any step in this process will require the researcher to remember all the downstream parts that are dependent on the change and to re-do an analysis, or a table, or a plot, etc. etc.

The goal with an R Studio Notebook is to explicitly link all this so that changes in earlier steps automatically flow into the later steps. So, at the end of a project, a researcher can choose "run all" from the menu and the data are read, cleaned, analyzed, ploted, tabled, and put into a report with the text.

This means that you have to think of the organization of the R code that your write in a Notebook. Your cannot simply append new code to the end of a script if something earlier (or above) is dependent on it. You need to go back up and insert the new code at some earlier (and meaningful) point.

For example, an R chunk generates 100 random normal values and then plots these with a histogram. This was the chunk that I wrote

```
x <- rnorm(n)
qplot(x)
```

When I ran the chunk, I got the error "Error in rnorm(n) : object n not found". I was using the function `rnorm()` to generate values but I hadn't assigned any value to `n` yet, so I got the error. To get this to work properly, I could have just typed `n <- 100` in the console and then re-run the script but I want it to work properly on a fresh run of the chunk (after quitting and re-opening R Studio) so I instead inserted `n <- 100` at the start of the chunk, like this:

```
n <- 100
x <- rnorm(n)
qplot(x)
```

2.3.1 Create an R Notebook for this Chapter

1. The top-left icon in R Studio is a little plus sign within a green circle. Click this and choose "R Notebook" from the pull-down menu.
2. Change the title of the notebook to "Notebook_01-organization"
3. Delete the default R Markdown text starting with "This is an [R Markdown]…"

Now write some text documenting which packages you installed.

2.3.2 Create a "setup" chunk

1. Click on the "Insert" menu on the right hand side of the script (R Markdown) pane and choose "R". This will insert an R code chunk into your R markdown document.
2. The first R chunk of a notebook should be a setup chunk. Name the chunk "setup"
3. load the libraries ggplot2 and data.table and click the chunk's run button (the green triangle to the right of the chunk)

```
library(ggplot2)
library(data.table)
```

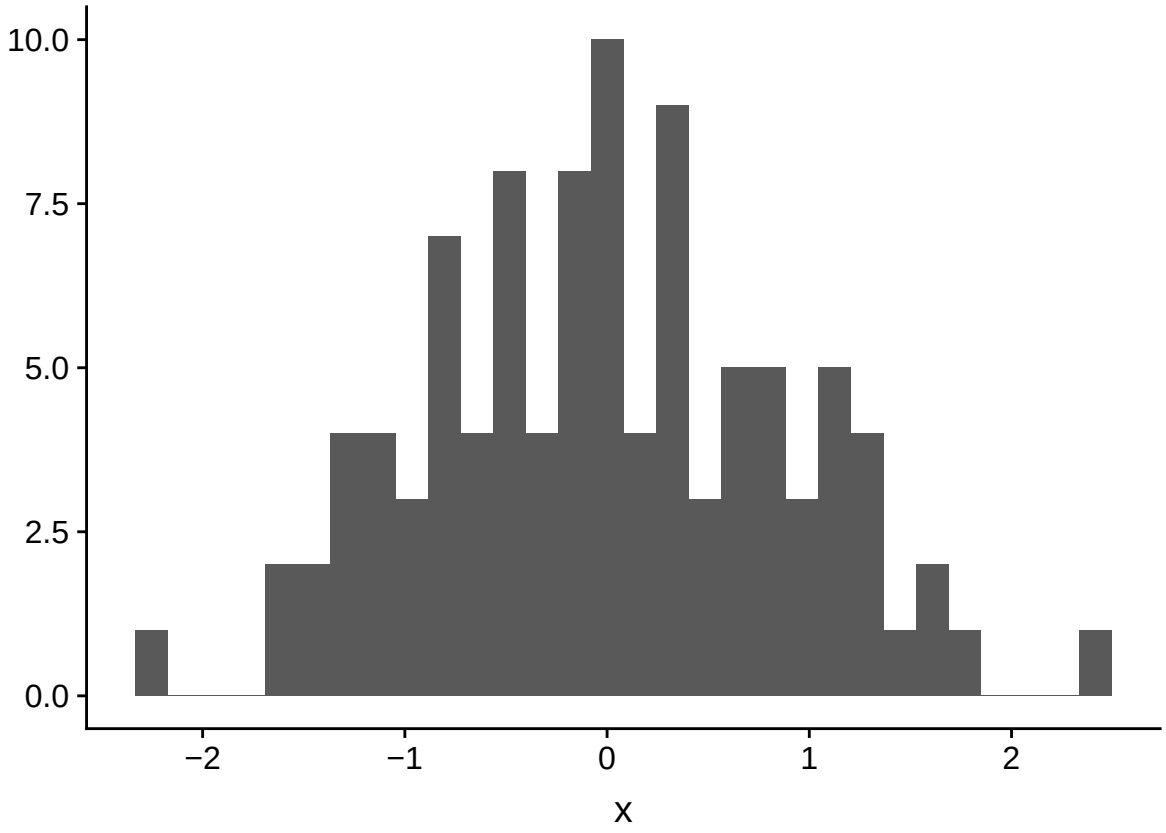
I added the chunk option "message=FALSE". Run your chunk with and without this as an option.

2.3.3 Create a "simple plot" chunk

4. Create a new chunk and label it "simple plot"
5. insert the following R script and then click the chunk's run button. Do you get a plot?

```
n <- 100
x <- rnorm(n)
qplot(x)
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



2.3.4 Create more R chunks and explore options and play with R code

Chapter 3

Data – Reading, Writing, and Fake

3.1 Create new notebook for this chapter

Be sure to save the notebook in the "notebooks" folder of your BIO_413 project. Annotate your notebook with notes! Update it as you learn more! We will use data.table for importing text files in tab-delimited or comma-separated formats and the readxl package for importing excel files.

```
library(ggplot2)
library(ggpubr)
library(data.table)
library(readxl)
library(emmeans)
library(mvtnorm)

knitr::opts_chunk$set(fig.width=6, fig.height=4)
```

3.2 Importing Data

Throughout this book, we will download data from the Dryad Digital Repository, which is a major resource for increasing reproducibility in science. My own view is that *all data* should be archived on some public server (exceptions include data that are proprietary or contain sensitive information – such as human health measures).

The downloaded data will be inserted into the "data" folder. To access these data in an R script, the script needs to know "where to look" or the "address." This address is the **directory path**. The default path for an R notebook is the directory containing the notebook .Rmd file. This file should be in the "notebooks" folder within "BIO_413". The "BIO_413" Folder is the parent of the "notebooks" folder. It is also the parent of the "data" folder. To see any content within the "data" folder, the R script needs to tell R to move back (or up) the directory structure out of the "notebooks" folder into the parent "BIO_413" folder and then forward (or down) into the "data" folder. This is done with

```
data_path <- "../data"
```

The `..` moves the address (of where to read input or write output) back one step and `/data` moves the address forward into the "data" folder. This folder will eventually contains lots of data from Dryad Digital Repository.

3.2.1 Excel File

The Excel dataset is from an experiment on the growth response of zebra finch chicks to an incubation call that presumably signals "hot environment" to the embryos (Mariette, M.M. and Buchanan, K.L., 2016. Prenatal acoustic communication programs offspring for high posthatching temperatures in a songbird. Science, 353(6301), pp.812-814). The source file is from the Dryad Repository here:

file name: "allDatasetsMarietteBuchanan2016.xls"

source: https://datadryad.org//handle/10255/dryad.122315

Steps

1. Copy the title of the Dryad page, which is "Data from: Prenatal acoustic communication programs offspring for high post-hatching temperatures in a songbird"
2. Create a new folder within "data" and paste in the copied title as the folder name
3. Remove the colon from the name, so the folder name is "Data from Prenatal acoustic communication programs offspring for high post-hatching temperatures in a songbird"
4. Download the .xls file into this folder

A .xls file is an old (pre 2007) Microsoft Excel file type. It is a binary file and can only be opened into a readable format with specialized software. The more modern Excel file type is .xlsx, which contains within it multiple xml components. An xml file is a text file, and so contains readable content, but the content is xml code to display something. In general, I am a big advocate of archiving stuff as text files (manuscripts, data, scripts, blog posts) because these will *always* be readable by future software. Microsoft Excel is not likely to die anytime soon and software that can read .xls and especially .xlsx files (again, .xlsx files are text files) is even less likely to disappear but we can feel even more confident if data are archived as text files. That said, a single microsoft excel file with multiple sheets is an efficient method for distributing data and the readxl package provides excellent tools for reading different sheets of a single .xls or .xlsx file.

The code below uses the function `read_excel()` from the package readxl. More about the amazing power of this package is the tidyverse page and chapter 11 in the *R for Data Science* book.

```
data_folder <- "Data from Prenatal acoustic communication programs offspring for high post-hatching tem
filename <- "allDatasetsMarietteBuchanan2016.xls"
file_path <- paste(data_path, data_folder, filename, sep="/")
chick <- data.table(read_excel(file_path, sheet="nestlingMass"))
head(chick) # check -- are there headers? are there the correct number of columns?
```

```
##       chick ID brood ID brood composition sex rank in nest
## 1:    N1.10LF3  N1.10m3             mixed   F            2
## 2: N1.10noCut3  N1.10m3             mixed   M            4
## 3:    N1.10RB3  N1.10m3             mixed   F            2
## 4:    N1.10RF3  N1.10m3             mixed   F            5
## 5:    N1.12LB3  N1.12m3             mixed   F            3
## 6:    N1.12LF3  N1.12m3             mixed   F            1
##    playback treatment nest temperature above ambient
## 1:              treat                       4.289583
## 2:               cont                       4.289583
## 3:               cont                       4.289583
## 4:               cont                       4.289583
## 5:               cont                       3.972917
## 6:              treat                       3.972917
##    max daily temp hatch day mean max temp hatch to day2
## 1:                      17.4                    18.83333
## 2:                      19.0                    20.53333
## 3:                      17.4                    18.83333
## 4:                      19.0                    20.53333
```

```
## 5:                     29.0                    24.63333
## 6:                     25.1                    24.80000
##    mean max temp hatch to day10 mean max temp hatch to day13 hatching mass
## 1:                         22.70                     23.05714           0.7
## 2:                         24.53                     23.41429           0.6
## 3:                         22.70                     23.05714           0.7
## 4:                         24.53                     23.41429           0.6
## 5:                         22.85                     22.91429           0.7
## 6:                         23.35                     23.24286           0.6
##              day1 mass           day2 mass         day10 mass day13 mass
## 1:  1.1000000000000001                 1.2                 NA        9.8
## 2: 0.80000000000000004  1.1000000000000001                 NA        9.1
## 3: 0.90000000000000002  1.3999999999999999                 NA        9.3
## 4:                 0.5 0.90000000000000002                 NA        7.7
## 5:                   1  1.3999999999999999 9.4000000000000004       10.1
## 6: 0.90000000000000002  1.3999999999999999 8.0999999999999996        9.6
##    day13 tarsus
## 1:        14.11
## 2:        12.90
## 3:        13.60
## 4:        13.06
## 5:        14.08
## 6:        13.46
```

NOTE

If you are getting errors when trying to read a file, it is probably a bug in the construction of the variable `file_path`, which is a string variable and the value has to be exactly match the directly path to the file you are trying to read. `file_path` is constructed by pasting together the variables `data_path`, `data_folder`, and `filename`. Type `file_path` into the console and look at the value. Then check

1. Spelling. Humans are very good at understanding misspelled words but the R language (or any computer language) is very literal. "../data" does not equal "./data" or "../ data" or "../data"
2. Capitalization. R is **case sensitive** (some programming languages are not). "../data" does not equal "../Data" or "../DATA".
3. is the file you are trying to read actually in the folder you are trying to read from?
4. is the notebook that you are writing in the folder "notebooks"? (the construction of `file_path` assumes that notebook is one folder deep within the project folder.

If the spelling or capitalization of any of these components is wrong, then `file_path` will be wrong. If there is any difference in any character in the string, then R will return an error. So spelling AND capitalization have to be perfect, not simply close. Humans are very good at understanding misspelled and OdDLy capitalized words but the R language (or any computer language) is very literal.

In this book, we will consistently uses the protocol for storing and retrieving downloaded files. The first three lines in the script above creates the directory path to the file. This path includes

1. data_path – the relative path into the folder "data" (relative to the location of the notebook file)
2. data_folder – the name of the folder within "data" containing the file
3. filename – the name of the file to read

These are all put together into a single path using the function `paste()`. Read about paste. It will be used repeatedly. The `read_excel(file_path, sheet="nestlingMass")` reads the nestlingMass sheet only. This function is embedded within the `data.table()` function and so is converted into a data.table. The

data.table is assigned to the object "chick"

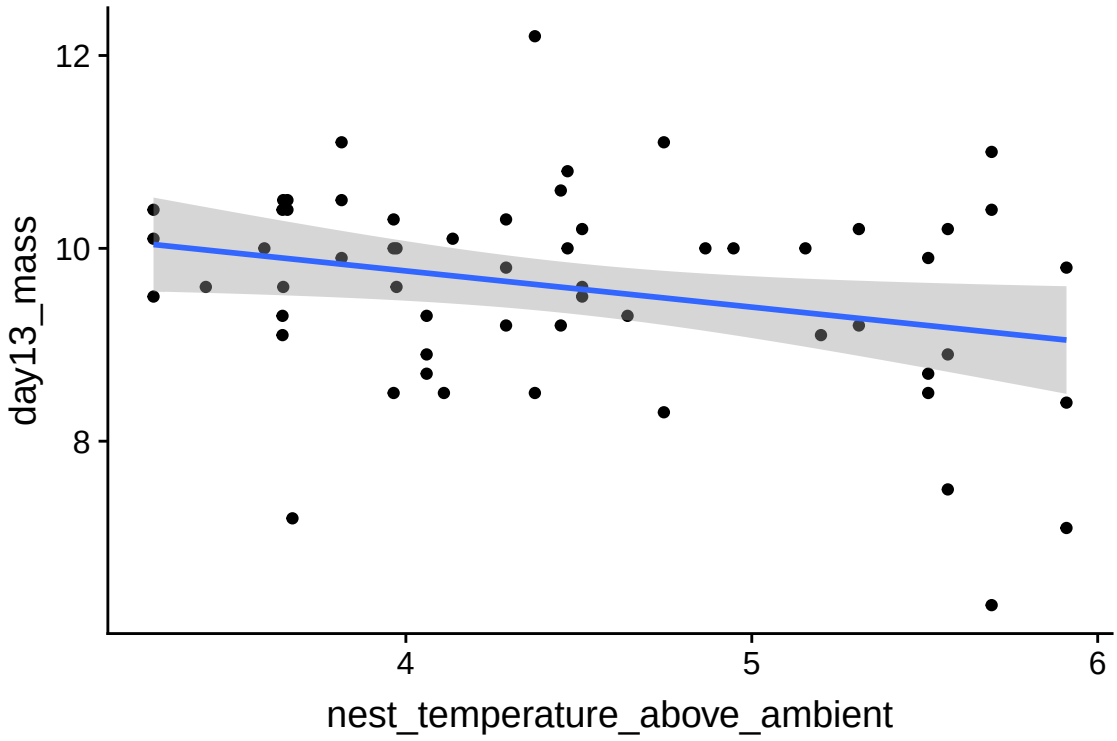
The `head(chick)` script simply displays the first few lines of the data.table. This is one way to check that the data were imported correctly. In this case, it is easy to see that the column names have spaces in them. It can sometimes be hard to work with column names with spaces and so this next line of code changes all spaces to an underscore

```
setnames(chick, old=colnames(chick), new=gsub(" ", "_", colnames(chick)))
```

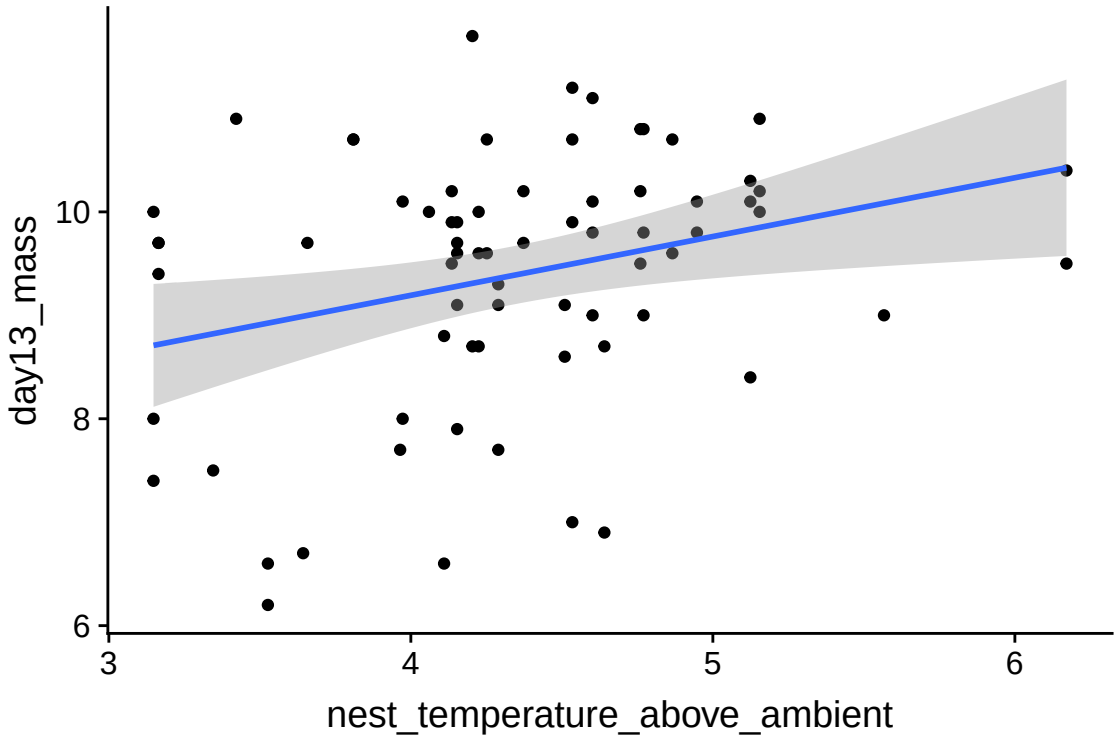
Resist the temptation to change the column names in the data file, which reduces reproducibility. Always increase reproducibility!

Just for fun, let's plot the data and reproduce Fig. 2A and B. We are using the `qplot` function, which is from the ggplot2 package. Two plots are made and only a subset of the rows are plotted in each (in A, the subset in which playback_treatment=="treat" and, in B, the subset in which playback_treatment=="cont"). This book uses the ggplot2 package extensively.

```
qplot(x=nest_temperature_above_ambient, y=day13_mass, data=chick[playback_treatment=="treat"]) +
  geom_smooth(method="lm")
```




```
qplot(x=nest_temperature_above_ambient, y=day13_mass, data=chick[playback_treatment=="cont"]) +
  geom_smooth(method="lm")
```



3.2.2 Text File

The example dataset comes from an experiment on the effect of neonicotinoid pesticides on bumble bee colony growth.

file name: "Whitehorn, O'Connor, Wackers, Goulson (2012) Data from 'Neonicotinoid pesticide reduces bumblebee colony growth and queen production'.csv.csv"

source: https://datadryad.org//resource/doi:10.5061/dryad.1805c973

Steps

1. Copy the title of the Dryad page, which is "Data from: Neonicotinoid pesticide reduces bumblebee colony growth and queen production"
2. Create a new folder within "data" and paste in the copied title as the folder name
3. Remove the colon from the name, so the folder name is "Data from Neonicotinoid pesticide reduces bumblebee colony growth and queen production"
4. Download the .csv file into this folder

A .csv file is a text file that is comma-delimted, which means that the entries of a row are separated by commas. A text file is readable by any text editor software and most other kinds of software. Datasets that are stored as text files are typically saved as either .csv (where the entries of a row are separated by commas) or .txt (where the entries are separated by tabs). The base R way to read a .csv file is using `read.csv`. The `read.table` function is more versatile, as the delimiter can be specified. The function `fread()` from the data.table package is fast, smart, and flexible. It is smart in the sense that it guesses what the delimter is. Unfortunately, because of spaces in the column labels for this file, fread guesses incorrectly (another reason why spaces in column labels should be avoided). To overcome this, the statement below specifies that the file contains a "header" (a line containing column labels)

```
data_folder <- "Data from Neonicotinoid pesticide reduces bumblebee colony growth and queen production"
filename <- "Whitehorn, O'Connor, Wackers, Goulson (2012) Data from 'Neonicotinoid pesticide reduces bu
file_path <- paste(data_path, data_folder, filename, sep="/")
```

