Research Design and Statistics

An Introductory Course

Bradley J. Cosentino

Updated on 2026-02-23

Contents

Preface

Why a new book on statistics? For many years I taught statistics using a standard approach with emphasis on sampling and classical hypothesis testing with frequentist inference. I taught the binomial test, t-tests, Chi-squared tests, and more. Over time I came to realize that my statistics course didn't reflect at all the way that I actually do statistics. I also increasingly got the sense that the standard approach to teaching statistics gave the impression that statistics can be done with a dichotomous key. For example, if the you are associating a continuous response variable to a categorical explanatory variable with two levels, use a t-test, quantify a P-value, and make a conclusion. Statistics is much more interesting and messy than that. Ultimately I wanted to revise so that there's a coherent organization around variations of a general linear model. So that's one feature of this book. All the various "tests" a communicated as variations of a general linear model.

There are two other big changes in this introductory approach to statistics. First, I use both frequentist and Bayesian approaches to inference. I do this for multiple reasons. Although I haven't completely ditched frequnetist methods in my own research, I have increasingly used Bayesian inference. I've moved in this direction for a variety of reasons, but in part because I am sympathetic to some of the criticisms of P-values and null hypothesis testing. I quite like the idea that Bayesian estimates are entire posterior distributions, and I find that the interpretation of Bayesian estimates are more intuitive to students than P-values. Second, I place a strong emphasis in this course on causal inference methods. When I took statistics in graduate school, I was taught many of the variations of regression modeling for different experimental designs, but there was never any discussion about whether the interpretation of regression coefficients were actually appropriate for my scientific goals. The only cost of adding variables to a regression model seemed to be decreased precision of the estimates. In reality, the interpretation of regression coefficients depends entirely on one's causal assumptions about a system, and I have found graphical causal models extremely useful for making causal assumptions clear. In this book, I teach the basics of graphical causal models and how they can be used to inform appropriate parameterization of general linear models given one's scientific goals.

The book uses R programming to illustrate the methods of research design and data analysis introduced throughout the book. If you or students are using R for the first time, I have included a basic introduction to data and R in chapter 3. The introduction to R in that chapter is pretty bare bones, but enough to build on throughout the book. All R code in the book is shaded in gray, and new code functionality that builds on the introductory material is explained throughout the book. If you are teaching a course with students who have a background in R, such as a data analytics or visualization course, then you can potentially skip Chapter 3.

Work in progress

I'm *drafting* this book as a companion to a completely revamped version of my BIOL230 Biostatistics book at Hobart and William Smith Colleges. I emphasize *drafting* here because it really is a work in progress. There will be typos, mistakes, some incoherent sentences, some ideas that need to be clarified, and formatting errors. And I have a few more chapters to write.

If you're reading this book and have some constructive feedback, I very much welcome your comments! Please provide your comments here.

Acknowledgments

Three scientists have really swayed my thinking on how we do and teach statistics: Richard McElreath, Judea Pearl, and Andrew Gelman. Their books, including *Statistical Rethinking* (McElreath), *Book of Why*, and *Regression and Other Stories* (Gelman, Hill, and Vehtari) ultimately pushed me to change my course. I greatly appreciate their work, and many others who have advocated for similar ideas (Bayesian inference, causal modeling, etc.).

Some of the formatting for this book was modified from Michi Tobler's excellent Primer of Evolution.

Copyright

About the Author

Bradley Cosentino is a Professor of Biology at Hobart and William Smith Colleges, where he conducts research on the ecology and evolution of wildlife and teaches courses in ecology, evolution, and statistics.

Chapter 1

Why Statistics? The Problem of Uncertainty

1.1 The nature of science

How do we know what we know? Ultimately this book is about scientific research as one method of advancing knowledge. When we do research, we're interested in seeking the truth. Scientists aren't alone in this truth-seeking venture. Journalists, philosophers, mathematicians, artists, historians, and so on; they have the same goal, just different questions and methods of inquiry. There's even a branch of philosophy called **epistemology**, which aims to seek truth about the methods of seeking truth.

So what distinguishes science from these other fields of truth-seeking? Defining science remains a contentious topic among philosophers, but at its most basic level, science is a process of testing ideas about the natural world with **empirical data**. Like other methods of inquiry, scientists use logic and reasoning to develop research questions, ideas, and expectations, but what separates science from non-science is the confrontation of those reasoned ideas with observations from the natural world.

In one sense, we're all scientists, as Andrew Jaffe aptly points out in his book **The Random Universe**. Daily life requires confronting ideas with data. Is it going to rain today? If you see a lot of cloud cover outside, your suspicion of rain might be increased. Or imagine you had a hard time falling asleep at night. You remember that you had an espresso after dinner, and that makes you think drinking coffee at night might affect your ability to fall asleep. Single observations like this are **anecdotes**, and won't be of much use in isolation. But imagine that you notice you have a hard time falling asleep nearly every time you have an espresso after dinner, no matter what you ate for dinner or what else you might have done during that day. Those repeated observations, and the patterns that emerge from those observations - can be extremely useful for testing ideas. This book is an introduction to the methods of a) designing studies to collect data in a systematic manner (**research design**), and b) analyzing and interpreting those data to provide insight into a research question (**statistics**).

Before we get into the weeds on research design and statistics, I want to make two points about "how we know what we know". First, notice that I defined science as a *process*, not a body of knowledge. Uncovering truths about the universe with science is an aspirational goal. The problem is that there's no certainty about the ideas we test in science. A scientific study can advance our current state of knowledge, but it can't find truth with certainty. We have to be open to the idea that knowledge about the world based on science might be wrong. Science is a process of investigating ideas with data in the presence of **uncertainty**. All knowledge is contingent. What we know at one point in time can be revised or overturned by what we learn from later investigations. If you're looking for a cookbook to find "the definitive answer", this isn't it.

Second, doing science doesn't guarantee the advancement of knowledge. Research isn't a production line where you provide all the inputs and get knowledge at the end. Knowledge is produced through structured inquiry *and* debate with others. Indeed, science is a process that involves logic, reasoning, and structured examination of observations, but it also involves social interactions, including all the cognitive biases of being a human (Hull 1988). The good thing is that knowledge isn't decided by any one person. The impact of your research on knowledge depends on convincing your peers. A scientist may well reach a conclusion in a biased manner, but the great thing about science is that there's a community of other scientists out there, and not all of them share the same biases. If the community is unswayed by the idea, it will be left behind. Knowledge is decided by the collective, not the individual. Jonathan Rauch calls this system of knowledge production **liberal science**, because while it encompasses the sciences, it also includes other methods of inquiry, such as history and journalism. The bottom line is that knowledge is advanced through a *public process* of evaluation. No one study, and no one individual has the final say or authority over the state of knowledge. When you do research, the onus is on you to do high-quality research that will stand up to public criticism and convince your peers.