```
bee <- fread(file_path, header=TRUE)
bee[, Treatment:=factor(Treatment, c("Control", "Low", "High"))]
head(bee)
```

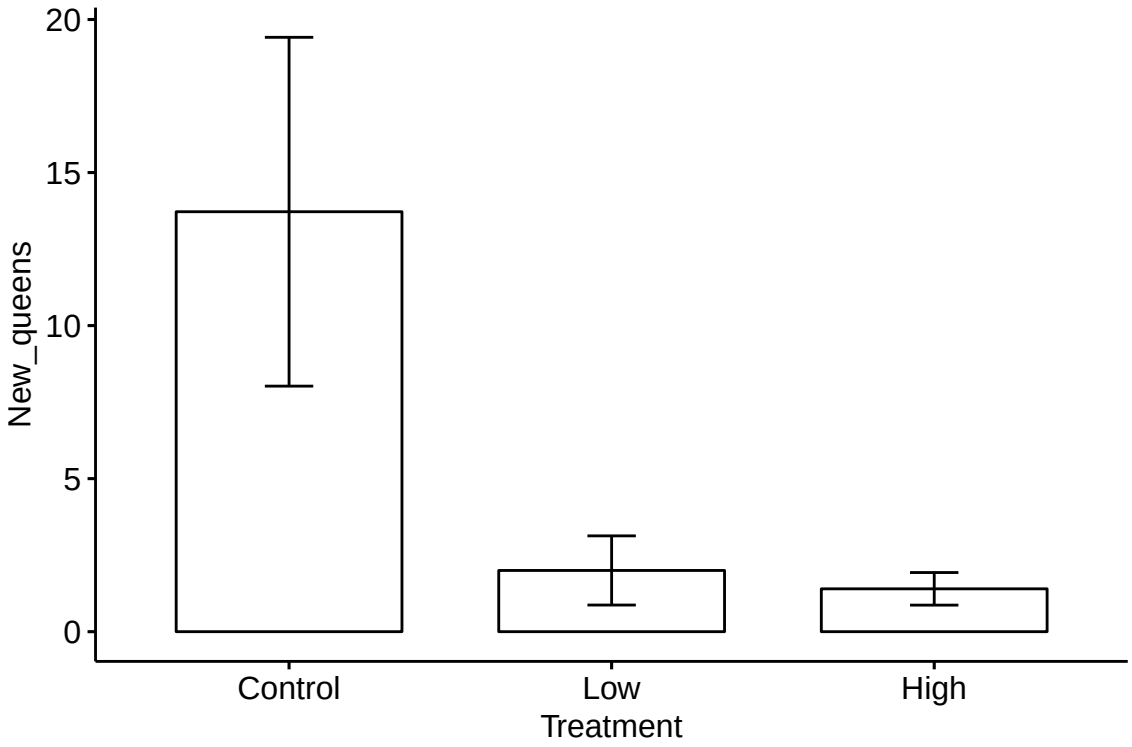
```
##    Treatment Nest ID No. workers      0      1      2   3   4    5    6
## 1:   Control      C1          13 712.95 748.30 800.57 865 966  997  850
## 2:   Control      C2          14 719.58 750.00 789.25 822 812  846  827
## 3:   Control      C3          17 704.92 736.31 767.99 837 976 1117 1050
## 4:   Control      C4          20 726.42 763.31 795.60 813 801  784   NA
## 5:   Control      C5          28 740.60 785.52 808.42 837 871  906  886
## 6:   Control      C6          15 727.10 751.90 774.80 807 847  859  827
##      7   8 V13 Workers left Males New queens Total unhatched pupae
## 1: 791 775  NA            2     0          1                    NA
## 2: 820 802  NA            6    15          0                    20
## 3: 866 808  NA            1     0          9                    NA
## 4:  NA  NA  NA            0     0          0                    12
## 5: 807 775  NA            3     0          0                    NA
## 6:  NA  NA  NA            0     0          0                   118
##    Queen pupae Empty cells
## 1:          NA          NA
## 2:           0         120
## 3:          NA          NA
## 4:           0          72
## 5:          NA          NA
## 6:          20         132
```

Here, as with the import of the Excel file, the first three lines create the directory path to the file. The treatment column is a factor variable containing three levels (Control, Low, and High). R automatically orders these alphabetically. For plotting and analysis, we might want a different order. For example, we want Control to be first in the order, since this is a natural "reference" level (what everything is compared to). And if we think of "Control" as no treatment, then it makes sense to have "Low" second in order and "Hight" last in order. The line `bee[, Treatment:=factor(Treatment, c("Control", "Low", "High"))]` re-orders these levels to this more meaningful order.

Again, there are spaces in the column names. **Here I'll leave it to you to change this**

Here is a reproduction of Fig 2.

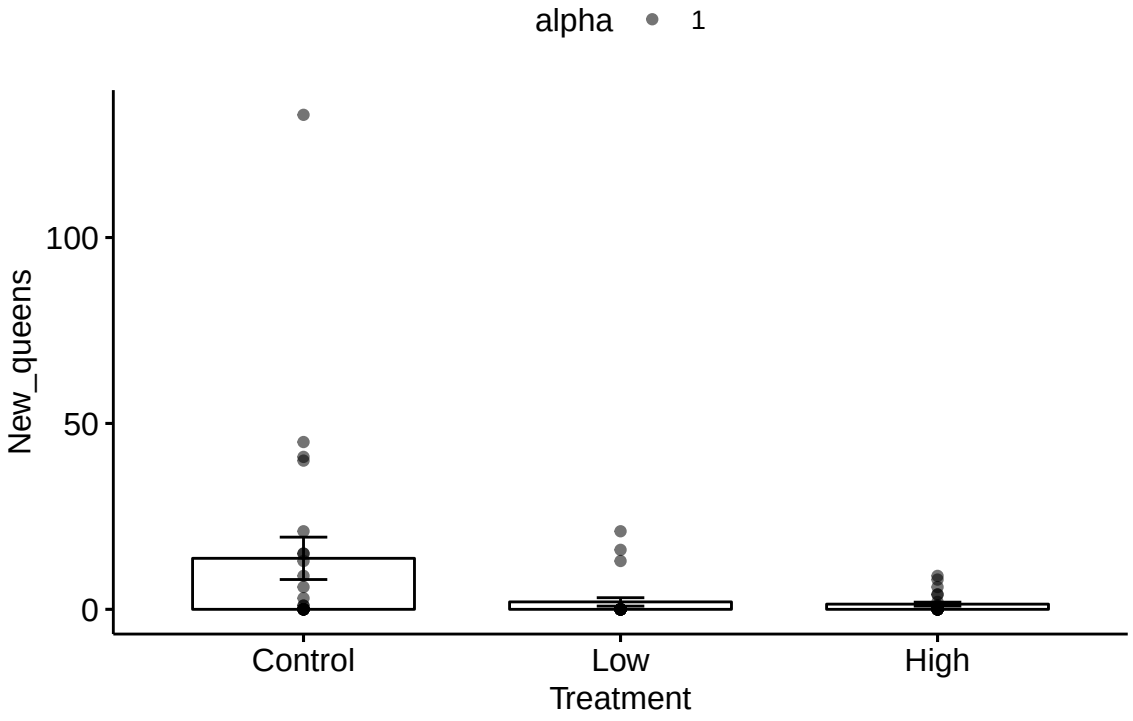
```
ggbarplot(data=bee, x="Treatment", y="New_queens", add = c("mean_se"))
```



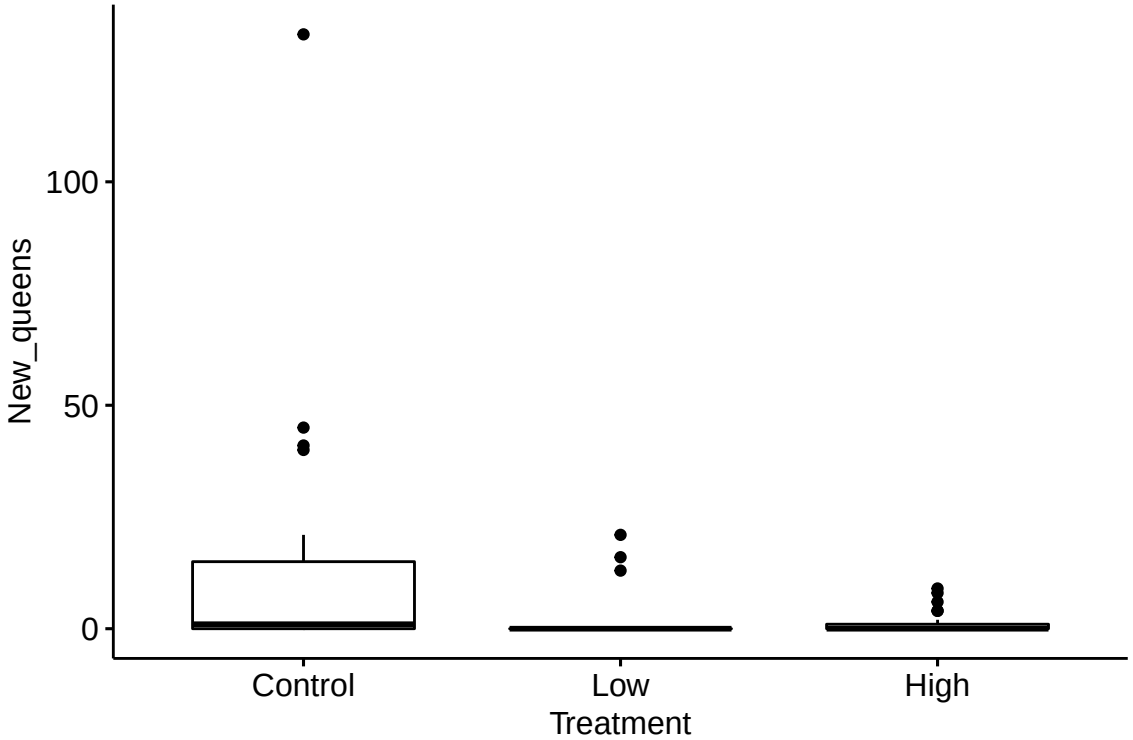
The plot suggests immediately some problems with the plot itself and the associated analysis. First, the y-axis is counts, which means that negative values are impossible. But the standard error bars look like they use standard errors computed from a model that allows infinetly large negative values, and the illustrated standard error bars imply that negative values exist. So these error bars are misleading. Second, it is good practice, especially if sample sizes are modest or small, to "show the data", which means, show the individual data points and not just a summary of the distribution.

Here are three alternative plots for exploratory purposes. The first simply "shows the data" but still uses the misleading standard error bars. The second uses a box plot. The last plots the means and 95% confidence intervals modeled with a GLM (generalized linear model) to account for the count data (the model used could be improved). Notice that the bar length above the mean is longer than the bar length below the mean (that is the interval is asymmetric about the mean). In order to stay focussed on importing data, I leave explanation of these plots and analysis to later chapters.

```
ggbarplot(data=bee, x="Treatment", y="New_queens", add = c("mean_se", "point"))
```

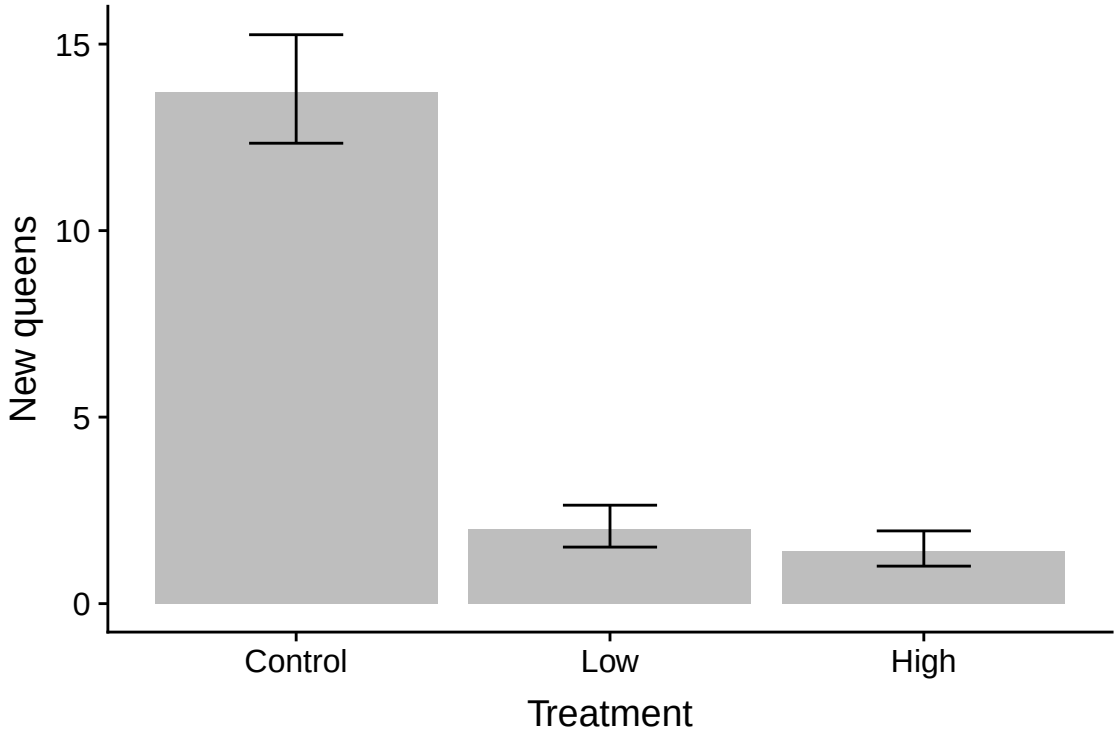



```
ggboxplot(data=bee, x="Treatment", y="New_queens")
```




```
fit.glm <- glm(New_queens ~ Treatment, data=bee, family=poisson())
means.glm <- emmeans(fit.glm, specs="Treatment", type = "response")
gg <- ggplot(data=data.frame(means.glm), aes(x=Treatment, y=rate)) +
  geom_col(fill="gray") +
  geom_errorbar(aes(x=Treatment, ymin=asymp.LCL, ymax=asymp.UCL), width=0.3) +
  ylab("New queens") +
```

```
  NULL
gg
```

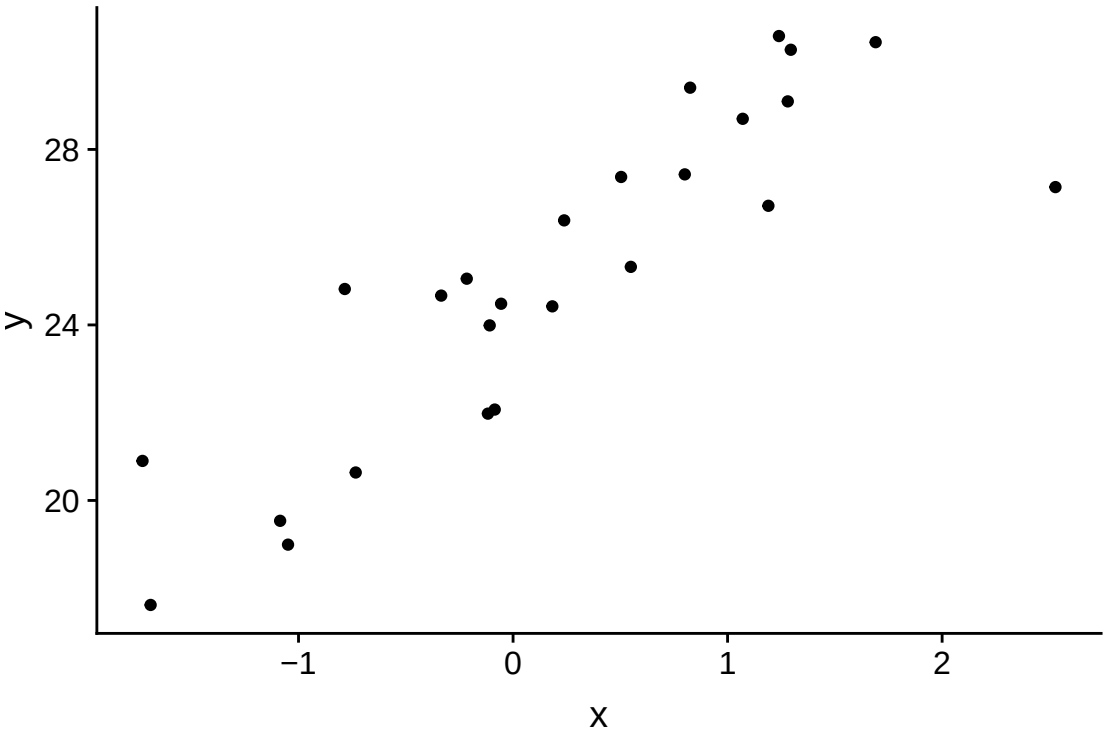


3.3 Creating Fake Data

3.3.1 Continuous X (fake observational data)

A very simple simulation of a regression model

```
n <- 25
beta_0 <- 25
beta_1 <- 3.4
sigma <- 2
x <- rnorm(n)
y <- beta_0 + beta_1*x + rnorm(n, sd=sigma)
qplot(x, y)
```




```
knitr::kable(coefficients(summary(lm(y ~ x))), digits=2)
```

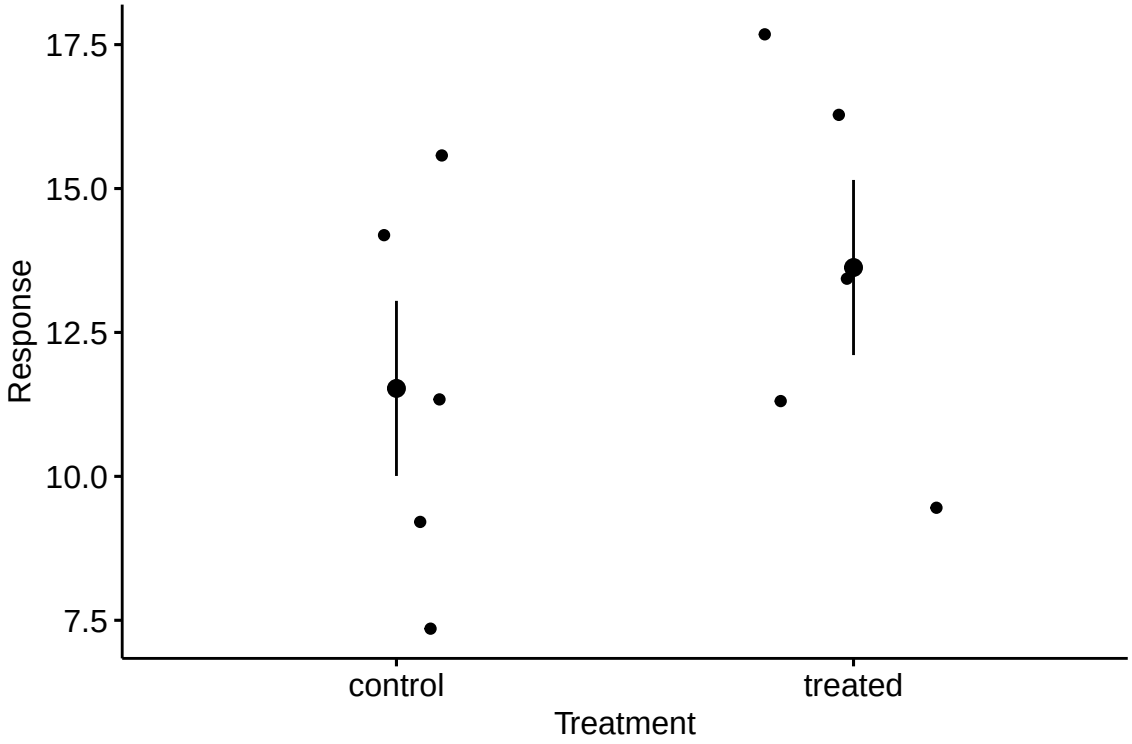
	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	24.46	0.39	62.43	0
x	3.05	0.37	8.25	0

The coefficient of x is the "Estimate". How close is the estimate? Run the simulation several times to look at the variation in the estimate – this will give you a sense of the uncertainty. Increase n and explore this uncertainty. Increase all the way up to $n = 10^5$. Commenting out the qplot line will make this exploration easier.

3.3.2 Categorical X (fake experimental data)

```
n <- 5

fake_data <- data.table(Treatment=rep(c("control", "treated"), each=n))
beta_0 <- 10.5 # mean of untreated
beta_1 <- 2.1 # difference in means (treated - untreated)
sigma <- 3 # the error standard deviation
# the Y variable ("Response") is a function of treatment. We use some matrix
# algebra to get this done.
# Turn the Treatment assignment into a model matrix. Take a peak at X!
X <- model.matrix(~ Treatment, fake_data)
# to make the math easier the coefficients are collected into a vector
beta <- c(beta_0, beta_1)
# you will see the formula Y=Xb many times. Here it is coded in R
fake_data[, Response:=X%*%beta + rnorm(n, sd=sigma)]
# plot it with a strip chart (often called a "dot plot")
ggstripchart(data=fake_data, x="Treatment", y="Response", add = c("mean_se"))
```




```
# fit using base R linear model function
fit <- lm(Response ~ Treatment, data=fake_data)
# display a pretty table of the coefficients
knitr::kable(coefficients(summary(fit)), digits=3)
```

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	11.528	1.521	7.579	0.000
Treatmenttreated	2.100	2.151	0.976	0.358

Check that the intercept is close to beta_0 and the coefficient for Treatment is close to beta_1. This coefficient is the different in means between the treatment levels. It is the simulated effect. Again, change n. Good values are $n = 20$ and $n = 100$. Again, comment out the plot line to make exploration more efficient.

3.4 Saving Data

Let's save the fake data to the "Fake_Data" folder. In the "output" folder create a new folder named "week 01". Then set the path to the output folder:

```
output_path <- "../output" # out to parent directory than down into Fake_data
```

This could be done at the beginning of the notebook, especially if many output files are saved. Regardless, now complete the file_path with the specifics of this save.

```
data_folder <- "week 01"
filename <- "my_first_fake_data.txt"
file_path <- paste(output_path, data_folder, filename, sep="/")
write.table(fake_data, file_path, sep="\t", quote=FALSE)
```

We used `write.table()` to create a tab-delimited text file using `sep="\t"` to specify tabs to separate the row elements. "$\widehat{\ }$" is the standard character string for a tab. Check in your Fake_Data folder and open the file in a text editor.

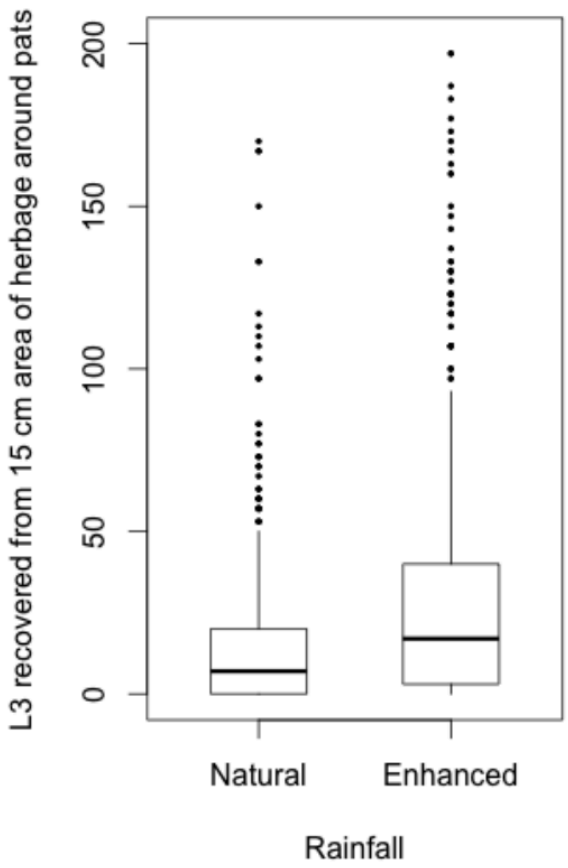


Figure 3.1: Fig. 1 from "Dung beetles reduce livestock…"

3.5 Problems

1. Download the dataset "data-Lodjak.et.al-2016-FuncEcol.xlsx" from the Dryad repository at https://datadryad.org/resource/doi:10.5061/dryad.rd01s. The .xlsx file presents the data cleanly but the trade-off is that the 1) multiple header rows, and 2) spaces in the header labels, 3) parentheses in the header labels make it more complex to import in a usable way. Import the data and plot Body Mass against Age (that is make Body Mass the "Y" variable and Age the "X" variable) using the qplot function. You should recode the column labels to remove spaces and parentheses using the setnames function.
2. Download the dataset "Results2015.txt" from the Dryad repository at https://datadryad.org//resource/doi:10.5061/dryad.65vk4. Try to reproduce Fig. 1. It's not easy. I've inserted the figure below.
3. (grad students only) Download and plot data from a Dryad Repository dataset of your choice.
4. (grad students only) Create fake experimental data with three treatment levels (control, lo_temp, high_temp). This will require three parameters: an intercept (beta_0), an effect of lo_temp (beta_1), and an effect of high_temp (beta_2). You should be able to plug and play from the script above even if you don't underdstand at this point what any of it is! Plot it as a strip chart, as above.

Chapter 4

Variability and Uncertainty (Standard Deviations and Standard Errors)

Uncertainty is the stuff of science. A major goal of statistics is measuring uncertainty. What do we mean by uncertainty? Uncertainty is the error in estimating a parameter, such as the mean of a sample, or the difference in means between two experimental treatments, or the predicted response given a certain change in conditions. Uncertainty is measured with a **variance** or its square root, which is a **standard deviation**. The standard deviation of a statistic is also (and more commonly) called a **standard error**.

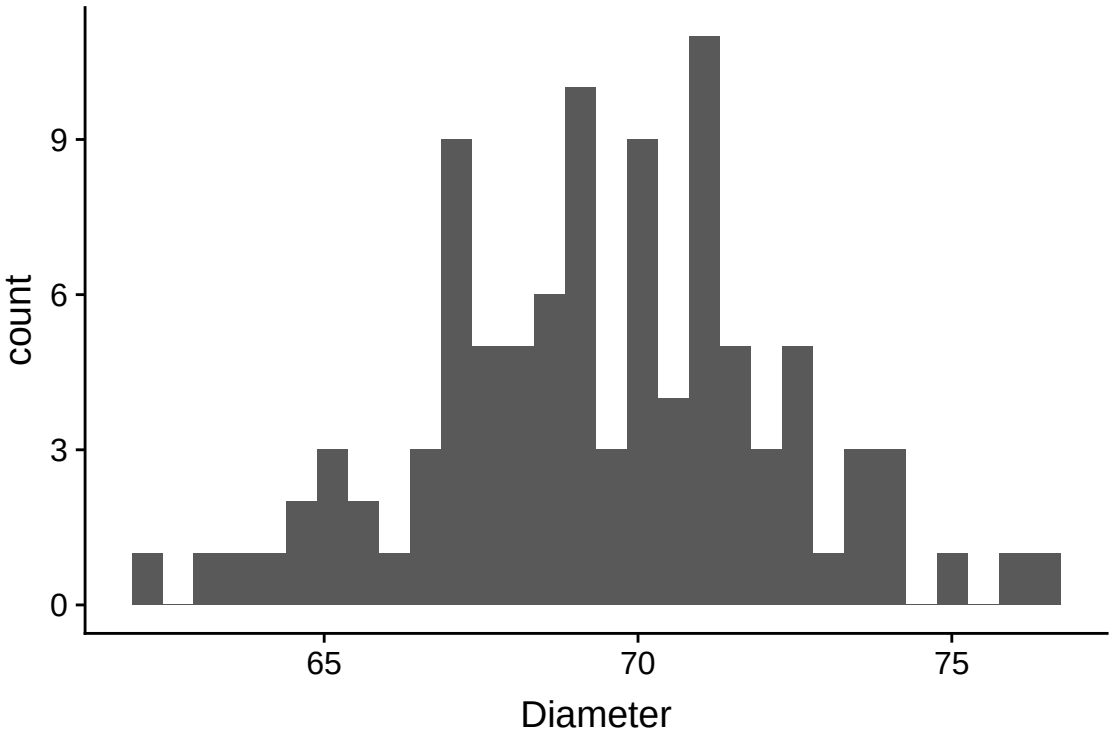
Uncertainty emerges because of variability. In any introductory statistics class, students are introduced to two measures of variability, the "standard deviation" and the "standard error." These terms are absolutely fundamental to statistics – they are the start of everything else. Yet, many biology professors confuse these terms and certainly, introductory students do too.

When a research biologist uses the term "standard deviation," they are probably referring to the sample standard deviation which is a measure of the variability of a sample. When a research biologist uses the term "standard error," they are probably referring to the standard error of a mean, but it could be the standard error of another statistics, such as a regression slope. An important point to remember and understand is that all standard errors *are* standard deviations. This will make more sense soon.

4.1 The sample standard deviation vs. the standard error of the mean

4.1.1 Sample standard deviation

The sample standard deviation is a measure of the variability of a sample. For example, were we to look at a histological section of skeletal muscle we would see that the diameter of the fibers (the muscle cells) is variable. We could use imaging software to measure the diameter of a sample of 100 cells and get a **distribution** like this



The mean of this sample is 69.4 and the standard deviation is 2.8. The standard deviation is the square root of the variance, and so computed by

$$s_y = \sqrt{\frac{\sum_{i=1}^{n} (y_i - \overline{y})^2}{n-1}} \tag{4.1}$$

Memorize this equation. To understand the logic of this measure of variability, note that $y_i - \overline{y}$ is the **deviation** of the ith value from the sample mean, so the numerator is the sum of squared deviations. The numerator is a sum over n items and the denominator is $n - 1$ so the variance is (almost!) an averaged squared deviation. More variable samples will have bigger deviations and, therefore, bigger average squared deviations. Since the standard deviation is the square root of the variance, a standard deviation is the square root of an average squared deviation. This makes it similar in value to the averaged deviation (or average of the absolute values of the deviations since the average deviation is, by definition of a mean, zero).

Notes on the variance and standard deviation

1. Variances are additive but standard deviations are not. This means that the variance of the sum of two independent (uncorrelated) random variables is simply the sum of the variances of each of the variables. This is important for many statistical analyses.
2. The units of variance are the square of the original units, which is awkward for interpretation. The units of a standard deviation is the same as that of the original variable, and so is much easier to interpet.
3. For variables that are approximately normally distributed, we can map the standard deviation to the quantiles of the distribution. For example, 68% of the values are within one standard deviation of the mean, 95% of the values are within two standard deviations, and 99% of the values are within three standard deviations.

4.1.2 Standard error of the mean

A standard error of a statistic is a measure of the precision of the statistic. The standard error of the mean is a measure of the precision of the estimate of the mean. The smaller the standard error, the more precise the estimate. The standard error of the mean (SEM) is computed as

$$SEM = \frac{s_y}{\sqrt{n}} \tag{4.2}$$

The SEM is often denoted $s_{\bar{y}}$ to indicate that it is a standard deviation of the mean ($\bar{y}$). In what sense is a standard error a measure of variability? This is kinda weird. If we sample 100 cells in the slide of muscle tissue and compute the mean diameter, how can the mean have a standard deviation? There is only one value! To understand how the SEM is a standard deviation, imagine 1) resampling 100 cells and 2) recomputing a mean from the re-sampled data an infinite number of times and each time, you write down the newly computed mean. The true standard error of the mean is the standard deviation of this infinitely long column of means. This means that a standard error of the mean, computed from a single sample using equation (4.2) is itself a sample statistic.

Notes on standard errors

1. The SEM is only one kind of standard error. A standard deviation can be computed for any statistic – these are all standard errors. For some statistics, such as the mean, the standard error can be computed directly using an equation, such as that for the SEM (equation (4.2). For other statistics, a computer intensive method such as the **bootstrap** is necessary to compute a standard error. We will return to the bootstrap at the end of this chapter.
2. The units of a standard error are the units of the measured variable.
3. A standard error is proportional to sample variability (the sample standard deviation, s_y) and inversely proportional to sample size (n). Sample variability is a function of both natural variation (there really is variation in diameter among fibers in the quadriceps muscle) and measurement error (imaging software with higher resolution can measure a diameter with less error). Since the SEM is a measure of the precision of estimating a mean, this means this precision will increase (or the SEM will decrease) if 1) an investigator uses methods that reduce measurement error and 2) an investigator computes the mean from a larger sample.
4. This last point (the SEM decreases with sample size) seems obvious when looking at equation (4.2), since n is in the denominator. Of course n is also in the denominator of equation (4.1) for the sample standard deviation but the standard deviation does not decrease as sample size increases. First this wouldn't make any sense – variability is variability. A sample of 10,000 cell diameters should be no more variable than a sample of 100 cell diameters (think about if you agree with this or not). Second, this should also be obvious from equation (4.1). The standard deviation is the square root of an average and averages don't increase with the number of things summed since both the the numerator (a sum) and denominator increase with n.

4.2 Using Google Sheets to generate fake data to explore uncertainty

In statistics we are interested in estimated parameters of a **population** using measures from a **sample**. The goal in this section is to use Google Sheets (or Microsoft Excel) to use fake data to discover the behavior of sampling and to gain some intuition about uncertainty using standard errors.

4.2.1 Steps

1. Open Google Sheets

2. In cell A1 type "mu". mu is the greek letter μ and is very common notation for the poplation value (the TRUE value!) of the mean of some hypothetical measure. In cell B1, insert some number as the value of μ. Any number! It can be negative or positive.
3. In cell A2 type "sigma". sigma is the greek letter σ. σ^2 is very common (universal!) notation for the population (TRUE) variance of some measure or parameter. Notice that the true (population) values of the mean and variance are greek letters. This is pretty standard in statistics. In cell B2, insert some positive number (standard deviations are the positive square roots of the variance).
4. In cell A8 type the number 1
5. In cell A9 insert the equation "=A8 + 1". What is this equation doing? It is adding the number 1 to to the value in the cell above, so the resulting value should be 2.
6. In Cell B8, insert the equation "=normsinv(rand())*B2 + B1". The first part of the equation creates a random normal variable with mean 0 and standard deviation 1. multiplication and addition transform this to a random normal variable with mean μ and standard deviation σ (the values you set in cells B1 and B2).
7. copy cell B8 and paste into cell B9. Now Higlight cells A9:B9 and copy the equations down to row 107. You now have 100 random variables sampled from a infinite population with mean μ and standard deviation σ.
8. In cell A4 write "mean 10". In cell B4 insert the equation "=average(B8:B17)". The resulting value is the **sample mean** of the first 10 random variables you created. Is the mean close to μ?
9. In cell A5 write "sd 10". In cell B5 insert the equation "stdev(B8:B17)". The result is the **sample standard deviation** of the first 10 random variables. Is this close to σ?
10. In cell A6 write "mean 100". In cell B6 insert the equation "=average(B8:B107)". The resulting value is the **sample mean** of the all 100 random variables you created. Is this mean closer to μ than mean 10?
11. In cell A7 write "sd 100". In cell B7 insert the equation "=stdev(B8:B107)". The resulting value is the **sample standard deviation** of the all 100 random variables you created. Is this SD closer to σ than sd 10?

The sample standard deviation is a measure of the variability of the sample. The more spread out the sample (the further each value is from the mean), the bigger the sample standard deviation. The sample standard deviation is most often simply known as "The" standard deviation, which is a bit misleading since there are many kinds of standard deviations!

Remember that your computed mean and standard deviations are estimates computed from a sample. They are estimates of the true values μ and σ. Explore the behavior of the sample mean and standard deviation by re-calculating the spreadsheet. In Excel, a spreadsheet is re-calculated by simultaneously pressing the command and equal key. In Google, command-R recalculates but is painfully slow. Instead, if using Google Sheets, just type the number 1 into a blank cell, and the sheet recalculates quickly. Do it again. And again.

Each time you re-calculate, a new set of random numbers are generated and the new means and standard deviations are computed. Compare mean 10 and mean 100 each re-calculation. Notice that these estimates are variable. They change with each re-calculation. How variable is mean 10 compared to mean 100? The variability of the estimate of the mean is a measure of **uncertainty** in the estimate. Are we more uncertain with mean 10 or with mean 100? This variability is measured by a standard deviation. This **standard deviation of the mean** is also called the **standard error of the mean**. Many researchers are loose with terms and use "The" standard error to mean the standard error of the mean, even though there are many kinds of standard errors. In general, "standard error"" is abbreviated as "SE." Sometimes "standard error of the mean" is specifically abbreviated to "SEM."

The standard error of the mean is a measure of the precision in estimating the mean. The smaller the value the more precise the estimate. The standard error of the mean *is* a standard deviation of the mean. This is kinda weird. If we sample a population one time and compute a mean, how can the mean have a standard deviation? There is only one value! And we compute this value using the sample standard deviation: $SEM = \frac{SD}{\sqrt{N}}$. To understand how the SEM is a standard deviation, Imagine recalculating the spread sheet an infinite number of times and each time, you write down the newly computed mean. The standard error of the mean is the standard deviation of this infinitely long column of means.

4.3 Using R to generate fake data to explore uncertainty

note that I use "standard deviation" to refer to the sample standard deviation and "standard error" to refer to the standard error of the mean (again, we can compute standard errors as a standard deviation of any kind of estimate)

4.3.1 part I

In the exercise above, you used Google Sheets to generate p columns of fake data. Each column had n elements, so the matrix of fake data was $n \times m$ (it is standard in most fields to specify a matrix as rows by columns). This is *much* easier to do in R and how much grows exponentially as the size of the matrix grows.

To start, we just generate a $n \times p$ matrix of normal random numbers.

```
# R script to gain some intuition about standard deviation (sd) and standard error (se)
# you will probably need to install ggplot2 using library(ggplot2)
n <- 6 # sample size
p <- 100 # number of columns of fake data to generate
fake_data <- matrix(rnorm(n*p, mean=0, sd=1), nrow=n, ncol=p) # create a matrix
```

the 3rd line is the cool thing about R. In one line I'm creating a dataset with n rows and p columns. Each column is a sample of the standard normal distribution which by definition has mean zero and standard deviation of 1. But, and this is important, any sample from this distribution will not have exactly mean zero and standard deviation of 1, because it's a sample, the mean and standard deviation will have some small errror from the truth. The line has two parts to it: first I'm using the function "rnorm" (for random normal) to create a vector of n*m random, normal deviates (draws from the random normal distribution) and then I'm organizing these into a matrix (using the function "matrix")

To compute the vector of means, standard deviations, and standard errors for each column of `fake_data`, use the `apply()` function.

```
means <- apply(fake_data,2,mean) # the apply function is super useful
sds <- apply(fake_data,2,sd)
sems <- sds/sqrt(n)
```

`apply()` is a workhorse in many R scripts. Learn it. Know it. Live it.

The SEM is the standard deviation of the mean, so let's see if the standard deviation of the means is close to the true standard error. We sampled from a normal distribution with SD=1 so the true standard is

```
1/sqrt(n)
```

```
## [1] 0.4082483
```

and the standard deviation of the p means is

```
sd(means)
```

```
## [1] 0.3731974
```

Questions

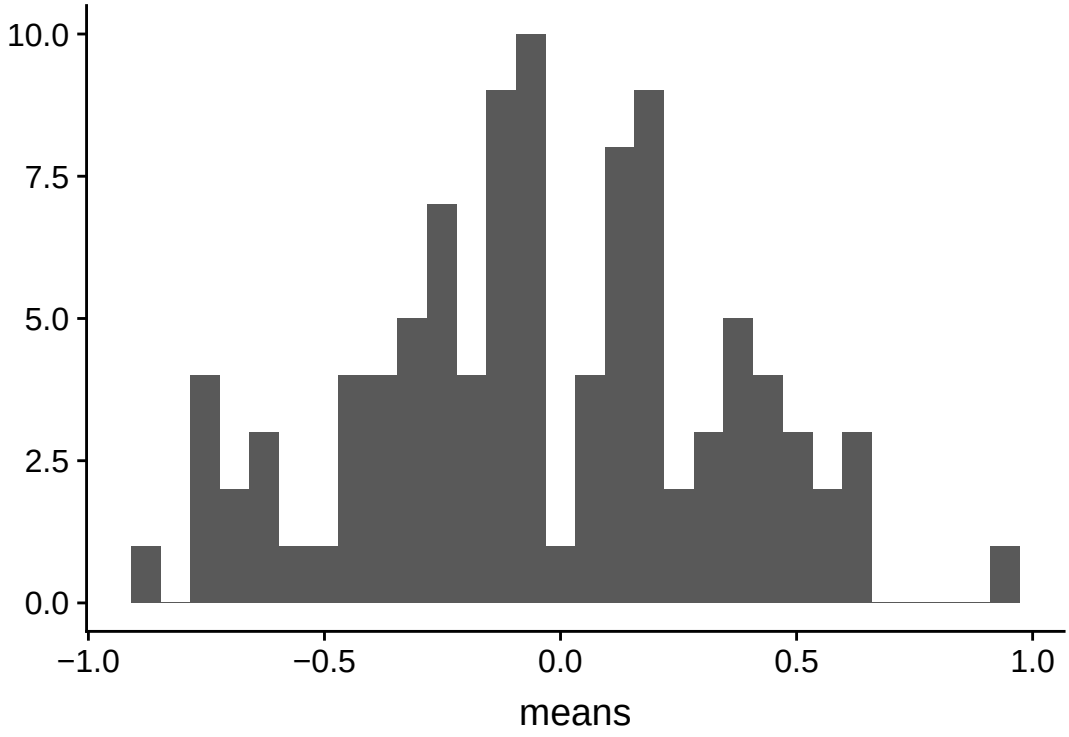
1. how close is `sd(means)` to the true SE?
2. change p above to 1000. Now how close is sd(means) to the true SE?
3. change p above to 10,000. Now how close is sd(means) to the true SE?

4.3.2 part II - means

This is a visualization of the spread, or variability, of the sampled means

```
qplot(means)
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



Compute the mean of the means

```
mean(means)
```

```
## [1] -0.039961
```

Questions

1. Remember that the true mean is zero. How close, in general, are the sampled means to the true mean. How variable are the means? How is this quantified?
2. change n to 100, then replot. Are the means, in general, closer to the true mean? How variable are the means now?
3. Is the mean estimated with $n = 100$ closer to the truth, in general, then the mean estimated with $n = 6$?
4. Redo with $n = 10000$

4.3.3 part III - how do SD and SE change as sample size (n) increases?

```
mean(sds)
```

```
## [1] 1.017144
```

Questions

1. what is the mean of the standard deviations when n=6 (set p=1000)

2. what is the mean of the standard deviations when n=100 (set p=1000)
3. when n = 1000? (set p=1000)
4. when n = 10000? (set p=1000)
5. how does the mean of the standard deviations change as n increases (does it get smaller? or stay about the same size)
6. repeat the above with SEM

```
mean(sems)
```

```
## [1] 0.4152472
```

Congratulations, you have just done a Monte Carlo simulation!

4.3.4 Part IV – Generating fake data with "for loops"

```
n <- 6 # sample size
n_iter <- 10^5 # number of iterations of loop (equivalent to p)
means <- numeric(n_iter)
sds <- numeric(n_iter)
sems <- numeric(n_iter)
for(i in 1:n_iter){
  y <- rnorm(n) # mean=0 and sd=1 are default so not necessary to specify
  means[i] <- mean(y)
  sds[i] <- sd(y)
  sems[i] <- sd(y)/sqrt(n)
}
sd(means)
```

```
## [1] 0.4090702
```

```
mean(sems)
```

```
## [1] 0.3883867
```

Questions

1. What do `sd(means)` and `mean(sems)` converge to as `n_iter` is increased from 100 to 1000 to 10,000?
2. Do they converge to the same number?
3. Should they?
4. What is the correct number?

Question number 4 is asking what is E(SEM), the "expected standard error of the mean". There is a very easy formula to compute this. What is it?

4.4 Bootstrapped standard errors

The bootstrap is the best invention since duct tape. Really.

A standard error is the standard deviation of an infinite number of hypothetically re-sampled means. A bootstrap standard error of a statistic is the standard deviation of the statistic from a finite number of resamples of the data! Wait what?

Let's download some data to explore this concept. The data are archived at Dryad Repository.

1. URL: https://datadryad.org//resource/doi:10.5061/dryad.31cc4
2. file: RSBL-2013-0432 vole data.xlsx
3. sheet: COLD VOLES LIFESPAN

The data are the measured lifespans of the short-tailed field vole (*Microtus agrestis*) under three different experimental treatments: vitamin E supplementation, vitamin C supplementation, and control (no vitamin supplementation). Vitamins C and E are antioxidants, which are thought to be protective of basic cell function since they bind to the cell-damaging reactive oxygen species that result from cell metabolism.

I've read in the file using read_excel and converted to a data.table named `vole`. I used `setnames` to rename the columns to lifespan, control, vitamin_E, and vitamin_C. The data are in a **wide format** – that is instead of a single "treatment" column, there are three columns ("control", "vitamin C", "vitamin E") with value = 1, if that row (or lifespan) was assigned the treatment of the column label and zero otherwise. In general, we want data.tables to be in long format. Wide formats can be useful for some computations but not really for these data.