I say this not to scare anyone off from doing science! Science is fun and rewarding. But it's also hard. Composing a clear research question, designing a systematic approach to collect data, and analyzing and interpreting data are not straightforward tasks! There are compelling arguments that the "publish or perish" pressure in academia has incentivized bad research practices that ultimately contribute to the publication of studies that are wrong [1]. Scientists have paid too little attention to research practices and workflows that prioritize accuracy of scientific inference over throughput. I can look back at some of

[1] See Horton 2015 and McElreath 2023 for starters.

my own publications and find examples where I now questions the decisions I made, and where I'd almost certainly approach the problem differently today. I wish more attention was paid to these issues when I was learning about research design and statistics, and that has been a motivating force to write this book. My hope is that this book will help empower you with basic skills needed to design and conduct high-quality science and to effectively evaluate the science of others.

1.2 Goals of scientific research

In 1986 a study by Menkes et al. was published in the prestigious New England Journal of Medicine on the relationship between lung cancer risk and dietary factors, specifically intake of vitamin A, vitamin E, beta-carotene, and selenium. One finding was that the risk of lung cancer was negatively associated with beta carotene levels in the blood. In other words, lung cancer was observed far less often in people with the highest levels of beta carotene. When summarizing the conclusions of their study, the authors wrote that their study suggests "low serum levels of beta carotene increase the risk of subsequent squamous-cell carcinoma of the lung". The study has been cited 762 times as of this writing.

Ten years later, the New England Journal of Medicine published another study on the relationship between lung cancer risk and beta carotene by Omenn et al., this time showing a **positive** relationship. The relationship was weak, but the direction of the relationship was opposite of that reported by the 1986 study. What gives?

Are either of these studies wrong? Well, that really depends on the goal of each study. If the goal was to test whether beta-carotene has a **causal effect** on lung cancer risk, then surely one of the studies must be wrong. For reasons we'll get into later, the 1996 study had a more appropriate research design for inferring a causal effect of beta carotene, despite the claim I quoted from the authors of the 1986 study.

But what if a cause-and-effect relationship wasn't of interest? Instead, what if the researcher's primary goal was to design a statistical model that could accurately **predict** who is most likely to have lung cancer? The 1986 study found that individuals with the highest levels of beta-carotene had four times the risk of lung cancer as individuals with the lowest levels of beta-carotene. Beta-carotene may not be an important cause of lung cancer, but it might be a useful marker for lung cancer risk if it's associated with other variables that cause lung cancer. For example, perhaps high beta-carotene is found among individuals who exercise a lot and don't smoke, and it's those latter factors that causally explain lung cancer risk.

The takeaway here is that the way we assess the value of a study depends heavily on its research goal. Are we seeking to explain, predict, or describe?

Textbooks and courses on statistics tend to start with an overview of different types of data and proceed to recommend specific statistical tests based on the type of data. They often bypass discussion about the nature of the research question and implications for how the study should be designed. Frankly I've used that approach myself in my own statistics course. But here's the problem: it suggests that *how* you should analyze the data has little relationship to *why* you're analyzing the data. The truth is that the way you collect and analyze data often depends on your research goals.

1.3 Three general goals of scientific research

At the broadest level of investigation, there are three main types of scientific research: explanation, prediction, and description (Hernan et al. 2019, Hamaker et al. 2020. Sometimes we simply want to describe some aspect of the world, other times we want to forecast something in the future, and still other times we want to explain the causes behind certain patterns or outcomes. Let's break down these three goals:

1. **Description:** The simplest of the three, the goal of description is to numerically summarize a quantity of interest. For example, what is the proportion of women in four-year colleges in the United States? What is the typical weight of a pint of Ben and Jerry's ice cream? What is the typical body temperature of healthy adults? In each of these cases, the goal is simply to describe some aspect of the world numerically.

2. **Prediction:** The goal of predictive research is to forecast some future event based on historical or current data. Here the main objective is to accurately project what an outcome will be in the future. For example, weather forecasting, predicting stock market movements, or estimating the growth of a newborn baby are all examples where prediction is the main goal.

   In prediction, the relationship between variables doesn't need to be causal; what matters is how well the variables help forecast the outcome. For example, beta-carotene levels might predict lung cancer risk even if they beta-carotene levels are not a cause of lung cancer. In this sense, studies like the one from 1986 can still be valuable. Problems arise however when the goal of a study - like the 1986 study - is explanation but the research design is more targeted towards prediction.

3. **Explanation (i.e., causal inference):** Here the goal is to understand whether a particular variable is a cause of some other variable. Does the dietary intake of beta carotene affect lung cancer risk? Does being in a wealthy school district affect the likelihood of being admitted to an Ivy League university? Does restoring forest increase biodiversity? These are all questions about the causal relationship between variables.

There is much philosophical discussion about the meaning of **cause**. In this book, I will use “cause” to refer to the impact of an intervention of one variable on some other variable. Examining causation requires one to consider a **counterfactual** world. What would happen to an outcome if the world was different? For example, would the likelihood of being admitted to an Ivy League university differ if individuals in a wealthly school district were actually in a poor school district? For those individuals in a wealthy school district, the counterfactual is being in a poor school district. For individuals in a poor school district, the counterfactual is being in a wealthy school district. In this case, causal inference requires examining the impact of changing the wealth level of a school district on Ivy league admittance.

Causal inference was the primary focus of the 1996 study on beta carotene and lung cancer risk. They conducted an experiment in which they some individuals were given beta-carotene, and others were not. For reasons we will discuss later, the essential element of the study design was the random assignment of the beta-carotene treatment to individuals, which is the defining attribute of what we will call an **experiment**. Then they looked at the outcome, whether or not individuals developed lung cancer. This research design made it easier to make causal conclusions about beta carotene compared to the 1986 study, which was not an experiment. As we will find out, causal inference is more challenging when scientists can’t do an experiment, but it’s not impossible!

I think it’s fair to say that most scientists are interested in causation at some level. As we’ll find out, making causal conclusions can be tricky, especially in disciplines where experimentation is hard or impossible. Sometimes scientists are hesitant to use the word “cause” in their papers, even if explanation is their goal. Instead they’ll use other phrases that clearly imply causation without being direct about it; “X affects Y”, or “X drives Y”, or “X determines Y”, or “X contributes to Y”, or “X plays a role in Y”, etc. Suffice to say that the softened language doesn’t get around the challenges of making causal inferences.

1.4 Research design and statistical analysis depend on your goal

The three main research goals - description, prediction, explanation - are not mutually exclusive. Many research projects involve more than one of these goals, but each has different implications for how you should collect and analyze data. Description is probably the most straightforward in that it often involves quantifying simple summary statistics and reporting patterns. For example, if you want to describe a typical weight of a pint of Ben and Jerry’s ice cream, you need to measure the weights of some pints.

Whereas descriptive research often stops with simple summary statistics, prediction goes a step further in using the data to forecast the unseen. For example, a model might be built using beta carotene levels and other aspects of lifestyle, occupation, and environmental exposure to predict the likelihood of developing lung cancer. As such, prediction tends to be a more complicated goal than descriptive research, requiring statistical procedures to predict the future, and validation to ensure the predictions are accurate.

Causal inference tends to be a bit more complicated. Consider the 1986 study where the goal is to understand whether beta carotene levels affect lung cancer risk, but without an experimental intervention. Should you just look at the raw association between lung cancer occurrence and beta carotene among all the individuals in the study? Should you separate out the smokers from the nonsmokers and look at the association between lung cancer and beta carotene separately in each group? What about other differences between the individuals in the study? Occupation (which vary in exposure to contaminants), radon gas levels in the home, air pollution levels in the neighborhood, lifestyle factors like exercise, etc.. The list of other potential causes of lung cancer goes on and on, and some of these causes may be causally related themselves!

1.5 You can't escape uncertainty in science

No matter what your research goal is, one thing I'd like to convince you of in this book is that using data to draw conclusions about your question is fraught with uncertainty. Too often statistical analysis - the methods we use to leverage data as evidence for our research questions - is presented like a recipe: 1) State a question and hypothesis, 2) Collect data, 3) Run a test, 4) Use a statistical criterion to decide if the result is "significant" (usually a P-value), and 5) Conclude whether the data confirms or rejects the hypothesis. If only it were so simple.

Consider a study evaluating if a vaccine is effective for reducing the prevalence of an infection. The recipe version of statistics starts by assuming the vaccine has no effect (the "null hypothesis"), then asks if the observed data would be unusual if the null hypothesis was true. One problem with this approach is that it gives the impression that statistics is a form of Euclidean proof. In geometry, Euclid started with axioms and deduced conclusions that must be true. In the standard approach to statistic, we begin with assumptions - a null hypothesis and statistical model - and then deduce the kind of observations that would be expected if the assumptions were true. The commonly used P-value is a measure of how surprising the data would be *if* the null hypothesis was true, but too often it's misunderstood as the probability of the null hypothesis being true. This misunderstanding encourages scientists to make binary conclusions - either the null hypothesis is rejected or not - as if statistical analysis provides a definitive verdict on ideas.

The approach to statistics in this book attempts to reverse the emphasis. In philosophy, the use of specific observations to draw general conclusions is called induction. Ultimately that's what we're doing in science. But we will emphasize here that the conclusions we draw based on our observations are never certain. We cannot definitively reject or prove an idea. Why not? The challenge is that our conclusions depend not only on the data we observe, but also on the *assumptions* we make when we connect ideas to data. How did we sample a population? How did we measure the data? What sources of variation did we control or fail to control as part of our study design? The tools we use to connect data to ideas are never perfect, and so honest conclusions about ideas must be made probabilistically. In this book, we will attempt to answer questions not in a binary way, but rather as a matter of degree. How strongly do the data we observe sway our view on the plausibility of an idea? To do this, we will need a language to describe our uncertainty about ideas based on data, and that language will be probability. This is why statistics exists as a discipline. Its goal is not to manufacture definitive answers to research questions, but to provide a rigorous framework for reasoning about ideas in the face of uncertainty.

Chapter 2

Scientific workflow: Connecting ideas to data

The collection and analysis of data is a means to an end. I could devote the next year of my life to visiting city parks and recording the number of bird species that I see at each park, but who cares? In doing science, our goal is to use empirical data to derive understanding about how the world works. Understanding about what, exactly? That's up to you! Knowledge and understanding aren't gained from the data alone. The analysis and interpretation of data is only meaningful in light of a **research question**.

In science, a research question is a question about how the natural world works, and therefore a question that can be addressed with empirical data. In this chapter, I will describe different types of research questions one can ask and elements of what makes for a *good* research question. Then I will provide a high-level overview of how we will go about using data to provide insight into our questions. Together, the process of defining a research question and using data to address the question is what I will refer to as **scientific workflow**.

2.1 Research questions

2.1.1 Clarify your primary goal: description, prediction, or explanation?

A good research question clearly identifies the main task for using data. Do you want to describe some aspect of the world? Do you want to forecast a future event? Or, do you want to explain some phenomenon? Again, these are not necessarily mutually exclusive goals, but many research questions will fall into one of these buckets.

Consider the general topic of birds and city parks again. There are plenty of research questions one could ask about that study system. Table 2.1 provides a classification of research questions by primary goal, illustrating how different types of questions align with descriptions, predictions, and explanations.

Table 2.1: Examples of description, prediction, and explanation questions.

Description	Prediction	Explanation
What is the average number of bird species in city parks?	How many bird species will be in a park with particular attributes, such as size, amount of forest, or number of visitors?	Does increasing the amount of forest in a park cause the number of bird species to increase?
How different are the types of bird species from park to park?	What will the total number of birds be in the park next year?	Does restricting visitor access increase the number of bird species?

2.1.2 Identify the scope of inference: The who, what, when, and where of your study.

Let's take a look at one of the simple descriptive research questions from Table 2.1: What is the average number of bird species in city parks? Now, this isn't the most Earth-shattering research question, but it's simplicity will help make it clear why it's important to be specific about your question. The design of your research - how you go about collecting the data - should follow clearly from your research question. This requires you to clearly define the scope of inference for your question, namely who the question is about, what needs to be measured, when you need to measure it, and where you need to measure it.

The "who" and "what" parts of our research question are rather clear. We want to know about birds (who), and specifically, the mean number of city parks (what). But we need more. Where and when will we measure the mean number of bird species? We've answered the where question in part: city parks. But what city parks? Every city park in the world? City parks in Prague, Czech Republic? Moreover, when will you do the measuring? All we need to do is add some more detail on when and where. For example: What is the *current* mean number of bird species in city parks *within the boundary of Chicago, IL*?

Why do we need such specificity in our research questions? At the most basic level, specificity will help make it clear how to approach the process of collecting data. But at a broader level, the who, what, when, and where also defines the scope of inference of your research question. In statistics, the scope of inference is defined by the **population**. It's the entire group that we want to generalize about when asking a research question. In some cases, the population is rather small. For example, I might define the population as three particular city parks in Chicago: Union Park, Arrigo Park, and Eckhart Park. In other cases, the population will be very large, such as all the parks in Chicago.

Why does this matter? Defining the population, or the scope of your question, affects the types of generalizations you can make based on your research. Suppose I'm working on a research question involving causation, such as "What is the effect of screen time on cognitive performance?" In this case the hypothesis might be that the amount of time people spend on their phones, tablets, computers, and other devices affects there performance on cognitive tests. A study finding an effect of screen time on cognitive performance would be of great interest to the public. But what's the population for the study? If you address this question with a study on children, then you can't use the study to make generalizations about the effect of screen time on cognitive performance in adults. If you are interested in the effect of screen time on cognitive performance in adults, and you collect data from children, you're not really addressing the question.

So let's make this clear: When you state a research question, it is crucial to define the who, what, when, and where of your research question. These components don't necessarily need to be stated all at once in a single sentence. For example, it it's ok to frame your question as "What's the mean number of birds in Chicago city parks?", and then then clarify the timescale and the particular city parks you want to generalize about. Some of those details would be included in the Methods section of a paper on your study. Nonetheless, it's important to think about these questions as you start to articulate your research questions because it will help you design the methods of collecting data, and it will clarify the scope of inferences you can make based on your study.