Compute the standard error of the mean of the lifespan for the control group using equation (4.2). One simple way to do this for the control group is to extract the subset of the data satisfying the condition control = 1 (the value in the column "control" equals 1). In R, these conditional querries use `==`.

```
y <- na.omit(vole[control==1, lifespan]) # subset of data satisfying condition and omitting missing dat
n <- length(y) # the sample size
se <- sd(y)/sqrt(n-1) # standard error of the mean
```

Okay, the SEM using equation (4.2) is 31.9. Let's compare this with a bootstrap estimate of the SEM.

```
n_iter <- 2000 # number of bootstrap iterations
means <- numeric(n_iter) # we will save the means each iteration to this
inc <- 1:n # the first sample is the actual sample
for(iter in 1:n_iter){ # the for loop
  means[iter] <- mean(y[inc])
  inc <- sample(1:n, replace=TRUE) # re-sample for the next iteration
}
se_boot <- sd(means)

#compare
se
```

```
## [1] 31.89536
```

```
se_boot
```

```
## [1] 30.93843
```

dayum!

Chapter 5

A linear model with a single, continous X

5.1 A linear model with a single, continous X is classical "regression"

To introduce modeling with a single continous X variable, I'll use data from

1. Source: Dryad Digital Repository. https://doi.org/10.5061/dryad.b3h4q
2. File: "FCM data dryad.csv"

The data are from **?**, who showed that North American red squirrel (*Tamiasciurus hudsonicus*) mothers from Yukon, Alaska produce faster growing pups in years with increased squirrel density. Remarkably, they even showed that perceived (but not actual) density results in faster growing pups. To begin to investigate how pregnant mothers control the future growth rate of pups, the researchers measured the relationship between local squirrel density and the amount of fecal cortisol metabolites from pregnant mothers. Cortisol is a hormone that is secreted as part of stress response. The researchers were interested in cortisol because it had previously been shownt that, in mammals, blood cortisol levels in pregnant mothers have numerous effects on offspring long past birth. If increased squirrel density causes increased blood cortisol levels then we would expect to find a positive relationship between *Density* and

Figure 5.1 is a **scatterplot** of the data with the amount of cortisol metabolites in the feces on the Y axis and local squirrel density on the X axis. The line through the data is a graphical representation of a linear model fit to the data and the gray cloud around the line is a graphical representation of the uncertainty in the model. The researchers wanted to model the "effect" of squirrel density on the amount of cortisol metabolites in the feces of the pregnant mothers. Graphically, this effect is the slope of the line in Figure 5.1.

The model fit to the data is

$$FCM_i = \beta_0 + \beta_1 Density_i + \varepsilon_i \tag{5.1}$$

which contains both the linear predictor and the error. For inference, for example, computing standard errors of the coefficients, We need to model the error. Here, we use the simplest model of error which is "IID $N(0, \sigma)$". This means, the modeled error is

1. Independent – individual error values are independent of other values.
2. Identical – individual error can be thought of as a sample from a single **random distribution** (the same for each individual value). For this model, this distribution is

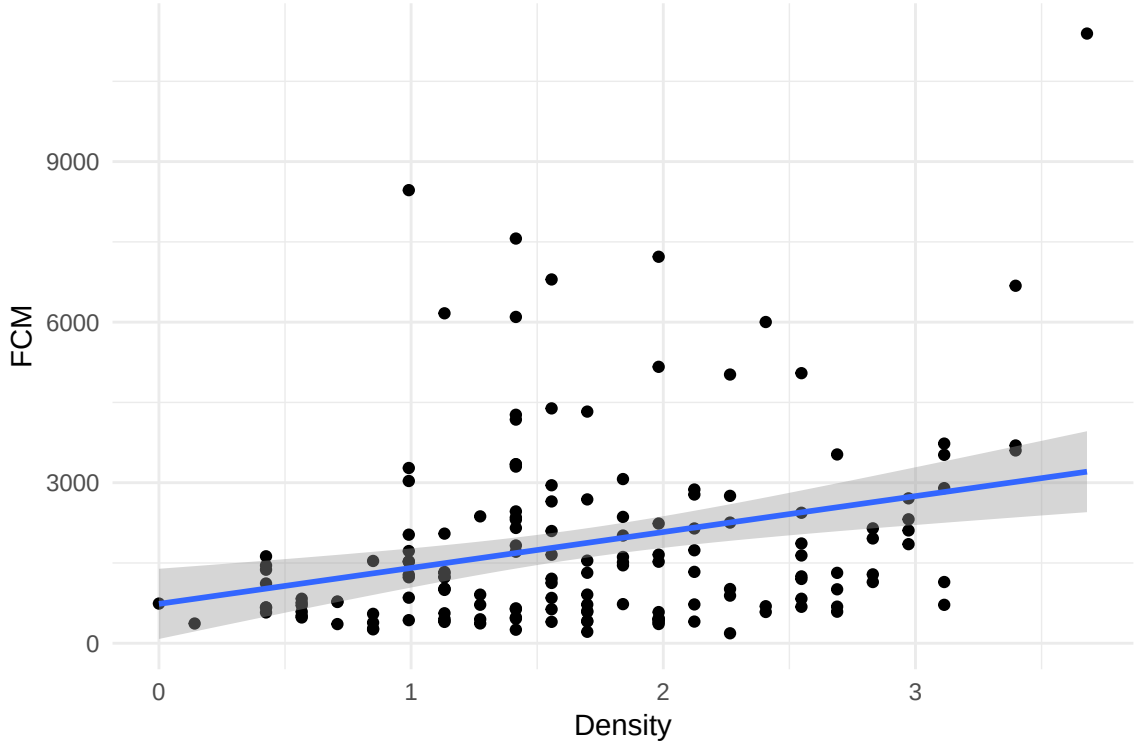


Figure 5.1: A scatterplot of Fecal cortisol matabolites and squirrel density.

3. $N(0, \sigma)$ – the modeled distribution is "Normal" or "Gaussian", with a mean of zero and a standard deviation of σ.

The predictor part of the model is

$$\mathrm{E}[FCM|Density] = \beta_0 + \beta_1 Density \tag{5.2}$$

In words, model (5.2) reads "the expected value of FCM conditional on density is beta-knot plus beta-one times density". An **expected value** is a long run average – if we were to sample lots and lots of red squirrel populations with $Density = x$ (where x is a specific value), we'd expect the average FCM across these samples to be $\beta_0 + \beta_1 x$.

> Let's unpack this. $\mathrm{E}[Y]$ is the **expectation** or **expected value** of Y. An expection is the long-run average of Y if we were to run an experiment or re-sample a population many times. The sample mean of Y is an estimate of $\mathrm{E}[Y]$. $\mathrm{E}[Y|X]$ is a conditional expectation of Y – it is the expectation given additional conditions. Using the red squirrel example, these conditions are a specific value of $Density$. If FCM is linearly related to $Density$ (the right-hand side of equation (5.2)) then the expected value of FCM given a local density of 2.8 squirrels differs from the expected value of FCM given a local density of 1.4 squirrels (the units of $Density$ are squirrels per 150 meter radius of the individual female's midden).

In model (5.2), there is a single X variable (FCM). While the X variables are often called the "dependent" variables, in this model FCM does not "depend" on the independent variable $Density$ in any causal sense – meaning if I were to intervene and set $Density$ to some value x, I would expect FCM to equal $\beta_0 + \beta_1 x$. Rather, FCM only "depends" on $Density$ in a probablistic sense – if $Density = x$ then the most probable value of FCM is $\beta_0 + \beta_1 x$. With some strong assumptions model (5.2) can be turned into a model of causal dependency, which is the focus of chapter xxx.

β_0 and β_1 are the **parameters** of model (5.2). Specifically β_0 is the model **intercept** and β_1 is the modeled **effect** of *Density*. Again, the effect (β_1) has a probabilistic, and not causal, interpretation. This interpretation is

$$\beta_1 = \mathrm{E}[FCM|Density = x + 1] - \mathrm{E}[FCM|Density = x] \tag{5.3}$$

Or, in words, "beta-1 is the expected value of FCM when density equals x + 1 minus the expected value of FCM when the density equals x." β_1 is simply the difference in expected values given a one unit difference in *Density*. A very short way to state this is "β_1 is a difference in conditional means".

5.1.1 Using a linear model to estimate explanatory effects

The goal of the statistical model here is to estimate β_1 – the probabalistic effect of *Density* on FCM. This estimate, and a measure of the uncertainty of this estimate, are in the table of coefficients of the fit model

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	736.0	331.9	2.2	0.0281
Density	671.1	178.9	3.8	0.0002

where the entries in the column "Estimate" are estimates of the parameters β_0 and β_1 in model (5.2). The entries in the column "Std. Error" are the standard errors (SE) of the estimates, which are measures of the uncertainty of the estimates.

The parameter estimates in the table above are the coefficients of the fitted model

$$FCM_i = b_0 + b_1 Density_i + e_i \tag{5.4}$$

where the subscript i refers to the ith individual. The coefficients b_0 and b_1 are the y-intercept and the slope of the line in Figure 5.1. The coefficient for *Density* (b_1) is 671.1, and (given the definition of the parameter β_1 in equation (5.3)) we expect squirrel mothers with a local density of 2 squirrels within a 150 m radius of her midden to average 671.1 more units of FCM (ng of fecal cortical metabolites per gram dry food) than mother squirrels with a local density of only 1 squirrel within a 150 m radius of her midden.

5.1.1.1 Probabilistic vs. causal conditioning

Remember that this coefficient is estimating a probabilistic parameter. Consequently, the coefficient b_1 is simply a descriptor of a pattern of relationship between local density and fecal cortisol metabolites - no causal effect is implied. With the strong assumptions explained in chapter xxx, however, b_1 can estimate a causal effect.

5.1.2 Using a linear model for prediction

Model (5.4) gives the measured value of FCM for each squirrel. The equation includes the linear predictor ($b_0 + b_1 Density_i$) and the **residual** from the predictor (e_i). The predictor part is called "predictor" because it is the equation for predicting the value of an individual's FCM given that individual's value of *Density*:

$$\widehat{FCM} = b_0 + b_1 Density \tag{5.5}$$

where $\widehat{FCM}$ is read as "FCM hat" and is the **predicted value** or simply "prediction". Very often, we use the predictor part (equation (5.5)) to predict unknown or future values given different modeled inputs (the X).

5.1.3 Reporting results

The authors of the squirrel fcm data published a figure and table similar to fig. xxx and table above but used a slightly more complex linear model. Here is how the author's reported the results:

> Across 6 years (2006 to 2011), we found a positive relationship between local density and concentrations of fecal cortisol metabolites [FCM; $t_1 55 = 3.63$, P = 0.0002 (table S4 and Fig. 3A)].

I would advocate reporting the estimate and a confidence interval instead of t and p. For example "Across 6 years (2006 to 2011), the probabilistic effect of local density on fecal cortisol metabolites is 671.1 (95% CI: 317.7, 1024.5). If a p-value is report *in addition* to the effect and CI, always report the exact p-value, which emphasizes the continuous nature of evidence against the null, and not something like"$p < 0.05$", which artificially dichotomizes the evidence against the null.

5.2 Working in R

5.2.1 Exploring the bivariate relationship between Y and X

Questions

1. Import the "FCM data dryad.csv" data from the Dryad repository as the data.table `fcm`
2. How are different words in the column labels demarcated? Is this good practice?

Here we want to fit a model of `FCM.ng.g.dry` as a function of `Raw.Squirrel.Density`. The authors used prior knowledge to expect a positive relationship between these two variables. Use qplot to generate a scatterplot of FCM against $Density$

Questions

3. Is there a trend? If so, does the trend look linear or non-linear?
4. Does the residual variation (the deviation from the trend on the Y axis) look homogenous along the X-axis?
5. Are there any obvious outliers?

5.2.2 Fitting the linear model

We will fit a linear model to the data using the `lm` function, which is very general and will be our workhorse throughout the class. The minimal input to the function is a model formula and the name of the data.frame (remember, a data.table is a data.frame). A formula is of the form `Y ~ X`. All of the output we assign to the object `fit`.

Let's fit the linear model to the data using density as the predictor

```
fit <- lm(FCM.ng.g.dry ~ Raw.Squirrel.Density, data=fcm)
```

R will look for the specified Y and X variables in the column names of `fcm`. If these are not found, R will return an error, for example

```
fit <- lm(FCM_ng_g_dry ~ Raw_Squirrel_Density, data=fcm)
```

will return the error "Error in eval(predvars, data, env) : object 'FCM_ng_g_dry' not found". This means your spelling and capitalization have to be exact!

5.2.3 Getting to know the linear model: the summary function

The `lm` function returns an `lm` object, which we've assigned to the name `fit`. `fit` contains lots of information about our fit of the linear model to the data. Most of the information that we want for most purposes can be retrieved with the `summary` function, which is a general-purpose R command the works with many R objects.

```
summary(fit)
```

```
##
## Call:
## lm(formula = FCM.ng.g.dry ~ Raw.Squirrel.Density, data = fcm)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2107.5 -1108.3  -434.9   511.8  8186.8
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)              736.0      331.9   2.217 0.028078 *
## Raw.Squirrel.Density     671.1      178.9   3.752 0.000248 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1732 on 154 degrees of freedom
##   (7 observations deleted due to missingness)
## Multiple R-squared:  0.08374,    Adjusted R-squared:  0.07779
## F-statistic: 14.07 on 1 and 154 DF,  p-value: 0.0002484
```

What is here:

Call. This is the model that was fit

Residuals. This is a summary of the distribution of the residuals. From this one can get a sense of the distribution (for inference, the model assumes a normal distribution with mean zero). More useful ways to examine this distribution will be introduced later in this chapter.

Coefficients table. This contains the linear model coefficients and their standard error and associated t and p values.

1. The column of values under "Estimate" are the coefficients of the fitted model (equation (5.4)). Here, 735.9604344 is the intercept (b_0) and 671.1379749 is the effect of *Density* (b_1).
2. The column of values under "Std. Error" are the standard errors of the coefficients.
3. The column of values under "t value" are the *t-statistics* for each coefficient. A t-value is a **signal to noise ratio**. The coefficient b_1 is the "signal" and the SE is the noise. Get used to thinking about this ratio. Any t less than 2 is indicative of too much noise to say much about the signal. A t between 2 and 3 means the noise is large enough to suggest an effect. A t greater than 3 is pretty good evidence of an effect.
4. The column of values under "Pr(>|t|)" is the p-value, which is the exact probability associated with a particular t. What is the p-value a test of? The p-value tests the hypothesis "how probable are the data if the coefficient is zero?". Formally $P = \text{freq}(\text{t}' \geq \text{t}|\text{H}_\text{o})$, where t' is the hypothetical t-value, t is the observed t-value, and H_o is the null hypothesis. We will return to p-values in Chapter xxx.

Signif. codes. I am surprised that base R returns this. These are useless because the concept of "levels of significance" is muddled, as will be discussed in Chapter xxx.

Beneath the Signif. codes are some model statistics which are useful

Residual standard error This is $\sqrt{\sum e_i^2/(n-2)}$, where e_i are the residuals in the fitted model. "degrees of freedom" is the number of e_i that are "allowed to vary" after fitting the parameters, so is the total sample size (n) minus the number of parameters fit. The fit model has two fit parameters (b_0 and b_1 so the df is $n-2$. Note that this is the denominator in the residual standard error equation.

Multiple R-squared. This is an important but imperfect summary measure of the whole model that effectively measures how much of the total variance in the response variable "is explained by" the model. Its value lies between zero and 1. **It's a good measure to report in a manuscript**.

F-statistic and p-value. These are statistics for the whole model (not the individual coefficients) and I just don't find these very useful.

Note that the p-value for the coefficient for Raw.Squirrel.Density is very small and we could conclude that the data are not consistant with a model of no slope. But did we need a formal hypothesis test for this? We haven't learned much if we have only learned that the slope is "not likely to be exactly zero". What we want to know is not *if* there is a relationship between FCM and $Density$, which is imperfectly answered with a p-value, but *the sign and magnitude* of the relationship and the uncertainty in this estimate. For this, we don't need the p-value. Instead, we want to interpret the coefficient to its SE directly (for a quick-and-dirty interpretation) or the confidence interval of the effect (for a more formal interpretation). Please read this paragraph again. We will come back to it over and over.

5.2.4 display: An alternative to summary

Much of what we want to know about a model fit is returned by the `display` function from the `arm` package.

```
display(fit)
```

```
## lm(formula = FCM.ng.g.dry ~ Raw.Squirrel.Density, data = fcm)
##                      coef.est coef.se
## (Intercept)          735.96   331.94
## Raw.Squirrel.Density 671.14   178.90
## ---
## n = 156, k = 2
## residual sd = 1732.02, R-Squared = 0.08
```

The `display` function does not give a t-value or a p-value of the coefficients because the authors of the arm package do not think p-values are very valuable. We don't need a t because one can mentally compute the approximate ratio of the coefficient to its SE and get a sense of the signal to noise, and that's all the authors of the display function think we need.

5.2.5 Confidence intervals

Confidence intervals for the coefficients of the model are obtained by

```
confint(fit)
```

```
##                          2.5 %   97.5 %
## (Intercept)           80.21785 1391.703
## Raw.Squirrel.Density 317.73057 1024.545
```

`confint` returns by default the 95% confidence interval (CI) of all parameters. The most useful way of thinking about the meaning of a CI is

A confidence interval contains the range of parameter values that are consistent with the data, in the sense that a t-test would not reject the null hypothesis of a difference between the estimate and any value within the interval

A more textbook way of defining a CI is: A 95% CI of a parameter has a 95% probability of including the true value of the parameter. It does not mean that there is a 95% probability that the true value lies in the interval. This is a subtle but important difference. Here is a way of thinking about the proper meaning of the textbook definition: we don't know the true value of β_1 but we can 1) repeat the experiment or sampling, 2) re-estimate β_1, and 3) re-compute a 95% CI. If we do 1-3 many times, 95% of the CIs will include β_1 within the interval.

Confidence intervals are often interpreted like p-values. That is, the researcher looks to see if the CI overlaps with zero and if it does, concludes there is "no effect". First, this conclusion is not correct – **the inability to find sufficient evidence for an effect does not mean there is no effect, it simply means there is insufficient evidence to conclude there is an effect**!

Second, what we want to use the CI for is to guide us about how big or small the effect might reasonably be, given the data. Again, A CI is a measure of parameter values that are "consistent" with the data. If our biological interpretations at the small-end and at the big-end of the interval's range radically differ, then we don't have enough *precision* in our analysis to reach an unambiguous conclusion. Remember this.

5.2.6 How good is our model?

How well does variation in *Density* "explain" variation in FCM? The answer to this is in the R^2 value, which is given in `display(fit)` and in `summary(fit)` and accessed directly with

```
summary(fit)$r.squared
```

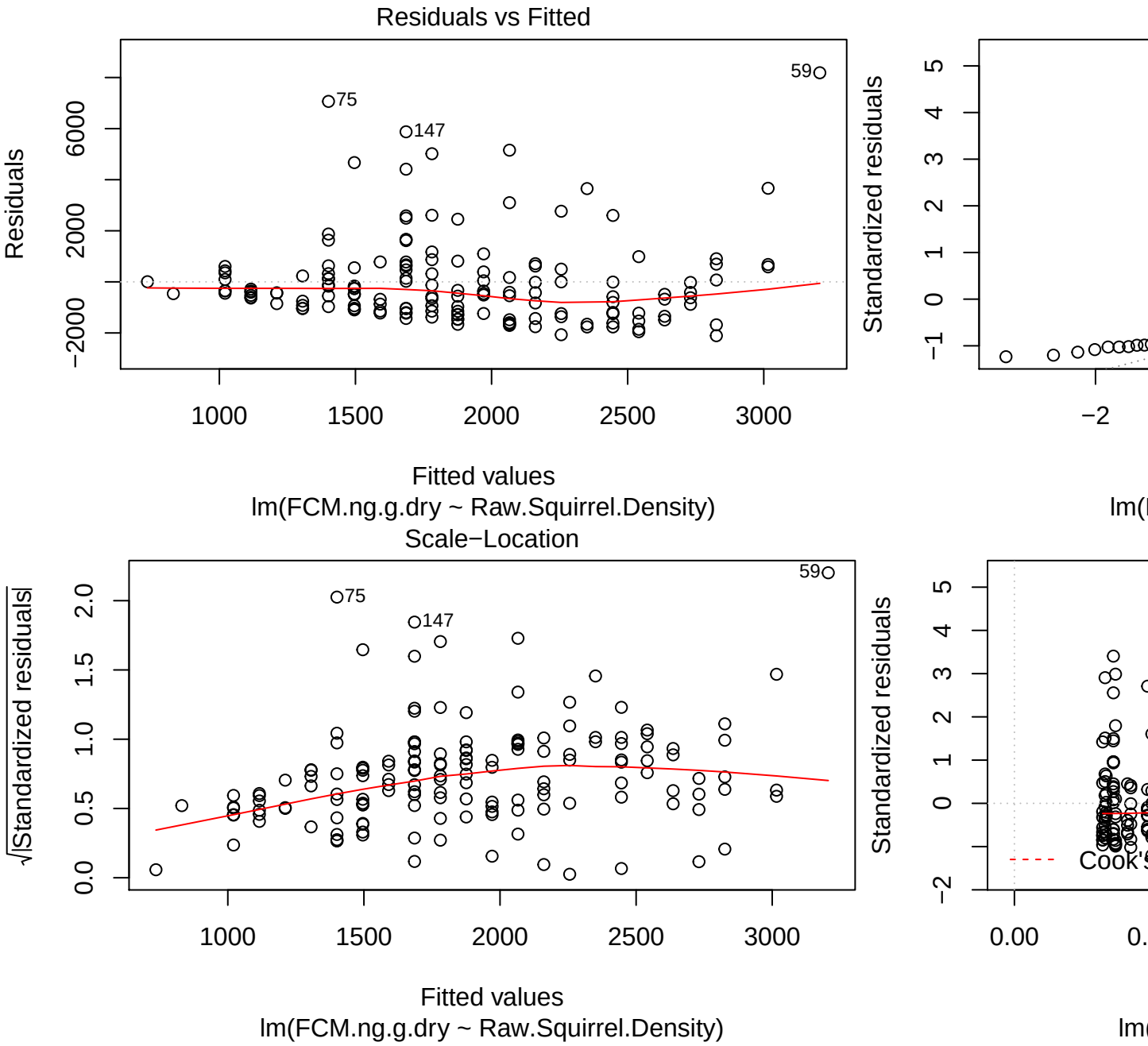
```
## [1] 0.08373756
```

R^2 is the fraction of the total variance of Y explained by the model, or more specifically, the linear predictor. It will vary from zero (the model explains nothing) to one (the model explains everything). If $R^2 = 0$ the response is completely unpredictable by the predictors. We can think of the values of the response as white noise or all error. This doesn't mean that the values are "not caused" or "random" or not predicted by some other variable. It only means the values are random with respect to the X variable(s) in the model. If $R^2 = 1$ we can *exactly* predict the response from the X variables in the model. So the bigger the R^2, the better the model in the sense that the response is more predicatable. **Super importantly**, "explains" is in a probabilistic and not causal sense. We will explore this concept much more in future worksheets.