I teach a statistics class as part of a biology curriculum. Most students who take my class are biology majors, but sometimes students from other majors that require statistics enroll in my course. In any given semester, I might ask: What proportion of the students in my class are biology majors? That's a research question. It may not be a very interesting research question because it's purely descriptive, but simple descriptions are often useful. For example, I could report the fraction of non-biology majors taking my statistics course to administrators who want to know how my department contributes to academic programs other than biology.

There are many examples like this. A company might first want to summarize the wages of their employees before considering a pay increase. A hospital might track the proportion of patients who return with post-surgical complications. A conservation biologist might calculate the average population size of a species in different habitat types.

In this chapter we'll look at some basic numerical and graphical quantities that can be used to describe data. But before we dig in, I want to distinguish between two scenarios with very difficult implications for how we should interpret *descriptive statistics*. Consider my question about the proportion of students in my statistics class majoring in biology. That question is very specific about *who* I want to make conclusions about. What proportion of the students in *my class* are biology majors? This is an example of a research question that is very limited in scope. In a typical semester, I have about 25 students in my statistics class. For my research question, those 25 students represent the **population** of interest.

The population defines the group from which data will be collected and inferences will be drawn. This is a critical concept because the population of interest defines the scope of inferences that can be made with data. All scientific studies involve making inferences with data, so it's critical that we define who we're making inferences about, and therefore who we *can't* make inferences about. For example, suppose I find that 60% of students in my class are biology majors. I can't use that observation to draw any conclusions about the proportion of biology majors in other statistics classes. Similarly, if a company wishes to describe the wages of its employees, the population is clearly its employees, and the data collected can't be used to describe wages at other companies.

2.1.3 Make it interesting

Remember how science is a public process? The knowledge gained from science is not decided by any one individual. Thus, if you want your science to contribute to the advancement of knowledge, your science has to be of interest to others.

I'm not going to claim to be the final authority about what makes a question interesting; any given research question can be of interest to some and not to others. That said, I think interesting questions tend to have broader appeal to the public, addressing longstanding uncertainty or real-world problems (e.g., climate change, social equity). "What factors caused the evolution of *Homo sapiens* from our chimpanzee ancestors?" is an interesting question because many people want to know where human beings came from. "How does the SARS-CoV-2 virus infect people?" was an interesting question in 2019 because it had direct relevance for policy decisions about how to slow the COVID-19 pandemic.

Questions about causal explanation tend to be of greater interest than descriptive questions. Consider the causal questions in Table 2.1. Although they require some more detail on timescale and the population of interest, those questions could be of interest to others because they can inform public policy about urban design and biodiversity (and because a lot of people think birds are cool). If we know that adding forest will change the type of bird species present, then we might prioritize restoring forest to some parks to support certain species of birds.

Contrast the causal questions in Table 2.1 with this question: "How many bird species are in Union Park?". That's a valid research question, but it's purely descriptive and much more narrow in scope, and thus probably of less interest. Now, that's not to say descriptive research questions are never of interest! In 2018 the New York Times, the paper of record in the United States, "published an article" about a research team that was attempting to answer the question "How many squirrels are in Central Park?". That's a purely descriptive research question, but it happened to be about a species and a place that is of broad public appeal.

2.1.4 Make it answerable with data

Remember that the confrontation of ideas with empirical data is really what distinguishes science from non-science. Thus, good *scientific* research questions are ones that can be addressed by collecting and analyzing data. My question about the amount of forest in urban parks and the types of birds can clearly be addressed by collecting data on forest cover and the types of birds present.

The challenge here is that even though a question might address a phenomenon in the natural world, there might not be an obvious type of empirical data that can settle the question. Questions about opinion are challenging for this reason. “What is the best style of pizza?” New York style? Chicago deep dish? Neapolitan? Detroit? This is an extremely important question! The problem is that there’s not a definitive empirical definition of “best”. Whereas I can clearly count bird species and ask where the number of bird species is maximized, I cannot clearly define what makes the “best pizza” because it is a matter of opinion. This is not to say that questions about opinion are off limits. “What style of pizza do people like the most?” is a question that we can address by collecting data on public opinion. Similarly, “What style of pizza is purchased the most in the world?” can clearly be addressed with data. Sometimes a question needs to be revised to make it more precise and tractable.

2.1.5 Ground it in theory

Good research questions focused on explanation tend to follow logically from broader scientific theories. A scientific **theory** is a broad explanation about some aspect of the natural world. In other words, theories are explanations for *why* something happens the way it does. The theory of evolution is a well-substantiated set of ideas about how living things change over time. The theory or relativity is a well-substantiated set of ideas about space, time, and gravity. Good questions tend to fall under these broader theories, connecting a specific situation (the focus of your research question) to a more general idea about how the world works.

Consider the causal question in Table 2.1: “Does increasing the size of parks increase the number of bird species?”. One relevant theory in ecology for this question is the theory of island biogeography. According to this theory, the number of species in patches of habitat should be determined by the balance of species immigrating to the patch and going extinct from the patch. The theory suggests that the size and proximity of habitat patches to other habitat play an important role in affecting immigration and extinction, and thus the number of species. In line with this theory, I might hypothesize that bird diversity in urban parks will be affected by the size of the parks and their proximity to other parks. A **hypothesis** is a specific statement about a particular system of interest that is grounded in a broader theory. In this case, the specific situation is about urban parks and birds, and my idea about urban parks and birds is grounded in island biogeography theory.

I can take this one step further and make a **prediction** about what I expect to observe in the data if my hypothesis is true. For example, I might predict that bird diversity is greatest in the biggest parks and parks that are the least isolated from other parks. If I don't find that pattern in the data, then I might want to revisit the broader theory. Maybe island biogeography theory is not sufficient, and I need to consider other broader ideas that could be applied to the specific case of urban parks and birds. Maybe it's not just the size of the habitat patch that matters, but also the types of vegetation in the patch, for example.

The key here is that your research question should have a logical connection to broader ideas in your discipline. Those broader ideas typically occur in a nested hierarchy. For example, the theory of island biogeography is part of the broader theory of biogeography, which consist of the set of ideas that help us explain the distribution of species on Earth. The logical connection between your research question and hypothesis to these broader ideas is critical to make your question coherent and relevant to other scientists.

2.1.6 Make sure it's feasible

Good research questions are logistically feasible. Consider this extreme case: "What is the effect of removing humans from the Earth on biodiversity?" It's certainly an interesting question to ponder what biodiversity would look like if there were no humans, but this is not a feasible research study. Clearly it is not ethical to experimentally remove humans from the Earth, and even an observational design examining specific geographic areas where humans don't live would be challenging because human impacts on biodiversity occur across massive spatial scales (e.g., global climate change). Any comparison of areas with and without humans also carries the caveat that the places without humans don't really address the question, which is about how *removing* humans affects biodiversity.

Feasibility also matters at a smaller scale. For questions about urban parks and birds, I have to consider whether I can find enough urban parks that vary in the characteristics of interest to answer the question. Do I have the appropriate expertise to identify birds? Do I have access to the resources required to pull off the project (funding, equipment, time, etc.)? If my research question involves a temporal component, such as how urban park characteristics cause change in bird communities, will I be able to monitor bird populations for enough time to see a change, should one occur (potentially years, if not decades)? If I don't have the resources to do sampling on my own, is there an adequate dataset that's publicly available?

So good research questions are ones that are interesting, addressable with data, and grounded in theory, but also logistically feasible. This book will directly address how to design a study and analyze the data given a research question, and I suspect you will have a much better sense for the feasibility of particular research questions after exploring those issues.

2.1.7 Avoid analyzing the data before stating your question

Sometimes researchers try to look for patterns in the data before clarifying their question and hypothesis. In the literature this has been called HARKing, or "hypothesizing after the results are known" (Kerr 1998). The main problem with this research practice is that it increases the risk of making erroneous conclusions. As we will find out in coming chapters, sometimes patterns in the data occur simply by chance, in which case researchers will propose a causal hypothesis for a pattern that actually isn't real. Also remember that patterns in the data don't always reflect direct causal effects, as we saw with beta carotene and lung cancer risk.

So is it every OK to search for patterns in the data? Yes, absolutely. Consider this question: *"Why do some people recover more quickly from viral infections than others?"* The nature of this question is clearly causal. Indeed, the interest is in *explaining* why some people recover faster from viruses than others. Yet, no potential explanation has been proposed! This type of causal question is really a starting point. If there is little theoretical work on the question in the primary literature, then some initial exploratory analysis might be warranted, in which patterns of viral recovery among people are simply described. These questions are sometimes called "reverse causal questions" (Gelman and Imbens 2013, Bailey et al. 2024), in contrast to "forward causal questions" where a clear cause and effect are specified. Reverse causal questions can have an important place in science because identifying patterns can motivate or inform the development of more specific causal hypotheses.

The field of data science is very much focused on pattern recognition in the data, often for the goal of prediction. Consider this question: *"Does exercise predict a person's recovery time from viral infections?"* This question is clearly about prediction - "predict" is right in the question. Here the goal is to determine if the amount of exercise predict's a person's recovery. Maybe people who exercise more recover from viruses more rapidly. That doesn't mean exercise *causes* faster recovery, as exercise might be caused by some other factor that also affects viral recovery. Noncausal associations in the data can be very helpful for making accurate predictions, such as who is most likely to suffer from depression, who is most likely to earn a PhD, or whether the stock market is likely to tumble in the next five years. But it's critical to state up front that the goal in these causes is prediction and *not* causal explanations.

Consider this commonly used example. Suppose you want to predict how many people will go swimming at the beach. You scour through the data, and you find that one of the strongest predictors of whether people will swim is if they are wearing shorts. Great! You can collect data on shorts-wearing that will accurately predict the number of people swimming at your beach. But of course, wearing shorts doesn't cause swimming. Warm temperature is probably a common cause of shorts-wearing and swimming.

So it's really important to state your research question first, and to make it clear what your goal is. Causal questions in particular should be highly specific and grounded in theory. *"Does smoking slow a person's recovery time from viral respiratory infections"*. That's a very specific, forward causal question grounded in theory with a clearly identified cause (smoking) and effect (recovery time). If your questions are more about description or prediction, you simply need to be clear about that goal to avoid confusion.

Explore More

Huntington-Klein. The effect. Lipowski. E.E. 2008. Developing great research questions. Am J Health-Syst. Pharm.

Farrugia, P. et al. 2010. Research questions, hypotheses and objectives. Canadian Journal of Surgery.

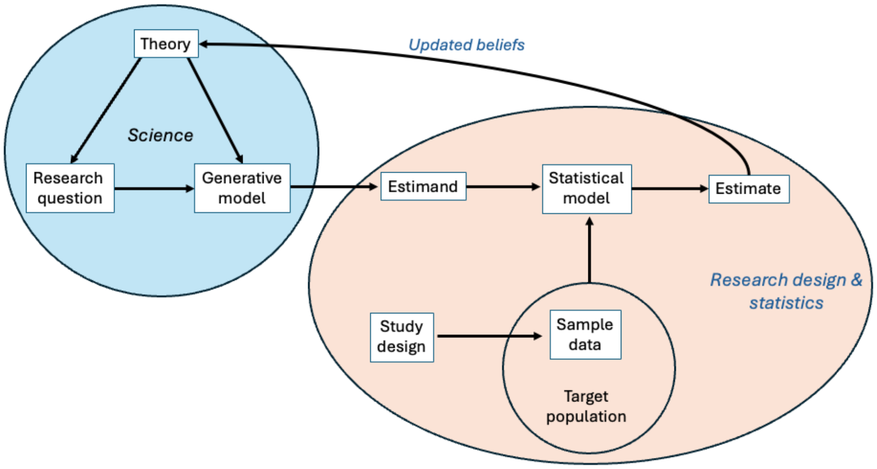
2.2 Connecting ideas to data

Identifying a good research question is hard. What comes after the question isn't much easier, but it is at least goal-driven: once you know what you want to learn, you can design a study and analysis that have a chance of answering it. In this section, I sketch an overview of **scientific workflow**, starting with a research question and showing how it guides decisions about study design, data collection, and data analysis. The goal here is not to provide a deep dive on components of the workflow, but rather to sketch the *big picture* of how to connect data to ideas. We will work through the details of each component of the reserach design and analysis steps in the remainder of the book. As an example for our big picture overview, we'll use a question from the medical literature on diet and blood pressure:

> *Does a DASH diet lower systolic blood pressure compared to a typical diet?*

DASH stands for Dietary Approaches to Stop Hypertension and emphasizes fruits and vegetables. This topic has been studied extensively (e.g., Appel et al. 1997: https://www.nejm.org/doi/full/10.1056/NEJM199704173361601), but the design and data used are synthetic so we can focus on workflow rather than details of any one study.

The image below describes a high-level overview of scientific workflow showing how connections can be made between ideas and data. The figure is a simplified version of a workflow published by Deffner et al. (2024). The arrows illustrate logical connections between different components of a research project, linking elements that are largely "science side" (idea generation) and "statistics side" (data collection and analysis). The general idea is that theory should inform a research question and assumptions about causal relationships among variables relevant to the question (the **generative model**). Given a research question and causal assumptions, we can design a study and analysis, and then use the results to update our understanding of the system. The workflow emphasizes that science is iterative: conclusions from data can revise our beliefs and refine theory.



Let's walk through these components using our example question on the DASH diet and systolic blood pressure.