5.2.7 Model checking

`plot` is a very useful base R function for "model checking" or "model diagnostics" to see if our model fit is acceptable.

```
plot(fit)
```



Compare the four diagnostic plots using the guidelines from here http://data.library.virginia.edu/diagnostic-plots/

Questions

6. Look at the plots you just made. What is a residual? What is a fitted value?

5.2.8 exploring a lm object

`fit` contains much information but simply typing `fit` into the console gives us only the model and the coefficients. `names` is a super important R function. It gives us the names of all the parts of some R object. `fit` is an lm object. `names(fit)` gives us all the parts contained in an lm object.

```
names(fit)
```

```
##  [1] "coefficients"  "residuals"     "effects"       "rank"
##  [5] "fitted.values" "assign"        "qr"            "df.residual"
##  [9] "na.action"     "xlevels"       "call"          "terms"
## [13] "model"
```

You can see any of these parts using the dollar sign

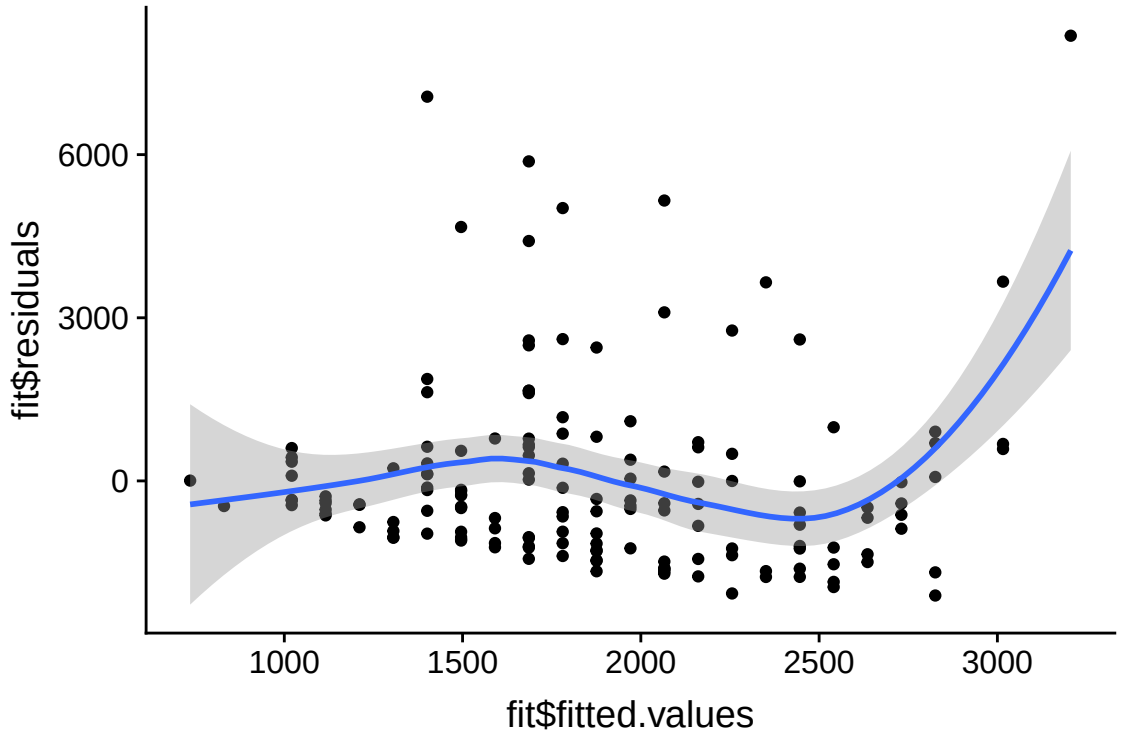
Questions

7. What does `fit$residuals` return? Answer using equation (5.4)
8. What does `fit$fitted.values` return? Answer using equation @ref(eq:fcmi

You can use qplot to make a plot similar to the first plot of `plot(fit)`

```
qplot(fit$fitted.values, fit$residuals, geom=c('point', 'smooth'))
```

```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```



5.3 Problems

1. Using the chick data from Chapter 3. Compare the effects of nest_temperature_above_ambient on day13_mass by fitting two separate linear models 1) one using only the control group and one using the treated group. The grouping variable is playback_treatment. These models were plotted in Chapter 3 so `lm` will return the linear model behind these plots.

Report the results using the two effect estimates and a 95% confidence interval (we will learn in a later chapter a more sophisticated way of comparing the effects between the groups)

file name: "allDatasetsMarietteBuchanan2016.xls"

source: https://datadryad.org//handle/10255/dryad.122315

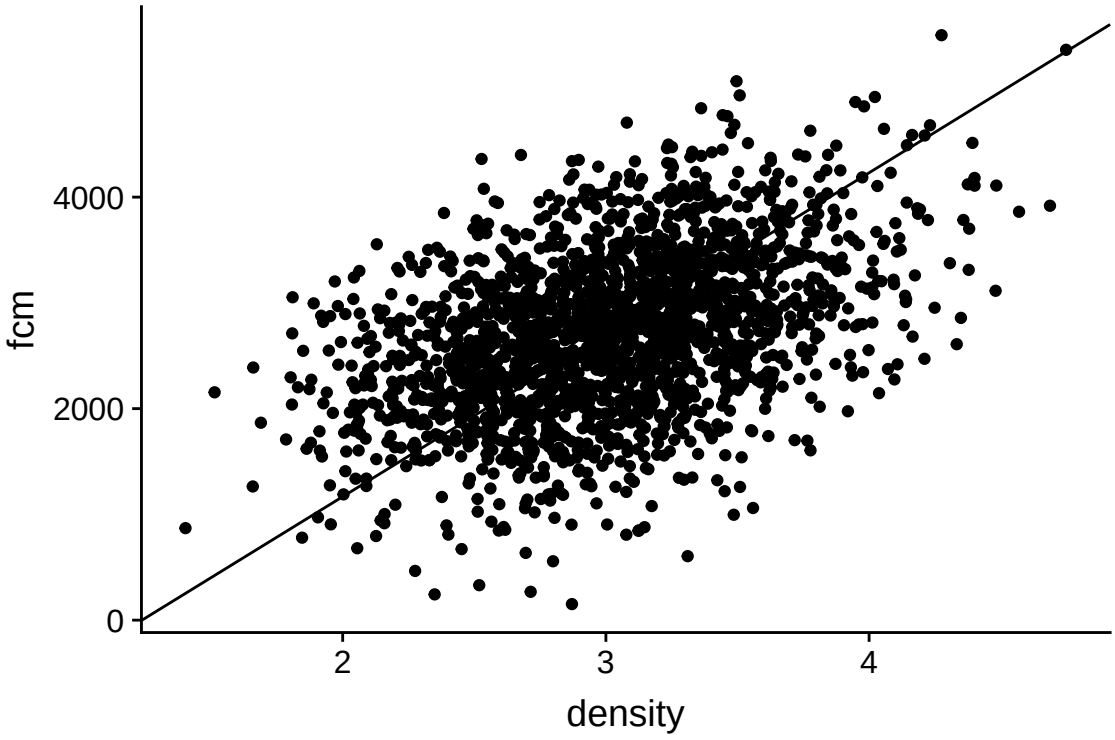
2. (Grad students only) – find a dataset using Dryad that has data that can be fit by a simple linear model with a single continuous X (its okay if the authors fit the data with a more complex model). Fit the data and report the results with a plot and text.

Chapter 6

Least Squares Estimation

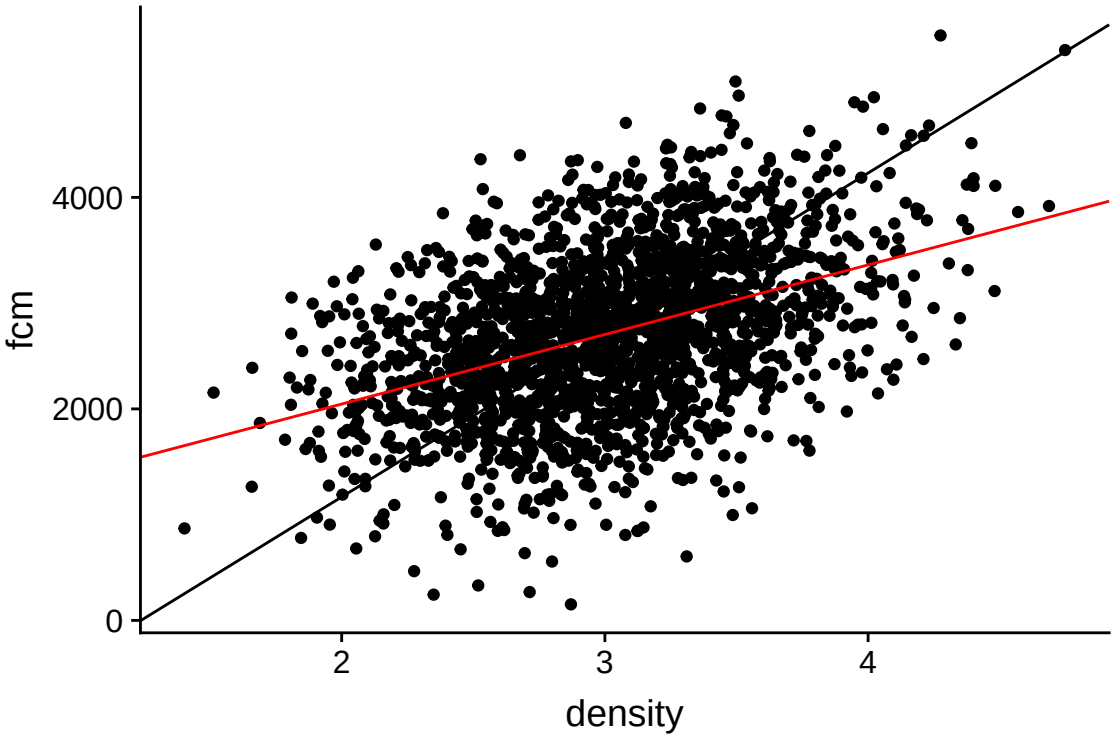
The linear models in the last chapter and for much of this book are fit to data using a method called "ordinary least squares" (OLS). This chapter explores the meaning of OLS and related statistics, including R^2, as well as some alternative methods for bivariate regression.

6.1 OLS regression



The fake data illustrated in the scatterplot above (Figure **??**) were modeled to look something like the squirrel fecal cortisol metabolite data in the previous chapter. If a typical student is asked to draw a regression line through the scatter, they typically draw a line similar to that in Figure **??**. This line is not the OLS regression line but the major axis of an elipse that encloses the scatter of points–that students invariably draw this line suggests that the brain interprets the major axis of an elliptical scatter of points as a trend (This major axis line is an alternative method for estimating a slope and is known as standard major-axis regression. More

about this at the end of this chapter.)



The OLS regression line is the red line in Figure **??** – the standard major axis line is left for comparison). The OLS regression line

1. passes through the bivariate mean $(\bar{x}, \bar{y})$ of the scatter, and
2. minimizes the sum of the squared deviations from each point to it's modeled value $\sum (y_i - \hat{y}_i)^2$

There are an infinite number of lines that pass through the bivariate mean (think of anchoring a line at the bivariate mean and spinning it). The OLS line is the line that minimizes the squared (vertical) deviations ("least squares").

For a bivariate regression, the slope (coefficient b_1 of X) of the OLS model fit is computed by

$$b_1 = \frac{\text{COV}(X, Y)}{\text{VAR}(X)} \tag{6.1}$$

This equation is worth memorizing. We will generalize this into a more flexible equation in a few chapters.

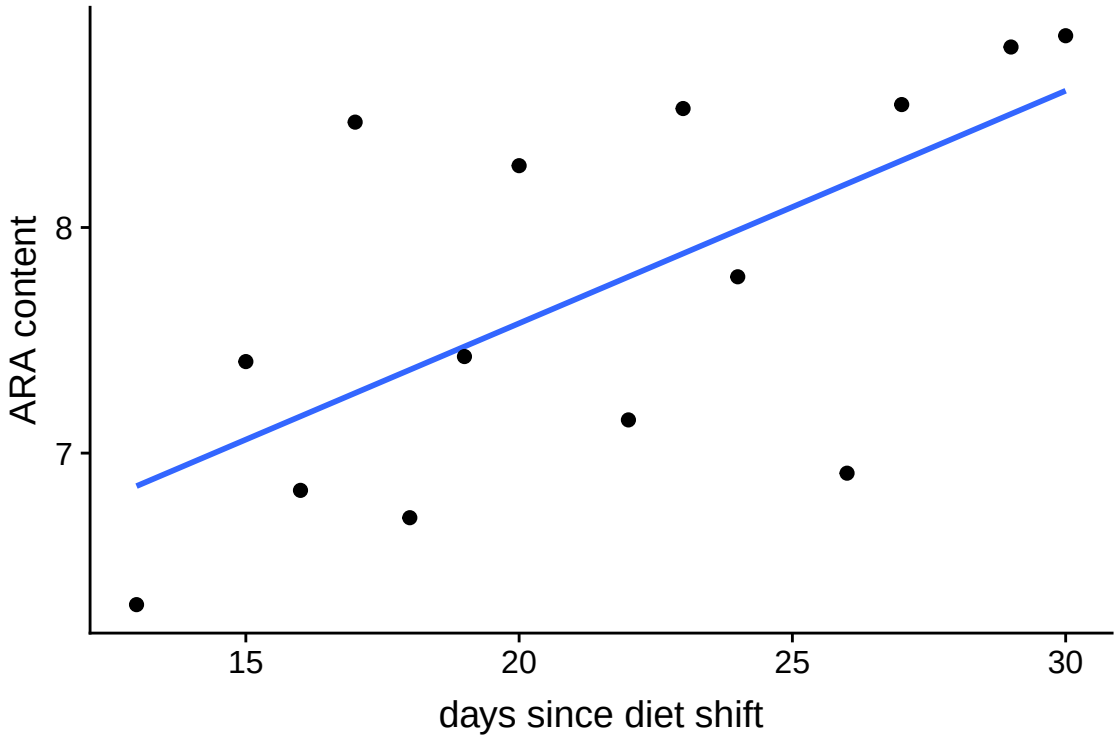
6.2 How well does the model fit the data? R^2 and "variance explained"

Let's switch to real data.

1. Source: Dryad Digital Repository. https://doi.org/10.5061/dryad.056r5
2. File: "Diet-shift data.xls"

Fish require arachidonic acid (ARA) and other highyly unsaturated fatty acids in their diet and embryo and yolk-stage larvae obtain these from yolk. Fuiman and Faulk (xxx) designed an experiment to investigate if red drum (*Sciaenops ocellatus*) mothers provision the yolk with ARA from recent dietary intake or from

stored sources in somatic tissues. The data below are from experiment 8. The x-axis is the days since a diet shift to more and less ARA ($days$) and the y-axis is the ARA content of the eggs (ARA).



The statistic R^2 is a measure of the fit of a model to data. The R^2 for the fit of the egg data is 0.42. R^2 is the fraction of two variances $\frac{\text{VAR}(Model)}{\text{VAR}(Y)}$, or, the fraction of the variance of Y "explained by the model." The value of R^2 ranges from zero (the fit cannot be any worse) to one (the fit is "pefect").

To understand R^2, and its computation, a bit more, let's look at three kinds of deviations.

Figure 6.1A shows the deviations from the measured values to the mean value (dashed line). These are the deviations in the numerator of the equation to compute the variance of $ARA_E GG_M G$. Figure 6.1B shows the deviations of the measured values from the modeled values. The sum of these deviations squared is what is minimized by the OLS fit. The bigger these deviations are, the worse the model fit. Figure 6.1C shows the deviations of the modeled values to the mean value. The bigger these deviations are, the better the model fit.

The sums of the squares of these deviations (or "sums of squares") have names:

$$\text{SS(total)} = \sum (y_i - \bar{y})^2 \tag{6.2}$$

$$\text{SS(error)} = \sum (y_i - \hat{y}_i)^2 \tag{6.3}$$

$$\text{SS(model)} = \sum (\hat{y}_i - \bar{y})^2 \tag{6.4}$$

Again, SS(total) is the numerator of the equation for the sample variance. It is called "s-s-total" because SS(total) = SS(model) + SS(error). That is, the total sums of squares can be **decomposed** into two **components**: the modeled sums of squares and the error sums of squares. Given these components, it's easy to understand R^2

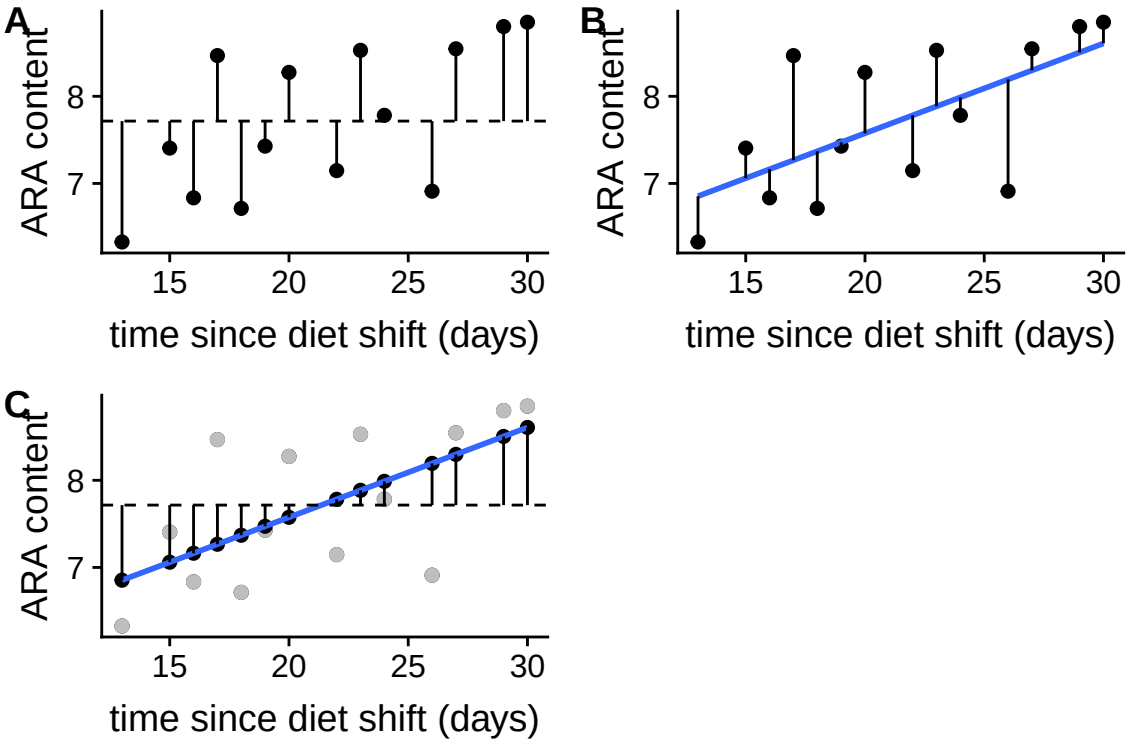


Figure 6.1: Three kinds of deviations from a fit model. A. Deviations of the measured values from the mean. These are in the numerator of the equation of the sample variance. The dashed line is the mean ARA content. B. Deviations of the measured values from the modeled values. The sum of these deviations squared is what is minimized in an OLS fit. C. Deviations of the modeled values from the mean ARA content. The measured values are in gray, the modeled values in black

$$R^2 = \frac{SS(model)}{SS(total)} \tag{6.5}$$

R^2 is the fraction of the total sums of squares that is due to (or "explained by") the model sums of squares. Above I said that R^2 is the fraction of *variance* explained by the model. Equation xxx is a ratio of variance, but the $(n-1)^{-1}$ in both the numerator and the denominator cancel out. Finally, many sources give the equation for R^2 as

$$R^2 = 1 - \frac{SS(error)}{SS(total)} \tag{6.6}$$

which is an obvious alternative given the decomposition. I prefer the former equation because it emphasizes the model fit instead of model ill-fit.

Chapter 7

A linear model with a single, categorical X

7.1 A linear model with a single, categorical X is the engine behind a single factor (one-way) ANOVA and a t-test is a special case of this model.

To introduce modeling with a single, categorical X variable, I'll use the vole data from

1. Source: Dryad Digital Repository. https://doi.org/10.5061/dryad.31cc4/1
2. File: "RSBL-2013-0432 vole data.xlsx"
3. Sheet: "COLD VOLES LIFESPAN"

Normal cellular metabolism creates reactive oxygen species (ROS) that can disrupt cell function and potentially cause cell damage. Anti-oxidants are molecules that bind ROS, inhibiting their ability to disrupt cell activity. A working hypothesis for many years is that supplemental anti-oxidants should improve cell function and, scaling up, whole-animal function (such as lifespan). The vole data explores this with supplemental Vitamins C and E, which are anti-oxidants, in the diet of the short-tailed field vole (*Microtus agrestis*).

The goal of the study is to measure the effect of anti-oxidants on lifespan. The researchers randomly assigned the voles to one of thre treatment levels: "control", "vitamin E", and "vitamin C". The variable *treatment*, is a single, categorical X variable. Categorical variables are often called **factors** and the treatment levels are often called **factor levels**. There are no units to a categorical X variable (even though a certain amount of each anti-oxidant was supplemented). The response (Y) is $lifespan$ measured in days.

The linear model with a categorical X variable with three levels is not immediately obvious, and so I don't present the model until after showing the table of model coefficients

7.1.1 Table of model coefficients

Here is the table of coefficients from the linear model fit

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	503.4	27.4	18.4	0.000
treatmentvitamin_E	-89.9	52.5	-1.7	0.090
treatmentvitamin_C	-115.1	54.5	-2.1	0.037

The table has estimates for three parameters. The first estimate (the intercept) is the mean response in the reference level. Here the reference level is the "control" group. The additional estimates are the differences

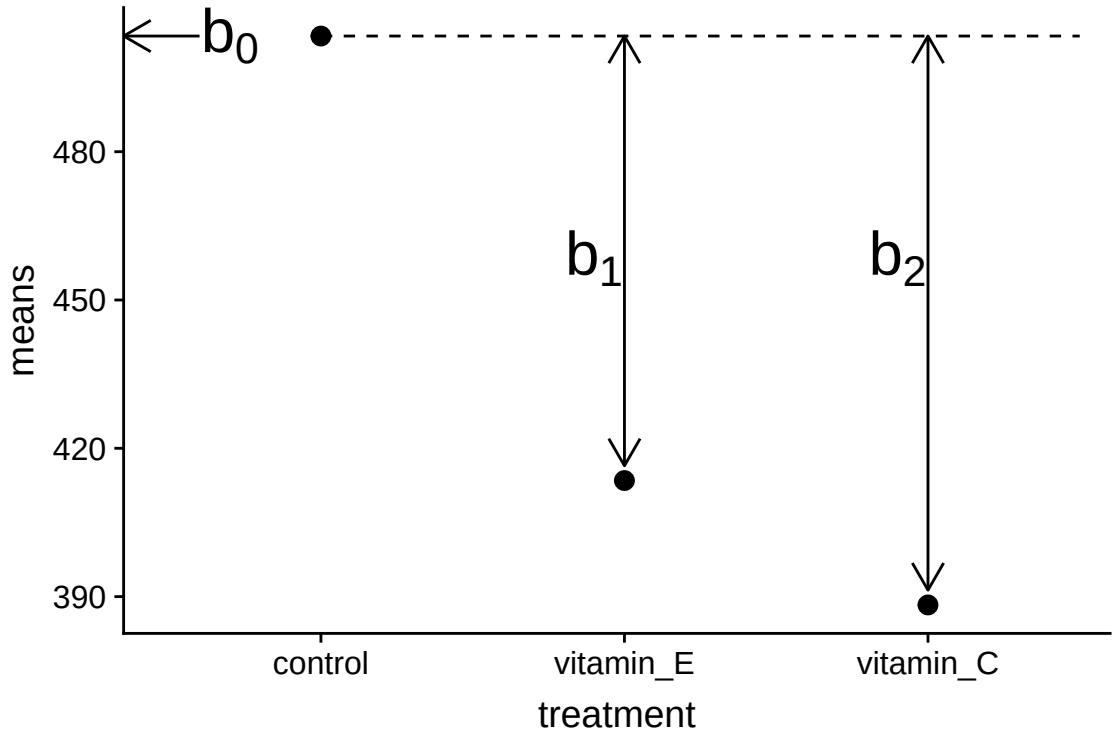


Figure 7.1: What the coefficients of a linear model with a single categorical X mean. The means of the three treatment levels for the vole data are shown with the filled circles. The length of the double-headed arrows are differences in means. The intercept (b_0) is the mean of the reference treatment level. The coefficients (b_1 and b_2) are the differences between the treatment level's mean and the reference mean. As with a linear model with a continuous X, the coefficients are effects.

in the mean between each of the other treatment levels and the reference level. These are the "effects" in the model. So typically with categorical X, when we speak of an *effect* we mean a difference in means. These estimates and their meaning are illustrated in Figure 7.1.

(note. The default in R is to set the level that is first alphabetically as the reference level. In the vole data, "control" comes before "vitamin_E" and "vitamin_C" alphabetically, and so by default, it is the reference level. This makes sense for these data – we want to compare the lifespan of the vitamins E and C groups to that of the control group. The reference level can be changed of course.)

7.1.2 The linear model

We can see an immediate difference between the coefficient table for a linear model fit to a single, categorical X and that for a single, continuous X. For the latter, there is a single coefficient for X. For the former, there is a coefficient for each level of the categorical X *except* the "reference" level.

The linear model for a single, continuous X with three factor levels is

$$lifespan = \beta_0 + \beta_1 vitamin\_E + \beta_2 vitamin\_C + \varepsilon \tag{7.1}$$

and the estimates in the coefficient table are the coefficients of the fit model

$$lifespan_i = b_0 + b_1 vitamin\_E + b_2 vitamin\_C + e_i \tag{7.2}$$

Remember, b_0 is the mean of the control group, b_1 is the difference in means between the vitamin E and control groups, and b_2 is the difference in means between the vitamin C and control groups (Figure 7.1).

In this model, *vitamin_E* and *vitamin_C* are **dummy variables** that contain a one, if the data is from that treatment level, and zero otherwise. This is called dummy coding or treatment coding. The `lm` function creates these dummy variables under the table, in something called the **model matrix**, which we'll cover in the next chapter. You won't see these columns in your data. But if you did, it would look something like this

lifespan	treatment	vitamin_E	vitamin_C
621	control	0	0
865	control	0	0
583	vitamin_E	1	0
561	vitamin_E	1	0
315	vitamin_C	0	1
157	vitamin_C	0	1

There are alternative coding methods. Dummy coding is the default in R. Note that the method of coding can make a difference in an ANOVA table, and many published papers using R have published incorrect interpretations of ANOVA table outputs. This is both getting ahead of ourselves and somewhat moot, because I don't advocate publishing ANOVA tables.

7.1.3 Reporting results

What should be reported for the analyis of effects of anti-oxidant supplements on vole lifespan? Best practice includes reporting the raw data with a summary distribution and treatment effects with CIs. "Raw data" means the individual lifespans as a function of treatment level.

7.1.3.1 Harrel Plot of the data

The raw data, the distributions within treatment level, and the effects (difference in means) of treatment can be combined into a single plot that I call a Harrell plot (Figure **??**). Notice that the x-axis and y axes are flipped so that $lifespan$ is on the x-axis. It is still the "response" or "Y" variable! The Harrell plot contains two parts

1. The bottom contains a **strip chart** (often called a "dot plot") of the raw response measures grouped by factor level. Superimposed over the strip chart is a **box plot** summarizing the distribution of each factor level. The line in the center of a box is the median $lifespan$ for that group, the left and right edges of the box are the 25% and 75% quantiles of $lifespan$ for that grop, and the lines extending to the left and right of the box are the "whiskers", which are the smallest and largest value within $1.5IQR$ (inter-quartile range, which is the interval bounded by box).
2. The top is a **forest plot** of the effects and the 95% CI of the effects. For categorical X, the effects could be model coefficients or treatment **contrasts**, which are differences in means between treatment levels. Model coefficients are a subset of possible treatment contrasts.

The Harrell plot above shows the effects as model coefficients, which (again!) are differences between the mean of the response in a specific treatment level and the mean of the response in a reference level. Here the reference level is the control group.

7.1.3.2 In-text reporting

"The mean lifespan of cold-reared voles supplmented with vitamin E was -89.9 days shorter than the mean lifespan for the control group (95% CI: -194.1, 14.3). The mean lifespan of cold-reared voles supplmented with vitamin C was -115.1 days shorter than the mean lifespan for the control group (95% CI: -223.2, -6.9).

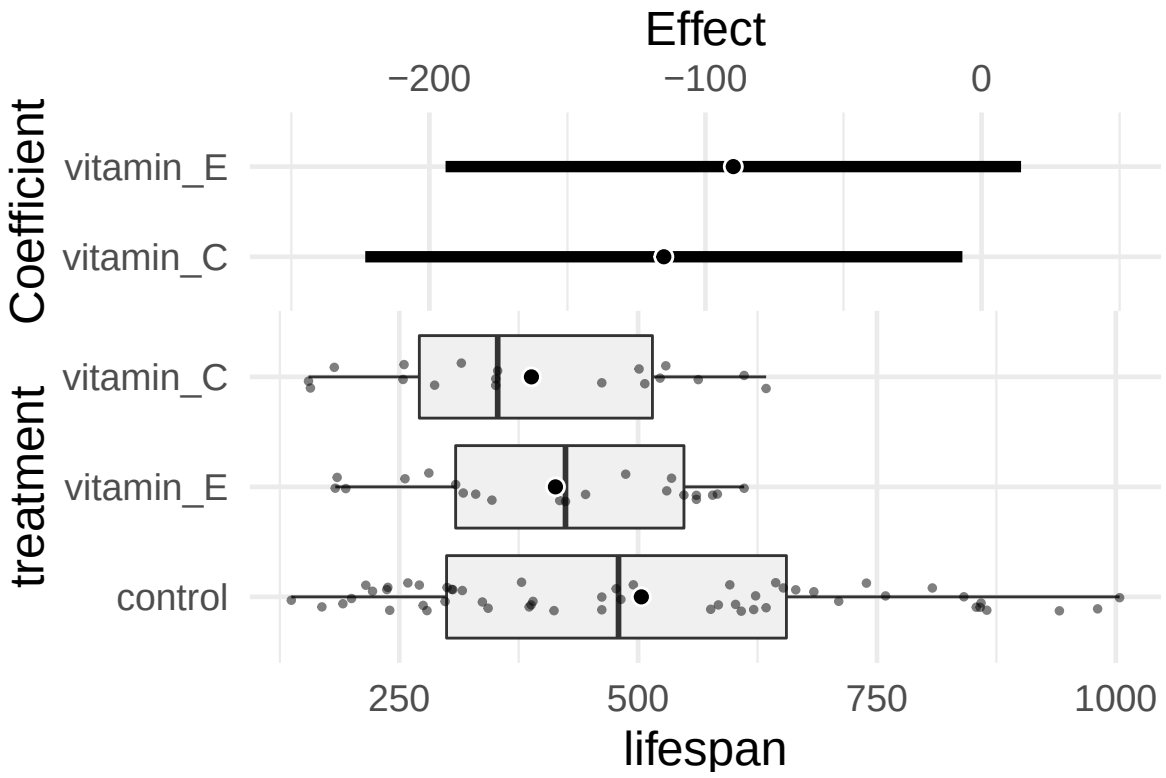


Figure 7.2: HarrellPlot of the raw data, distribution, and effects of the vole lifespan data.

7.1.3.3 Correct interpretation of the Confidence Interval is key

Remember, that the CI contains the range of parameter values that are consistent with the data (in the sense that a t-test wouldn't reject the hypothesis test). This means that a true value at the low end or the high end of the CI is consistent with the data. Your technical report/manuscript should discuss the consequences of this. For example, A small, increase in lifespan is consistant with the Vitamin E but not Vitamin C supplementation, if we use the 95% CI as a pretty good range for inferring "consistent with". Both a 223 day and a 7 day decrease in lifespan are consistant with the Vitamin C effect. 223 days seems like a huge effect, especially for a short lived vole. 7 days is certainly a much smaller effect, but this doesn't mean that it doesn't have important ecological, behavioral, or fitness consequences.

7.2 Working in R

7.2.1 Exploring the relationship between Y and X

Questions

1. Import the vole data from the Dryad repository as the data.table `vole_wide`. Replace the spaces in the column labels with the underscore "_".

The data are in "wide" format. A pretty good script for for converting these to long format is