2.2.1 Theory

Theory represents scientific knowledge that motivates the research question and informs the causal effects we think are plausible. There is a vast literature on cardiovascular disease and the risks of high blood pressure. Theory and prior evidence has suggested that diet should influence cardiovascular health via multiple physiological pathways, and more specifically, that that increasing fruit and vegetable intake (as emphasized in DASH) could reduce hypertension.

Theory also reminds us that diet is not the only factor expected to affect blood pressure. Age, exercise, smoking history, medication use, stress, and many other factors have been linked to hypertension. Indeed, most processes in the natural and social sciences are *multicausal*, meaning there's more than one cause of an outcome of interest. The challenge we face is that some factors create roadblocks to understanding the causal effects we actually care about. As just one example, consider the possibility that diet and exercise both influence blood pressure. That's not necessarily a problem, but it becomes a huge problem if diet and exercise are causally related to each other. For example, perhaps people vary in their degree of health consciousness, and people who are more health conscious tend to eat lots of fruits and vegetables and also exercise a lot. If we find a relationship between diet and blood pressure, is that because diet is having a causal effect on blood pressure, or is it actually an effect of exercise, and people who exercise a lot tend to have more healthy diets? In this example, health consciousness confounds the analysis, and we need to be aware of the issue so that we can use a research design or analysis to addresses the issue.

In this book we will use a variety of examples from across the sciences. I don't expect the reader to have deep knowledge of theory in those various fields, and I will do my best to provide sufficient theoretical background so that we can keep the focus on research design and analysis. But my point here is that effective research design and analysis requires knowledge of theory and prior research in your field.

2.2.2 Research question

What is your question? This one is pretty straightforward:

> Does a DASH diet lower systolic blood pressure relative to a typical diet?

As discussed earlier in the chapter, a good research question is grounded in theory and has a clearly defined scope of inference (who, what, when, and where).

2.2.3 Generative model

A **generative model** is a simplified story about how the data could be generated. In this book, we'll express causal assumptions using **directed acyclic graphs (DAGs)**. DAGs are graphical models that represent variables as nodes and causal relationships as arrows. They include not only the causal effect(s) of interest (here the effect of diet on change in blood pressure), but also those other factors that could confound the analysis.

The structure of a DAG is informed by one's domain knowledge - the body of theory and prior research in your particular field. Recall that in real-world settings, we expect that people vary in health consciousness, and that health consciousness affects both diet and exercise. We can use a DAG to communicate those causal assumptions. Below is a simple DAG that includes the variables health consciousness, diet, exercise, and blood pressure.

```
dag_obs <- dagitty('
dag {
bb="0,0,1,1"

"Health consciousness" [pos="0.25,-0.5"]
Exercise              [pos="0.3,0"]
"Diet" [pos="0.2,0.0"]
"Change in systolic BP" [pos="0.25,0.50"]

"Health consciousness" -> Exercise
"Health consciousness" -> "Diet"
Exercise -> "Change in systolic BP"
"Diet" -> "Change in systolic BP"
}
')

ex <- edges(dag_obs)

# Highlight the confounding structure (backdoor paths into Diet and BP)
highlight_obs <- (ex$v == "Health consciousness" & ex$w == "Diet") |
                 (ex$v == "Health consciousness" & ex$w == "Exercise") |
                 (ex$v == "Exercise" & ex$w == "Change in systolic BP")

rethinking::drawdag(
  dag_obs,
  col_arrow = ifelse(highlight_obs, "firebrick", "black")
)
```

This DAG implies that a simple association between diet and blood pressure change could reflect not only the effect of diet, but also differences in health consciousness and exercise between people who eat different diets. This is the basic idea of **confounding**: other variables can create associations that complicate causal interpretation. The confounding pathway is highlighted in red in the DAG. One value of creating a DAG like this is that it tells the researcher and scientific community alike that the confounding effect of health consciousness must be controlled either as part of the study design or statistical analysis.

2.2.4 Study design

Given a research question and generative model, we should then make decisions about how to collect data. In this book we will differentiate between two main types of study designs: **experiments** and **observational studies**. Essentially,

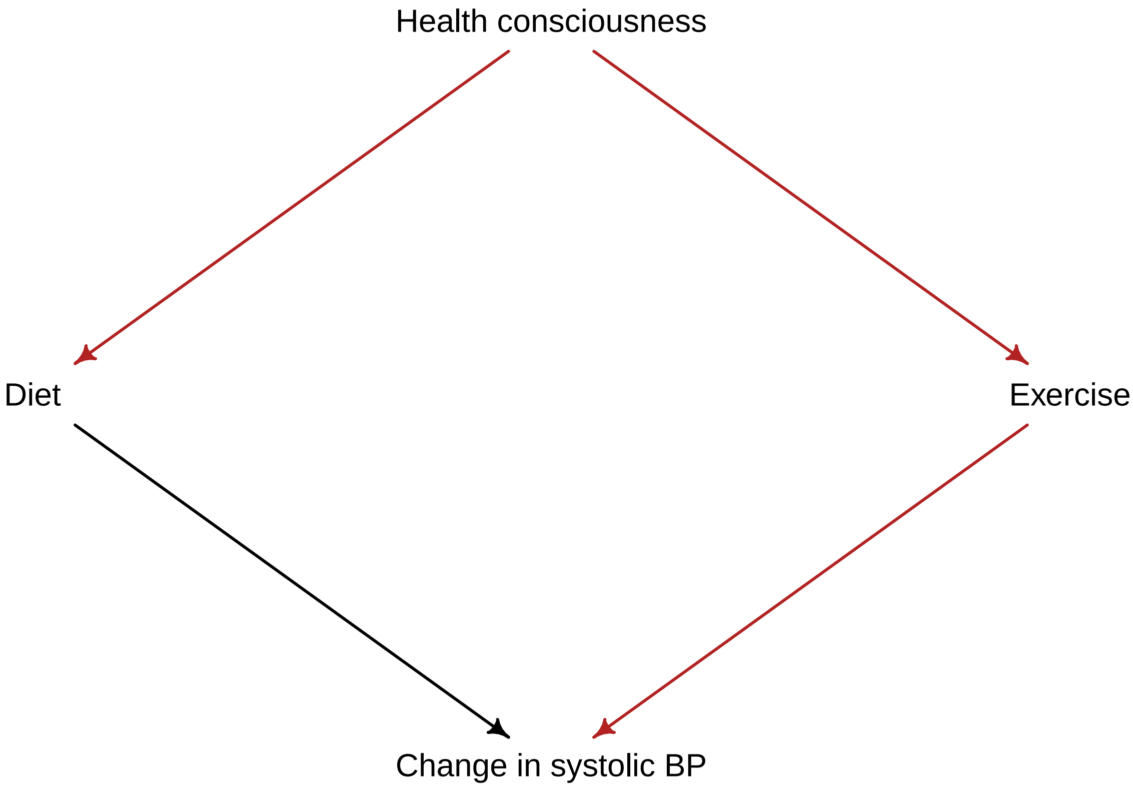


Figure 2.1: Observational setting: health consciousness confounds the relationship between diet and blood pressure change

experiments involve collecting data from individuals to which treatments (such as diet) have been randomly assigned. In contrast, observational studies involve collecting data from individuals in their natural settings without randomly assigning treatments.

Experiments are more powerful than observational studies for identifying causal effects because the process of randomizing treatments to individuals breaks up confounding between the treatment and other variables that affect the outcome of interest. Remember the generative model for the blood pressure example identified health consciousness as a confounding variable. People who eat more fruits and vegetables may be more health conscious and exercise a lot, and an association between diet and blood pressure may in part reflect the causal effect of exercise. But we're not interested in the effect of exercise for this study. We want to know the effect of *diet*.

Experiments offer a design solution. If we can *randomly* assign diets to individuals in an experiment, then we expect to have individuals of varying health consciousness (and thus exercise level) in each diet treatment. Randomization of diet to individuals blocks the arrow from health consciousness to diet in the DAG, such that any association between change in blood pressure and diet would be attributed to *diet*. If we used an observational design where we simply recorded diet and blood pressure among individuals in their natural settings - where people freely choose their diets and exercise levels - we would need to use statistical model that controls for the confounding effect of health consciousness.

We'll learn how to use statistical models with controls later in the book, but for now let's keep things simple and assume we use an experiment. Imagine we recruit 120 adults with elevated blood pressure, and we randomly assign 60 to a DASH diet and 60 to a typical diet. Each person maintains their diet for 8 weeks, and then we measure the change in systolic blood pressure from before to after the study. Pairing the measurements of blood pressure on each individual is a useful design approach because it helps control for other differences among individuals that could affect blood pressure.

2.2.5 Estimand

To use data to answer a research question, we need to identify the **estimand**: the target quantity (or quantities) we want to learn about from the data that will address the research question. For our blood pressure study, we define the estimand as the difference in the average change in systolic blood pressure during the study between the DASH diet and the typical diet. Importantly, estimands are *not* the quantities computed from a particular sample. They are quantities we want to know about in the **population of interest**.

2.2.6 Target population

Every research question needs a defined scope of inference called the **target population** (the who, what, when, and where discussed earlier in this chapter). For our blood pressure study, we might define the target population as adults with elevated blood pressure recruited from clinics in the United States during a defined time period. Target populations vary from study to study, but identifying the target population is essential because it determines who we can—and cannot—make claims about based on the study.

2.2.7 Sample data

One of the fundamental problems of statistics is that we typically cannot measure every individual in the target population. Instead, we collect data from a **sample** of individuals drawn from the target population. For reasons we will discuss later, ideally we select individuals *randomly* from the population to be included in the sample. Because we observe only a subset of randomly-drawn individuals, the quantities we compute based on the sample data will vary from sample to sample. This is a major source of uncertainty in the quantities we estimate from sample data, and it's a topic we will address throughout the book.

Let's take a look at how the sample data might look for our study. Datasets are generally organized in tables with rows representing subjects and columns representing variables. Below are the first eight lines of a synthetic dataset, each having observations for five variables:

- `id` = a unique ID code for each subject
- `diet` = the type of diet randomly assigned to the individual
- `sbp_pre` = systolic blood pressure measured at the beginning of the study
- `sbp_post` = systolic blood pressure measured at the end of the study
- `sbp_change` = difference in systolic blood pressure between the beginning and end of the study.

2.2.8 Statistical model

A **statistical model** is a mathematical description designed to connect the sample data to the estimand and measure uncertainty in the quantities we estimate. Below is an example statistical model that could be used for the blood pressure study. The first line of the model says that the change in blood pressure (c) for each individual i follows a normal probability distribution with mean μ_i and standard deviation σ. The second line says the expected mean blood pressure change for each individual i is defined by the mean for its diet treatment, α_{diet}. The third and fourth lines mathematically represent our prior knowledge about the parameters in the model before looking at the data.

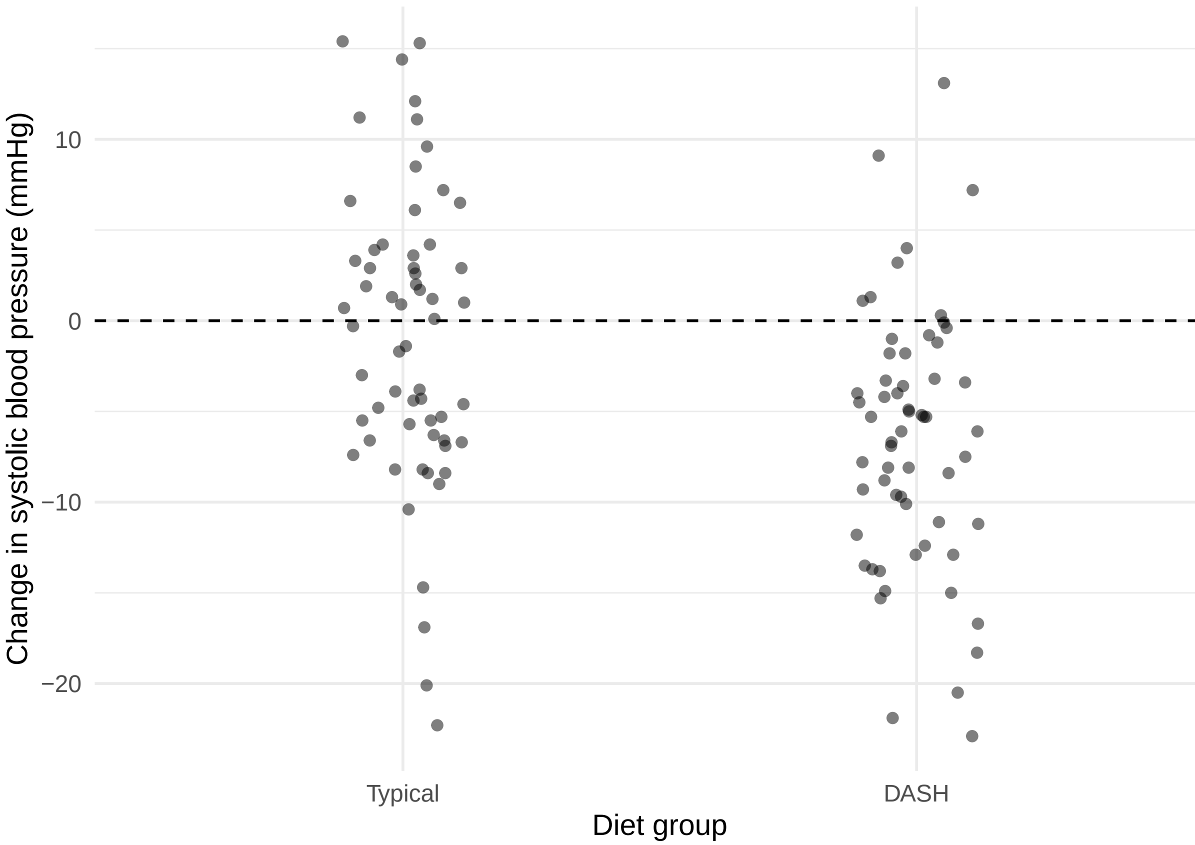
$$
\begin{aligned}
&c_i \sim \text{Normal}(\mu_i, \sigma) \\
&\mu_i = \alpha_{diet} \\
&\alpha_{diet} \sim \text{Normal}(0, 5) \\
&\sigma \sim \text{Exponential}(0.1)
\end{aligned}
$$

Don't worry about the details here. For now, the point is to show you that we'll need a mathematical tool to connect the data to the research idea, and that's the statistical model. Here the model allows us to estimate the mean change in blood pressure for each diet group, which can then be used to estimate the difference in blood pressure change between groups (the estimand). The model also allows for additional sources of variation in blood pressure change among individuals (represented by the standard deviation). These sources of variation could be any number of things, such as differences among individuals in medication, exercise, age, and so on. We don't need to enumerate all those sources of variation individually, but we do need a way to say - mathematically - that diet isn't the only source of variation in blood pressure change. We will build statistical models carefully later in the book.