```
vole_long <- melt(vole_wide, measure.vars=c("control", "vitamin_E", "vitamin_C"), variable.name="treatm
vole <- na.omit(vole_long)
# melt is from reshape2 package and is a workhorse in R
# the resulting data frame has 3 stacked copies of the same rows
```

```
# na.omit removes the superfluous two extra sets of rows created by melt
# the more compact way to do this is combine the steps:
# vole <- na.omit(melt(vole_wide, measure.vars=c("control", "vitamin_E", "vitamin_C"))
# but I suggest two steps so you can see what melt does. This isn't the best example of using melt.
```

Use the ggpubr package to create a box plot, grouped by treatment, with superimposed strip chart ("dots")

Questions

2. Do the response as a function of group look fairly normally distributed or are there red flags such as skewness, outliers, bimodal clusters, etc.
3. Is the direction of the effect consistent with the expected direction?

7.2.2 Fitting the model

As with a single, continuous X, we fit the model using the `lm` function and with the model formula of the form `y ~ x`. Note that the R formula can use the single categorical variable *treatment*. The code underneath lm will note that *treatment* is a factor with three levels and will automatically create the two dummy variables noted above in the linear model.

```
fit <- lm(lifespan ~ treatment, data=vole)
```

All of the same scripts to access the information in `fit` that we used with the continuous X analysis are the same. For example, the base R `summary` function gives the same information as in the continuous X example.

Questions

4. Review different output in the `summary` function and list which are useful and which are not so useful and why.

Other useful functions on the lm object ("fit") are `coefficients(summary())`, `coef()` or `coefficients()` and `confint`. Assigning the output of these functions to an R object allows you to increase reproducibility. For example, if I assign the coefficients to `b`

```
b <- coef(fit)
```

then I can report these in R markdown text by embedding r code directly in the text. For example, if I embed "r round(b["treatmentvitamin_E"], 1)" between a pair of single accent characters, then r markdown inserts -89.9 into the rendered text.

7.2.3 An introduction to contrasts

We often want to compare more than just the non-reference levels to the reference level. For example, we might want to compare the effects of the vitamin E supplementation to vitamin C supplementation. Or, we might want to combine (or "pool") vitamin C and vitamin E levels effects into a single "anti-oxidant" level and compare to the control. These comparisons of means are called linear **contrasts**. The emmeans package is a good package for obtaining contrasts for both simple linear models computed with `lm` and for more complicated statistical models. If you haven't already, download the emmeans package.

```
fit.em <- emmeans(fit, spec="treatment")
fit.em
```

```
##  treatment   emmean       SE df lower.CL upper.CL
##  control   503.3929 27.40978 93 448.9625 557.8233
##  vitamin_E 413.4762 44.75999 93 324.5917 502.3607
##  vitamin_C 388.3158 47.05685 93 294.8702 481.7614
```

```
##
## Confidence level used: 0.95
```

The `emmeans()` function returns various estimated means, depending on what is specified with the `spec=` parameter. Here the grouping variable "treatment" is specified, so the means returned are estimates of E($lifespan|treatment$), the modeled means for each level of treatment. For this simple analysis, the modeled means are simply the group means. Note that the default value returned is a table with the standard error and 95% confidence limits of the estimates.

Let's use the emmeans object to get the contrasts for all combinations of treatment levels.

```
summary(contrast(fit.em, method="revpairwise", adjust="none"), infer=c(TRUE, TRUE))
```

```
##  contrast                estimate       SE df  lower.CL   upper.CL t.ratio
##  vitamin_E - control    -89.91667 52.48574 93 -194.1429  14.309609  -1.713
##  vitamin_C - control   -115.07707 54.45772 93 -223.2193  -6.934834  -2.113
##  vitamin_C - vitamin_E  -25.16040 64.94462 93 -154.1275 103.806738  -0.387
##  p.value
##   0.0900
##   0.0373
##   0.6993
##
## Confidence level used: 0.95
```

I've sent to parameters to the contrast function and one to the summary function

1. method="revpairwise". `contrast` can create different combinations of differences between means. Here I've specified all pairwise differences (the "rev" reverses the order of the subtraction). Notice that the statistics (estimate, SE, etc) are equal to the same statistics for b_1 and b_2 of the linear model. I said earlier that these coefficients are contrasts!
2. adjust="none". In classical frequentist hypothesis testing, the p-value of a contrast in what are called "post-hoc tests" is adjusted to reflect "multiple testing" (more than one p-value is being computed). This adjustment is almost standard in biology, but the practice is hugely controversial. The concept of multiple testing is important, and we will return to this in a future chapter, but here I have chosen to show the unadjusted p-value. The reason is that I want the unadjusted confidence interval and the adjustment would adjust these as well. If deleted `adjust="none"` from the script, the contrast function would default to the **Tukey HSD** (Honestly Significant Difference) test. There are literally dozens and dozens of post-hoc tests, which largely reflects the misplaced emphasis on "better" p-values rather than parameter estimates and their uncertainty.
3. infer=c(TRUE, TRUE). This parameter controls what kind of inference to put in the table. The first value specifies the inclusion of the CI (emmeans uses "CL" for confidence limit), the second value specifies the inclusion of t and p-values.

7.2.4 Harrell plot

7.2.4.1 Installing the harrellplot package

The harrellplot package is available on github but not a cran repository and, therefore, takes a little more work to install. To install a package from a github repository, 1. load library(devtools) – this may need to be installed first using the R Studio Tools > Install Packages… tool 2. install harrellplot from github. In the console, type

```
install_github("middleprofessor/harrellplot")
```

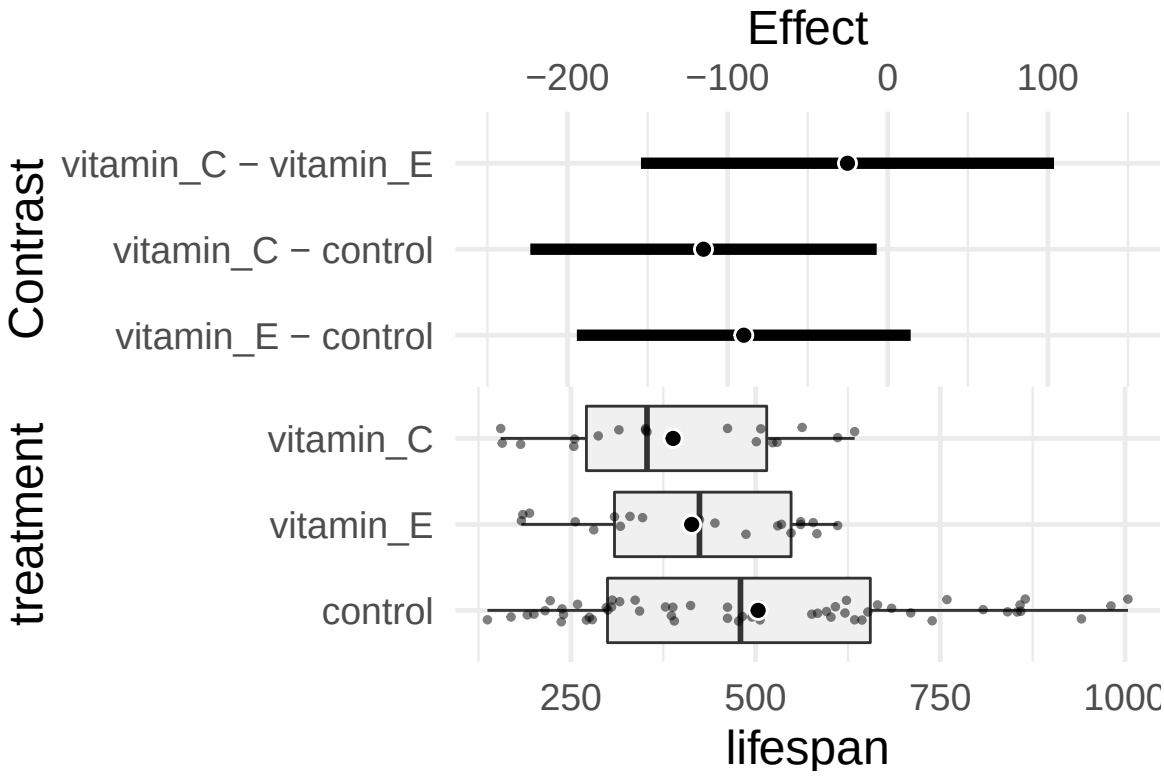
3. load the harrellplot package
4. harrellplot requires other packages including broom, Hmisc, car, lme4, and lmerTest. If you haven't installed these do. load these with the library() function at the start of your notebook.

7.2.4.2 Using harrellplot to make a nice, publishable plot of treatment effects

In the console type '?harrellplot to see the many parameters. Unlike ggplot2, variable names need to be specified with quotes in the harrellplot function. The harrellplot function is a list with several elements.

Here is the default plot

```
vole.harrellplot <- harrellplot(x="treatment", y="lifespan", data=vole)
vole.harrellplot$gg # gg is the plot object
```



Chapter 8

P-values

8.1 p-values

contrast	estimate	SE	df	lower.CL	upper.CL	t.ratio	p.value
vitamin_E - control	-89.9	52	93	-194.1	14.3	-1.7	0.090
vitamin_C - control	-115.1	54	93	-223.2	-6.9	-2.1	0.037
vitamin_C - vitamin_E	-25.2	65	93	-154.1	103.8	-0.4	0.699

Let's use the vole data to introduce the p-value. The table above gives a SE, t and p-value for each pairwise contrast among the three treatment levels. A typical report (one with several misconceptions) might read

"We found a significant effect of Vitamin C (t = -2.1, $p = 0.037$) on lifespan, but no effect of vitamin E (t = -1.7, $p = 0.09$) on lifespan."

A p value *is a continuous measure of evidence against the null.* As long as the data approximate the assumptions of the null hypothesis pretty well, a very small p-value, such as 0.002 or 0.0005, is pretty good evidence against the null hypothesis – but does not mean "an effect exists". To show an effect exists, we should have small p-values in multiple replicates *and* we should rigorously probe the hypothesis with different experiments that challenge the hypothesis in different ways. A small p is evidence for a research program to move forward with replication and probing. A big p-value, say 0.22 or 0.76, is pretty weak evidence against the null, but does not mean "there is no effect." If an experiment is well designed, a big p could suggest abandoning any hypotheses that predict biologically consequential effects. Unfortunately, a big p could also reflect a weak experimental design. Between small and big p values, such as 0.009 or 0.011, problems arise. These intermediate p-values beg for replication. A major problem of inference using p values is that there is no sharp boundaries between these three regions. Instead biologists typically use the $p < 0.05$ as a sharp boundary to declare that an effect exists or not.

Okay. so what *is* a p-value? When we do a t-test, we get a p-value. There are several ways to think about this probability. The most compact way is $P(data|null)$, which is literally read as the probability of the data given the null (or "conditional" on the null), but is really short for *the probability of the data, or something more extreme than the data, given that the null hypothesis is true.* The "probability of the data" is kinda vague. More specifically, we mean the probability of some statistic about the data such as the difference in means between group A and group B or the t-value associated with this difference. So, a bit more formally, the probability returned in a t-test is prob($t \geq t_{obs}|H_0$). This is the long run frequency of observing a t-value as big or bigger than the observed t-value (the one you actually got with your data) if the null is true. Let's parse this into "long run frequency of observing a t-value as big or bigger than the observed t-value" and "null is true".

A thought experiment: You open a google sheet and insert 12 standard, normal random deviates (so the true mean is zero and the true variance is one) in Column A, rows 1-12. You arbitrarily assign the first six

values (rows 1-6) to treatment A and the second six values (rows 7-12) to treatment B. You use the space immediately below these data to compute the mean of treatment A, the mean of treatment B, the difference in means (A - B), and a t-value. Unfortunately, google sheets doesn't have a t-value function so you'd have to compute this yourself. Or not, since this is a thought experiment. Now "fill right" or copy and paste these functions into 999 new columns. You now have 1000 t tests. The expected value of the difference in means is zero (why?) but the actual values will form a normal distribution about zero. Most will be close to zero (either in the negative or positive direction) but some will be further from zero. The expected t-value will also be zero (why?) and the distribution of these 1000 t values will look normal but the tails are a little fuller. This row of t values is a null distribution, because in generating the data we used the exact same formula for the values assigned to A and the values assigned to B. Now think of a t-value in your head, say 0.72 (remember that t values will largely range from about -3 to +3 although the theoretical range is $-\infty$ to $+\infty$. What is the probability of observing a t of 0.72 *or bigger* if the null is true? Look at the row of t-values! Count the number of $t \geq 0.72$ and then divide by the total number of t-values in the row (1000) and you have a probability computed as a frequency. But remember the frequentist definition is the long run frequency, or the expected frequency at the limit (when you've generated not 1000 or even 1,000,000 but an infinite number of columns and t-values).

Some asides to the thought experiment: First, why "as big or bigger" and not just the probability of the value itself? The reason is that the probability of finding the exact t is 1/infinity, which doesn't do us much good. So instead we compute the probability of finding t as big, or bigger, than our observed t. Second, the t-test probability described above is a "one-tail probability". Because a difference can be both in the positive direction and the negative direction, we usually want to count all the $t \geq 0.72$ and the $t \leq -0.72$ and then add these two counts to compute the frequency of *as extreme or more extreme* values. This is called a "two-tailed probability" because we find extremes at both tails of the distribution. Third, we don't really count $t \geq 0.72$ but take advantage of the beautiful mathematical properties of the theoretical t distribution, which allows us to compute the frequentist probability (expected long range frequency) given the t-value and the degrees of freedom using the t-distribution.

Now what do I mean with the phrase "null is true"? Most people equate "null is true" with "no difference in means" but the phrase entails much more than this. Effectively, the phrase means that the p-value is based on modeling the real data with a theoretical sample in which all the points were randomly sampled from the same distribution and that the assignment of the individual points to treatment was random. This model means the theoretical sample has three properties: First, random assignment to treatment after sampling from the same distribution means that the expected means are the same, or put differently, the expected difference in means between the assigned groups is zero. Second, random assignment to treatment after sampling from the same distribution *also* means that the expected variances of the two groups are equal. And third, random sampling means that the values of each point are independent – we cannot predict the value of one point knowing information about any other point. **Here is what is super important about this**: if we get a really low p-value, any one of these consequences may be untrue about our data, for example it could be that the true means of the two treatment groups really are different, or it could mean it is the variances that differ between the two groups, or it could mean that the data (or technically, the errors) are not independent of each other. This is why we need certain assumptions to make a p-value meaningful for empirical data. By assuming independent error and homogenous (equal) variances in our two samples, a low p value is evidence of unequal means.

8.2 Creating a null distribution.

Let's repeat: A pretty good definition of a p-value is: the long-run frequency of observing a test-statistic as large or larger than the observed statistic, if the null were true. A more succinct way to state this is

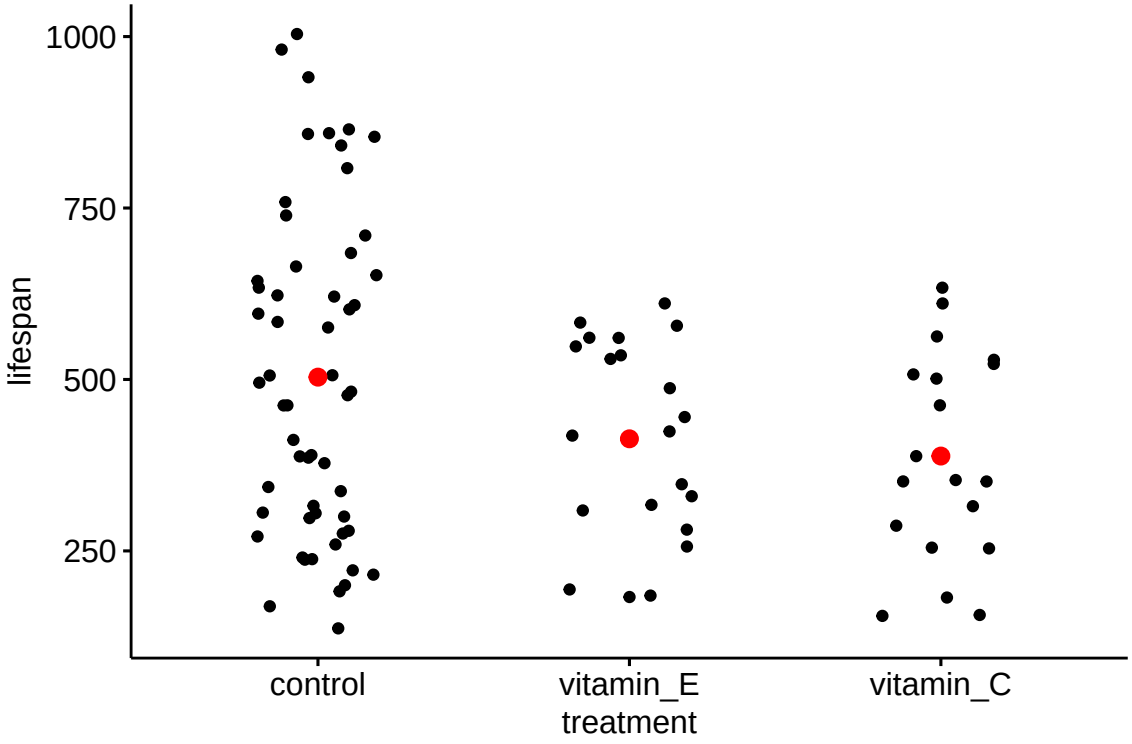
$$p = \text{prob}(t \geq t_o | H_o) \tag{8.1}$$

where t is a hypothetically sampled t-value from a null distribution, t_o is the observed t-value, and H_o is the

null hypothesis. Part of the null hypothesis is the expected value of the parameter estimated is usually (but not always) zero – this can be called the nil null. For example, if there is no vitamin E effect on lifespan, then the expected difference between the means of the control and vitamin E treatment levels is zero. Or,

$$\mathrm{E}(\bar{vitamin}_E - \bar{control}|H_o) = 0.0 \tag{8.2}$$

let's plot the data and look at the group means. Below is a strip chart of the vole data with superimposed treatment level means, using the function `ggstripchart` from the ggpubr package (can you make this?). I'm going to refer to this kind of chart as a "dot plot", which is what most biology researchers call this type of chart.



8.2.1 the Null Distribution

The mean lifespan in the vitamin_E treatment is -89.9 days shorter than the mean lifespan in the control treatment. And, the mean lifespan in the vitamin_E treatment is -115.1 days shorter than the mean lifespan in the control treatment. These are the measured effects, or the **observed differences in means**. How confident are we in these effects? Certainly, if the researchers did the experiment with *two* control treatment groups, they would measure some difference in their means simply because of finite sampling (more specifically, the many, many random effects that contribute to lifespan will differ between the two control groups). So let's reframe the question: are the observed differences unusually large compared to a distribution of differences that would occur if there were no effect? That is, if the "null were true". To answer this, we compare our observed difference to this **null distribution**. This comparison gives the probability (a long-run frequency) of "sampling" a random difference from the null distribution of differences that is as large, or larger, than the observed difference.

What is a null distribution? It is the distribution of a statistic (such as a difference in means, or better, a t-value) if the null were true. Here, I am generating a null distribution that is relevant to the cold vole data. See if you can understand the script before reading the explanation below.

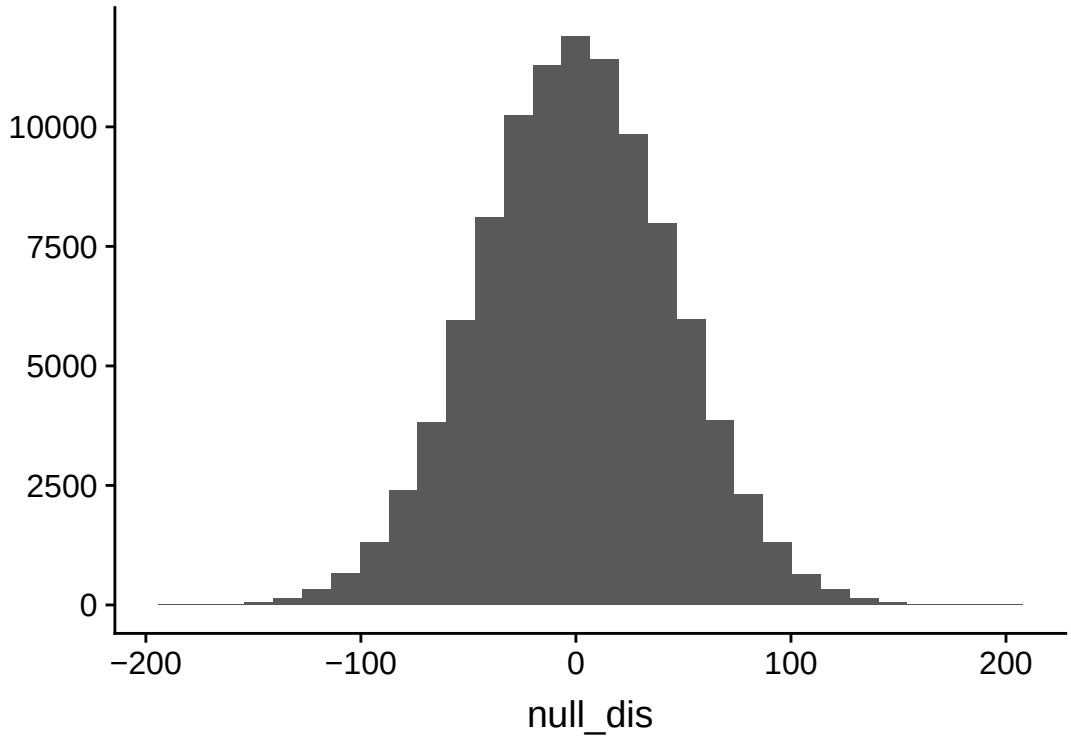


Figure 8.1: Null distribution for an infinitely large data set that looks curiously like the lifespans of the cold-rear voles from the control treatment.

```
seed <- 1
n_iter <- 10^5 # number of iterations
mu <- mean(vole[treatment=='control', lifespan])
sigma <- sd(vole[treatment=='control', lifespan])
n <- nrow((vole[treatment=='control',]))
sample1 <- matrix(rnorm(n*n_iter, mean=mu, sd=sigma), nrow=n) # 100,000 samples (each size n)
sample2 <- matrix(rnorm(n*n_iter, mean=mu, sd=sigma), nrow=n) # 100,000 samples
null_dis <- apply(sample2, 2, mean) - apply(sample1, 2, mean)
qplot(null_dis)
```

What have we done above? We've simulated an infinitely large population of voles that have a distribution of lifespans similar to that of the cold-reared voles assigned to the control group. The mean μ and standard deviation σ of the simulated lifespan are equal to the observed mean and standard deviation of the lifespans of the control voles. Then, the script:

1. randomly sample 56 values from this population of simulated lifespans and assign to sample1. We sample 56 values because that is the sample size of our control in the experiment.
2. randomly sample 56 values from this population of simulated lifespans and assign to sample2.
3. compute the difference $\bar{Y}_{sample2} - \bar{Y}_{sample1}$.
4. repeat 1-3 100,000 times, each time saving the difference in means.
5. plot the distribution of the 100,000 differences using a histogram

The distribution of the differences is a null distribution. Notice that the mode of the null distribution is at zero, and the mean (-0.11584) is close to zero (if we had set n to infinity, the mean would be precisely zero). *The expected difference between the means of two random samples from the same population is, of course, zero.* Don't gloss over this statement if that is not obvious. The tails extend out to a little more than +100 and -100. What this means is that it would be rare to randomly sample two sets of data from the same

population with mean μ and standard deviation σ and find a difference of, say, -257. In fact, in the 100,000 runs, there were no difference as large as |-257| (the absolute value of -257). The minimum and maximum differences sampled over the 100,000 iterations was -187 days and 201 days.

How do our observed differences compare? Let's focus on vitamin E. The vitamin_E effect is -89.9 days. There are 2110 sampled differences less than the observed value and 2126 greater than the absolute value of the observed value. Together this is 4236 so the frequency of differences from the simulated null distribution that as larger or larger than the observed difference is 0.042 (this compuation includes the observed value in both the numerator and denominator).

8.2.2 *t*-tests

A *t*-test is a test of differences between two values. These could be

1. the difference between the means of two samples (a "two-sample" *t*-test)
2. the difference between a mean of a sample and some pre-specified value (a "one-sample" *t*-test)
3. the difference between a coefficient from a linear model and a value (often zero)

A *t*-test compares an observed *t*-value to a *t*-distribution. The null distribution introduced above was a distribution of mean differences. This isn't generally useful, since the distribution of expected mean differences is a function of sample variability (standard deviations) in addition to sample size and, therefore, a mean-difference null distribution will be unique to every study. A *t*-distribution is a distribution of *t*-values under the null (statistical jargon for "given the null is true"), where a *t*-value is a difference standardized by its standard error. Standardizing by a standard deviation (remember that a standard error is an estimate of the statistic's standard deviation) removes the effect of sample variability on the distribution. A *t*-distribution, then, is only a function of sample size (or "degrees of freedom"). As n increases a t distribution becomes converges on the standard, normal distribution.

The difference between the mean of the vitamin_E treatment and the control treatment is -89.9. A two-sample *t*-test of this difference is

```
##
##  Two Sample t-test
##
## data:  vole[treatment == "vitamin_E", lifespan] and vole[treatment == "control", lifespan]
## t = -1.6275, df = 75, p-value = 0.1078
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -199.97362   20.14029
## sample estimates:
## mean of x mean of y
##  413.4762  503.3929
```

The *p*-value comes from comparing the observed t to a null t distribution and "counting" the values that are bigger than the observed t. These are counted in both tails, because p is the probability of a t more extreme than the observed value, and t can be more extreme in the negative direction and in the positive direction. We can simulate this with a finite, instead of infinite, null distribution using the t-distribution instead of the distribution of mean differences, as above. I hide the script, but its the same as above, except that the *t*-value is computed for each simulated experiment and not just the difference in means.

Hey that looks pretty good! A p value can be computed by counting the number of simulated *t*-values, *including the observed value*, that are more extreme (in either the positive or negative direction) than the observed t. Including the observed t, there are 10904 values that are more extreme than that observed. An approximate measure of p is this count divided by 100,001 (why is 1 added to the denominator?), which is 0.1090389. This simulation-based *p*-value is very (very!) close to that computed from the observed *t*-test.

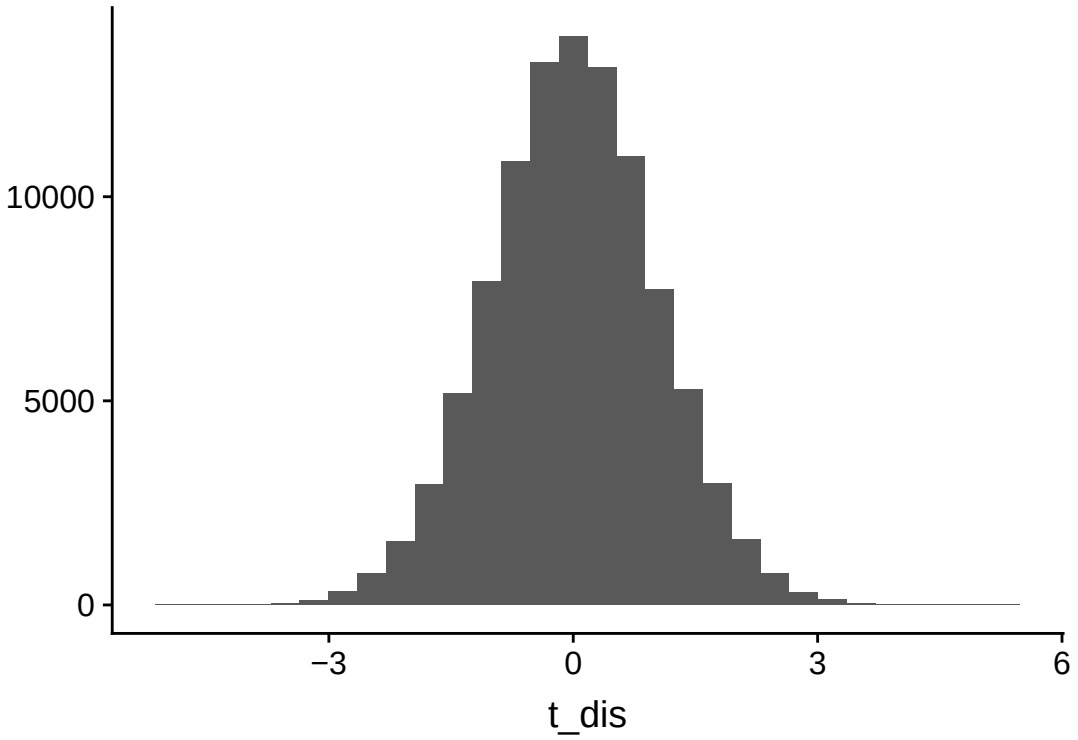


Figure 8.2: Null distribution of t-values. The simulation generated 100,000 t-tests with a true null.

8.3 Statistical modeling instead of hypothesis testing

This chapter is an introduction to a p-value by way of t-tests. I advocate that you analyze t-test like questions using statistical modeling instead of null hypothesis significance testing. The reason is that we learn much more from an estimate of the effect and a CI than from a t and p-value. But, it is also good to know that a t test is a special case of a linear model, and you can get that t and p using a statistical modeling approach should your boss want them (and you cannot convince them otherwise). Let's explore this.

1. Using the emmeans package, compute the effects (differences in means) of vitamin E and vitamin C on lifespan, relative to the control, with their 95% CI and the t and p values for the cold-reared vole data.
2. Compute a separate t-test of vitamin-E vs. control and vitamin C vs. control.

Are the t and p values the same? No! The reason is that the statistical model had three groups and the SE of the difference was computed from the sample standard deviation of all three groups. Each t-test computes the SE of the difference from only the two groups being compared. In general, the SE computed from all three groups is better because it uses more information. This is one reason to prefer the linear model instead of the separate t-tests.

3. To convince yourself that a t-test is a special case as of a linear model, compute the effects of the vitamin E treatment (relative to control) **but exclude the vitamin C data from the model fit**. Now compare the t and p values with the t-test. These should be the same.

4. Now use the default t.test function by deleting "var.equal=TRUE" from the function. Are t and p still equal to those from the statistical model? No! the reason is because the default t.test function uses a modification of the t-test called "Welsch's t-test". This test allows for heterogenity of variances. Several sources argue that one should always uses Welsch's test since it simplifies to the classical t-test when the sample variances are equal. This is true, but only relevant if you're into t-tests. And, we can model heterogenous variances using a statistical model. We'll do this in a later chapter.

5. Use the function `pairwise.t.test` to compute all pairwise t.tests among the three treatment levels. Is the p-value for the vitamin_E - control contrast the same as that if using t.test (with var.equal=TRUE) or the statistical model with vitamin_C data excluded? No! The reason is that pairwise.t.test adjusts the p-values for multiple testing as a default.

Pro tip: Before you use a new R function like t.test or pairwise.t.test, it is really advisable to read the help page and look at the defaults for the parameters! Researchers publish errors because they failed to look closely at what the R function was doing and they think the function is doing something else. Ooops!

8.4 frequentist probability and the interpretation of p-values

8.4.1 Background

There are at least three different meanings of **probability**.

1. **subjective probability** is the probability that an individual assigns to an event based on prior knowledge and the kinds of information considered reliable evidence. For example, if I asked a sample of students, what is the probability that a 30c homeopathic medicine could clear a *Streptococcus* infection from your respiratory system, their answers would differ because of variation in their knowledge of basic science, including chemistry and physics, their knowledge of what homeopathic medicines are, and how they weight different kinds of evidence.
2. **classical probability** is simply one divided by the number of possible unique events. For example, with a six-sided die, there are six possible unique events. The probability of rolling a 2 is $\frac{1}{6}$ and the probability of rolling an odd number is $\frac{1}{2}$.
3. **frequentist probability** is based on the concept of *long run frequency*. If I roll a die 10 times, the frequency of rolling a 2 will be approximately $\frac{1}{6}$. If I roll the die 100 times, the frequency of rolling a two will be closer, but to $\frac{1}{6}$. If I roll the die 1000 times, the frequency of rolling the die will be even closer to $\frac{1}{6}$. So the frequentist definition is the expected frequency given an infinite number of rolls. For events with continous outcomes, a frequentist probability is the long run frquency of *observing the outcome or one more extreme*.

8.4.2 This book covers frequentist approaches to statistical modeling and when a probability arises, such as the p-value of a test statistic, this will be a frequentist probability.

When we do a t-test, we get a p-value. There are several ways to think about this probability. The most compact way is $P(data|null)$, which is literally read as the probability of the data given the null (or "conditional" on the null), but is really short for *the probability of the data, or something more extreme than the data, given that the null hypothesis is true.* The "probability of the data" is kinda vague. More specifically, we mean the probability of some statistic about the data such as the difference in means between group A and group B or the t-value associated with this difference. So, a bit more formally, the probability returned in a t-test is prob$(t \geq t_{obs}|H_0)$. This is the long run frequency of observing a t-value as big or bigger than the observed t-value (the one you actually got with your data) if the null is true. Let's parse this into "long run frequency of observing a t-value as big or bigger than the observed t-value" and "null is true".

A thought experiment: You open a google sheet and insert 12 standard, normal random deviates (so the true mean is zero and the true variance is one) in Column A, rows 1-12. You arbitrarily assign the first six values (rows 1-6) to treatment A and the second six values (rows 7-12) to treatment B. You use the space immediately below these data to compute the mean of treatment A, the mean of treatment B, the difference in means (A - B), and a t-value. Unfortunately, google sheets doesn't have a t-value function so you'd have to compute this yourself. Or not, since this is a thought experiment. Now "fill right" or copy and paste

these functions into 999 new columns. You now have 1000 t tests. The expected value of the difference in means is zero (why?) but the actual values will form a normal distribution about zero. Most will be close to zero (either in the negative or positive direction) but some will be further from zero. The expected t-value will also be zero (why?) and the distribution of these 1000 t values will look normal but the tails are a little fuller. This row of t values is a null distribution, because in generating the data we used the exact same formula for the values assigned to A and the values assigned to B. Now think of a t-value in your head, say 0.72 (remember that t values will largely range from about -3 to +3 although the theoretical range is $-\infty$ to $+\infty$. What is the probability of observing a t of 0.72 *or bigger* if the null is true? Look at the row of t-values! Count the number of $t \geq 0.72$ and then divide by the total number of t-values in the row (1000) and you have a probability computed as a frequency. But remember the frequentist definition is the long run frequency, or the expected frequency at the limit (when you've generated not 1000 or even 1,000,000 but an infinite number of columns and t-values).

Some asides to the thought experiment: First, why "as big or bigger" and not just the probability of the value itself? The reason is that the probability of finding the exact t is 1/infinity, which doesn't do us much good. So instead we compute the probability of finding t as big, or bigger, than our observed t. Second, the t-test probability described above is a "one-tail probability". Because a difference can be both in the positive direction and the negative direction, we usually want to count all the $t \geq 0.72$ and the $t \leq -0.72$ and then add these two counts to compute the frequency of *as extreme or more extreme* values. This is called a "two-tailed probability" because we find extremes at both tails of the distribution. Third, we don't really count $t \geq 0.72$ but take advantage of the beautiful mathematical properties of the theoretical t distribution, which allows us to compute the frequentist probability (expected long range frequency) given the t-value and the degrees of freedom using the t-distribution.

Now what do I mean with the phrase "null is true"? Most people equate "null is true" with "no difference in means" but the phrase entails much more than this. Effectively, the phrase means that the p-value is based on modeling the real data with a theoretical sample in which all the points were randomly sampled from the same distribution and that the assignment of the individual points to treatment was random. This model means the theoretical sample has three properties: First, random assignment to treatment after sampling from the same distribution means that the expected means are the same, or put differently, the expected difference in means between the assigned groups is zero. Second, random assignment to treatment after sampling from the same distribution *also* means that the expected variances of the two groups are equal. And third, random sampling means that the values of each point are independent – we cannot predict the value of one point knowing information about any other point. **Here is what is super important about this**: if we get a really low p-value, any one of these consequences may be untrue about our data, for example it could be that the true means of the two treatment groups really are different, or it could mean it is the variances that differ between the two groups, or it could mean that the data (or technically, the errors) are not independent of each other. This is why we need certain assumptions to make a p-value meaningful for empirical data. By assuming independent error and homogenous (equal) variances in our two samples, a low p value is evidence of unequal means.

8.4.3 Two interpretations of the p-value

Since we want to be working scientists who want to use p-values as a tool, we need to know how to interpret (or use) the p-value to make reasonable inferences and how to avoid mis-interpreting the p-value and making unreasonable or even incorrect inferences. Ronald Fisher, the inventor of frequentist statistics, developed an interpretation of the p-value that is probably most useful for academic and applied research programs. Neyman and Pearson (Neyman-Pearson) gave the p-value a different interpretation, one that is probably most useful for industrial quality control. Today's biology researchers use an interpretation that is an odd hybrid of the two, which often leads to silly inference. Regardless, understanding the distinction between Fisher and Neyman-Pearson will inform how we write up our results in a manuscript. I'll describe these in the context of the two-sample t-test.

8.4.3.1 Fisher's interpretation

Fisher was working in the context of an agricultural field station, the goal of which is to discover better agricultural practices. Does this new fertilizer work better than our old fertilizer? This is the context of much of modern biosciences and clinical medicine. Fisher thought of p as evidence against the null; the smaller the p the better the evidence that the means differ, which, in an experimental context, implies a treatment effect. If an experiment results in a large p-value, we can move on and test other fertilizers. If an experiment results in a small p-value, we want to pursue this new fertilizer more. Do more experiments! Fisher never thought of a single experiment as definitive. The decision to move on or pursue is only partly informed by the p-value and Fisher offered no rule about what p-value lies on the threshold of this decision. When pressed, Fisher might say that $p = 0.05$ is a reasonable threshold.

8.4.3.2 Neyman-Pearson interpretation

Neyman-Pearson thought of p as the necessary and sufficient information to make a decision between accepting the null (or at least not rejecting the null) or rejecting the null and accepting an alternative hypothesis. This decision balances two sorts of errors: Type I (false positives), which they called α, and Type II (false negatives), which they called β. A false positive means the null was rejected but there really is no effect. A false negative means that the null was not rejected but there actually is an effect. α is set by the experimenter and is the long-term frequency (or "rate") of false positives **when the null is true** that the experimenters are willing to accept. This is easily understood in the context of manufacturing. I've just made a batch of beer that I now need to ship. I sample 10 cans and test the quality against a norm. If $p < \alpha$, we reject the null in favor of the alternative – something may be wrong with the batch, it differs from the norm. We throw the beer away. If $p > \alpha$, we do not reject the null, nor the beer! We ship it.

After setting α, the experimenter designs the experiment to achieve an acceptable rate of β. Since β is the false negative rate then $1 - \beta$ is the rate of not making a false negative error, that is, the rate of rejecting the null when there really is an effect. This is called the **power** of the experiment. An experiment with high power will have a low probability of a Type II error. An experiment with low power will have a high probability of a Type II error. Power is partly determined by sample size, the bigger the sample the smaller the p-value, all other things equal (think about why in the context of the formula for the t-value). Power is also a function of α. If we set a low α (say, $\alpha = 0.01$), the test is conservative. We are more likely to fail to reject the null even if the null is false. This is the balance. We want to make sure that we test our batch of beer using enough cans to find a bad batch if it exists, but we don't want to test too many cans because this is a waste of money. An experimenter sets α, computes the sample size needed to achieve a certain level of power $(1 - \beta)$, and then does the experiment.

In Fisher's interpretation, there is no α, no β, no alternative hypothesis, and no sharp decision rule. Instead, in Fisher, p is a continuous measure of evidence against the null and its value is interpreted subjectively by an informed and knowledgeable expert using additional information to make decisions. Neyman-Pearson rejected Fisher's conception of p as evidence against the null and used p as a tool to make a decision that maintains long-term type I error rates at alpha given a certain power. In Neyman-Pearson, p is compared to a threshold, α and this alone makes the decision. In Neyman-Pearson, p is not treated as continuous information. $p = 0.00000001$ is no more evidence to use to reject the null than $p = 0.049$.

8.4.4 Some major misconceptions of the p-value

Setting the type I error rate α to 0.05 is so pervasive that I'm going to simply use "0.05" instead of "alpha" in discussing misconceptions.

8.4.4.1 p is the probability that the null is true; $1-p$ is probability that the alternative is true

Many researchers believe that if $p > 0.05$ then "there is no effect." A frequentist hypothesis test cannot show that an effect doesn't exist, only that the null has a low probablity of producing a test statistic as extreme or more extreme than the observed effect.

Many researchers believe that if $p < 0.05$ then "there is an effect." Again, a frequentist hypothesis test cannot show that an effect exists, only that the null has a low probablity of producing a test statistic as extreme or more extreme than the observed effect.

1. The statement "There is no effect of predators on feeding behavior" is not a valid conclusion of a frequentist hypothesis test.
2. The statement "We found no effect of predators on feeding behavior" is misleading because a frequentist hypothesis test can neither find an effect nor find no effect.

The two errors above are gross misconceptions that are pervasive in the biology literature. A more subtle issue is the belief that a low p-value shows that the researcher's explanatory hypothesis is correct. For example, researchers believe the result "the prey fish fed 14.2 (95% CI: 9.2, 19.2) minutes shorter in the presence of the predator fish" confirms their hypothesis that prey modulate feeding duration as a function of their ability to assess the risk of predation. Some alternative explanations:

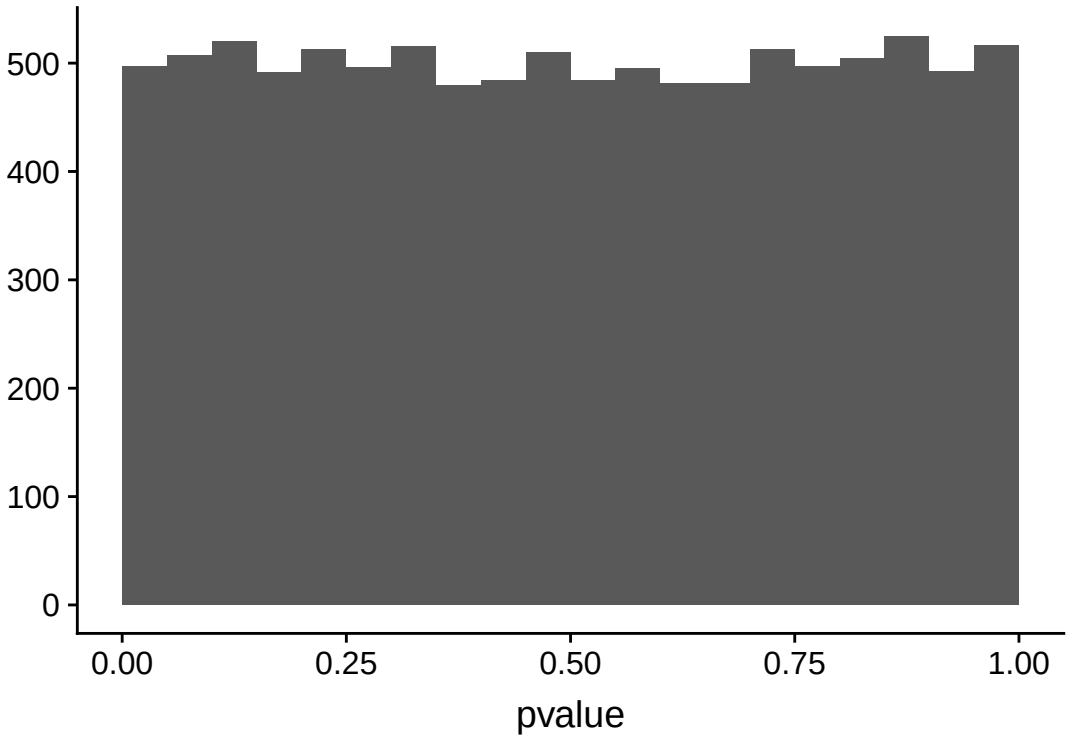
1. The predator fish also competes with the prey fish for the prey fish's food and with less food the prey fish spends less time feeding because it gives up when food density drops below some amount.
2. The predator fish is introduced to the prey tank by hand and odorant molecules from the researcher's hands are detected by the prey and the prey reduces feeding duration because of these odorants.

Importantly, no single experiment confirms an explanatory hypothesis. Instead, alternative explanations require multiple experiments with different controls to "rigrously probe" the preferred hypothesis.

8.4.4.2 The incredible inconsistency of the p-value.

Many researchers believe that a p-value is a precise measure – that if the experiment were replicated, a similar p would result. This belief requires at least two misconceptions. First, if the null were true, then *any* p-value is equally likely. $p = 0.00137$ is just as likely as $p = 0.492$. In other words, if the null were true, the p-value is not replicable at all! Second, the p value is highly dependent on the sample, and can be highly variable among replications, but there is no true p-value, so there can be no estimate or standard error. Let's explore these.

What is the distribution of p-values under the null? I often ask students, "if the null were true, what is the most likely p-value?" or "if the null were true, what kind of p-values would we expect, that is what is the expected distribution". A common answer is $p = 0.5$ is the most likely value and something like a normal curve, except the tails abruptly stop at 0 and 1, is the expected distribution.



The incredible inconsistency of the p-value

How replicable is the conclusion of an experiment if the p-value for a t-test is 0.03? If our conclusion is based on $p < 0.05$, then the conclusion is not very replicable. The simulation below shows the results of 15 replicates of an experiment with true power of 40%. There are five "significant" results (one less than expected) but several replicates have very high p-values.

8.4.4.3 Misconception: 0.05 is the lifetime rate of false discoveries

An important and widespread misconception is that if a researcher consistently uses $\alpha = 0.05$, then the frequency of incorrectly concluding an effect exists, or "discovering" an effect, over the lifetime of the researcher, will be 5%. This is incorrect. This "false discovery rate" is the frequency of false positives divided by the frequency of positives. This differs from the the Type I error rate, which is the frequency of false positives dived by the frequency of tests *in which the null is true*. This frequency depends on the frequency of nulls that are true.

Imagine we test

1. 1000 null hypotheses over a lifetime
2. 60% are true nulls, this means there are 600 true nulls and 400 true effects
3. alpha is 5%. This means we expect to find $p \leq 0.05$ 30 times (0.05×600) when the null is true
4. power is 25%. This means we expect to find $p \leq 0.05$ 100 times (0.25×400) when the null is false
5. We have made $30 + 100 = 130$ "discoveries" (all experiments with $p \leq 0.05$), but
6. 30 of the 130, or 23%, are "false discoveries".

That is 11.7% is the "false discovery rate".

Think about this. If the null is never true, you cannot have a false discovery–every $p \leq 0.05$ is a true discovery (the false discovery rate is 0%). And if the null is always true, every $p < 0.05$ is a false discovery (the false discovery rate is 100%).

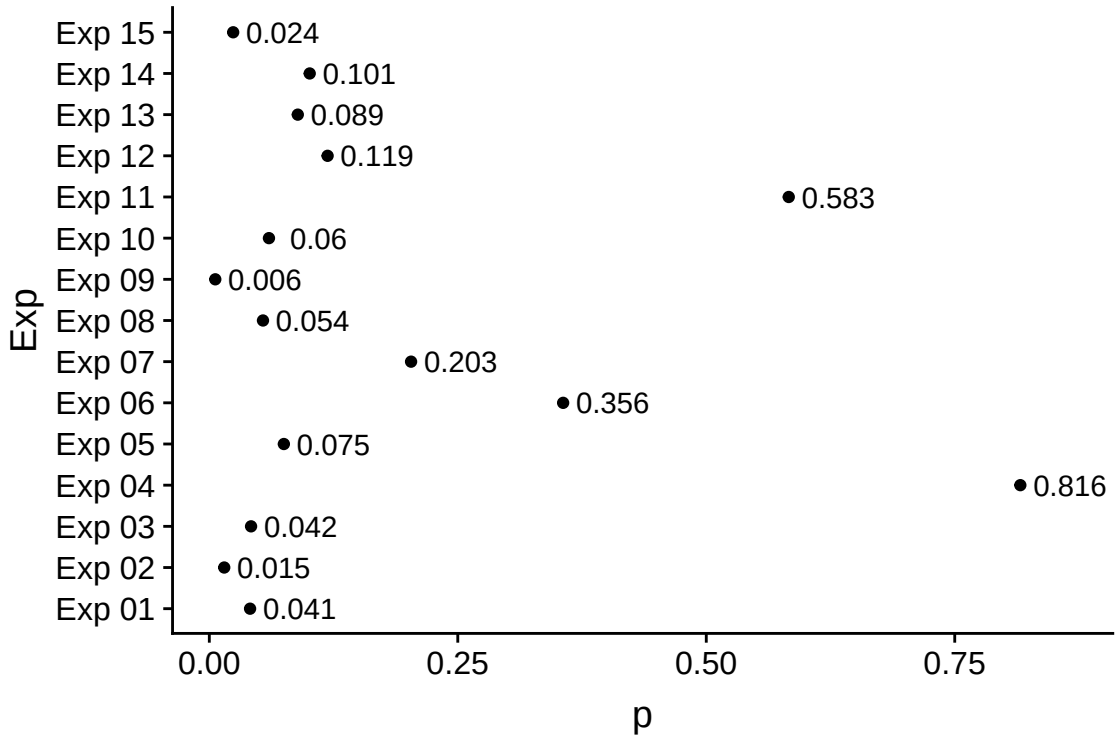


Figure 8.3: Variability of p-values when the power is 0.4

8.4.5 Recommendations

If you are working on basic science research simply report the exact p-value, along with a CI. If $p < 0.05$ (or some other α) do not report this as "significant" – in fact, avoid the word "significant". In the english language, "significant" implies big or important. Small p-values can result even with trivially small effects if n is big or sample variation is small. If p is smaller than say 0.001, then this is pretty good evidence that the data is not a fluke of sampling. But if p is closer to 0.01 or 0.05, this is only weak evidence of a fluke because of the sampling variability of p.

If you are working on quality control then a p value is a useful tool, but is only relevant compared to a decision rule with well-reasoned values of α and β – exact values of p are not very meaningful.

8.4.5.0.1 What the p-value does not mean

1. p is not the probability of the null being true. More formally, this probability is $P(null|data)$ but our p-value is $P(data|null)$. These are not the same. $P(null|data)$ is the probability of the null being true given the data. $P(data|null)$ is the probability of our data, or something more extreme than our data, conditional on a true null.
2. $1 - p$ is not the probability of the alternative
3. p is not a measure of effect size.
4. p in one experiment is not the same level of evidence against the null as in another experiment
5. p is not a great indicator of which is more likely, H0 or H1.
6. If one treatment level has $p < 0.05$ and another treatment level has $p > 0.05$, this is not evidence that the treatment levels have different effects on the outcome.

8.5 Problems

Problem 1 – simulate the distribution of p under the null. There are many ways to do this but a straightforard approach is to

1. Create a $2n \times m$ matrix of random normal deviates with mean 0 and sd 1
2. Do a t-test on each column, with the first n values assigned to one group and the remaining n values assigned to the second group. Save the p-value from each.
3. Plot a histogram of the p-values.
4. What is the distribution? What is the most likely value of p?

Problem 2 – simulate power. Again, many ways to do this but following up on Problem 1. 1. Create a $2n \times m$ matrix of random normal deviates with mean 0 and sd 1 2. Add an effect to the first n values of each column. Things to think about a. what is a good effect size to add? The effect/sd ratio, known as Cohen's d, is a relative (or standardized) measure of effect size. Cohen suggest 0.2, 0.5, and 0.8 as small, medium, and large standardized effects. b. should the same effect be added to each individual? Yes! It is the random component that captures the individual variation in the response. 3. Do a t-test on each column of the matrix, using the first n values in group 1 and the remaining n values in group 2. Save the p-values for each. 4. Compute the power, the relative frequency $p \leq 0.05$. 5. Repeat with different values of n, effect size, and sd, but only vary one at a time. How does power vary with these three parameters?

Problem 3 – grad students only. Simulate the false discovery rate. Explore the parameters: the frequency of true nulls and the power.

Appendix 1: Getting Started with R

8.6 Get your computer ready

8.6.1 Install R

R is the core software

Download R for your OS

8.6.2 Install R Studio

R Studio is a slick (very slick) GUI interface for developing R projects

Download R Studio Desktop

8.6.3 Resources for installing R and R Studio

On Windows

On a Mac

8.6.4 Install LaTeX

LaTeX ("la-tek") is necessary to use the pdf output of R Markdown.

On Windows

On a Mac

8.7 Start learning

8.7.1 Start with Data Camp Introduction to R

Data Camp: Introduction to R (free online course)

8.7.2 Then Move to Introduction to R Studio

R Studio Essentials, Programming Part 1 (Writing code in RStudio)

8.7.3 Develop your project with an R Studio Notebook

Getting Started with R Markdown

Introducing Notebooks with R Markdown

8.8 Getting Data into R

Getting your data into R

8.9 Additional R learning resources

Getting used to R, RStudio, and R Markdown

Link to list of R Studio webinars

Link to set of R package cheat sheets (amazing!)

Bookdown online books

8.10 Packages used extensively in this text

1. ggplot2
2. data.table
3. mvtnorm
4. lme4
5. nlme
6. emmeans
7. readxl
8. reshape2

Data Visualisation chapter from *R for Data Science*

Graphics for communication chapter from *R for Data Science*

Youtube: An Introduction to The data.table Package

Coursera: The data.table Package

Appendix 2: Online Resources for Getting Started with Linear Modeling in R

Regression Models for Data Science in R by Brian Caffo

Broadening Your Statistical Horizons: Generalized Linear Models and Multilevel Models by J. Legler and P. Roback

The Art of Data Science by Roger D. Peng and Elizabeth Matsui

Bibliography