2.2.9 Estimate

An **estimate** is a numerical quantity computed from the sample data. For this study design, the estimate of the estimand is the the difference in the mean change in systolic blood pressure between the DASH and typical diet groups based on the sample data. The graph below shows the observed change in blood pressure for all 120 individuals in the study, separated by diet type. We can clearly see that individuals within each diet treatment vary in their blood pressure change, which is exactly what we expect because people vary in ways other than diet that might affect their blood pressure. But notice how the cloud of points tends to hover around 0 for the Typical diet, whereas individuals in the DASH treatment tend to have more of a reduction in blood pressure. In other words, it appears qualitatively that the *average* change in blood pressure during the study looks more substantial for the DASH diet than the typical diet.

```r
ggplot(d_toy, aes(x = diet, y = sbp.change)) +
  geom_hline(yintercept = 0, linetype = "dashed") +
  geom_jitter(width = 0.12, height = 0, alpha = 0.5) +
  #geom_point(
  #  data = diet_summ,
  #  aes(x = diet, y = mu),
  #  inherit.aes = FALSE,
  #  size = 3
  #) +
  #geom_errorbar(
  #  data = diet_summ,
  #  aes(x = diet, ymin = lwr, ymax = upr),
  #  inherit.aes = FALSE,
  #  width = 0.15
  #) +
  labs(x = "Diet group", y = "Change in systolic blood pressure (mmHg)") +
  theme_minimal()
```



But we can't just rely on our eyes to detect patterns. We need to estimate the parameters in the statistical model, and specifically the estimand, the difference in blood pressure change between diets. We'll spend a great deal of time learning how to do this later in the book, for now I went ahead and fit the statistical model with the sample data so you can see the types of outputs you can expect. The graph below represents two outputs germain to the reserach question. In the left panel we see probability distributions for the average blood pressure change for each diet group. A probability distribution is just a technical term for saying some values of the mean blood pressure change are more likely than others. The figure suggests that the most likely values of blood pressure change are around -6.5 mmHg for the DASH diet, and -1 mmHg for the typical diet. The probability distributions emphasize that we can't be *certain* about the particular value of the mean blood pressure change in each group, but we can say - based on the data we collected - that some values are more likely than others. And it appears that the mean blood pressure decrease was more substantial in the DASH diet than the typical diet.

How substantial? We can estimate that too. Indeed, the difference in mean blood pressure change between diets is the estimand, so we definitely need to estimate the quantity. The panel on the right represents our estimate, being the probability distribution for the difference in mean blood pressure change between diets. Here we can see it's most likely that the DASH diet reduces systolic blood pressure by about 5-6 mmHg more than the typical diet. Importantly, we see it's extremely unlikely that the two diets have the same mean blood pressure change (represented by the value 0). Again, we can't be certain about the exact magnitude of the difference, but we can make probabilistic statements. Ultimately the inferences we make about our research question with data require *probabalistic reasoning*.

2.2.10 Summary

Overall the synthetic study here suggests it's very *likely* that the DASH diet leads to greater reductions in systolic blood pressure than the typical diet. Thus we've come full circle. We started with a research question informed by theory, used a generative model to inform our research design and analysis, and fit a statistical model that allows us to make an inference about the research question in the target population of interest. Of course the science doesn't end there. This particular study has updated our knowledge on the effect of diet on blood pressure change, and it raises plenty of new questions that could be addressed next (e.g., What explains the variation in blood pressure change within the DASH diet? What are the physiological mechanisms underlying the blood pressure reduction with the DASH diet? How would variation in fruit and vegetable intake affect blood pressure reduction, and how much fruit and vegetable intake should be recommend to maximize blood pressure reduction?). In this way we see that science is an iterative process.

Scientific workflow example

- **Research question:** Does a DASH diet lower systolic blood pressure compared to a typical diet?

- **Theory:** Diet can plausibly influence blood pressure via physiological pathways.
- **Generative model:** Factors like health consciousness can confound the relationship between diet and blood pressure and need to be considered in the study design and analysis to estimate the causal effect of diet.
- **Estimand:** The difference in average systolic blood pressure change under DASH vs typical diet in the target population
- **Target population:** The group we want to generalize about (e.g., adults with elevated blood pressure eligible for the study).
- **Study design:** Randomized experiment assigning participants to DASH or typical diet and measuring systolic blood pressure before and after.
- **Sample data:** The subset of individuals from the population selected for the study. One row per participant including diet group and blood pressure measurements.
- **Statistical model:** A model that connects sample data to the estimand and allows us to describe uncertainty in our estimates.
- **Estimate:** The difference in mean blood pressure change between diets based on the sample data and described with uncertainty.

That's basically the whole ball game in one short chapter. The rest of the book will flesh out the details for each step of the workflow, building up a toolkit you can use to connect your own research questions to empirical data.

Table 2.2: Toy dataset for the DASH example (synthetic values): one row per participant.

id	diet	sbp.pre	sbp.post	sbp.change
1	DASH	156.2	163.5	7.2
2	DASH	137.0	146.1	9.1
3	DASH	151.7	145.5	-6.1
4	DASH	149.0	148.2	-0.8
5	DASH	142.0	120.1	-21.9
6	DASH	119.8	115.6	-4.2
7	DASH	141.2	128.8	-12.4
8	DASH	137.8	128.4	-9.3
9	DASH	151.4	147.8	-3.6
10	DASH	144.3	144.6	0.3
11	DASH	145.1	136.7	-8.4
12	DASH	141.2	154.3	13.1
13	DASH	147.3	144.1	-3.3
14	DASH	147.3	137.7	-9.6
15	DASH	119.4	115.0	-4.5
16	DASH	166.2	157.4	-8.8
17	DASH	157.4	140.6	-16.7
18	DASH	152.6	148.6	-4.0
19	DASH	140.3	139.9	-0.4
20	DASH	158.3	153.3	-4.9
21	DASH	146.1	141.0	-5.0
22	DASH	148.0	141.3	-6.7
23	DASH	133.3	132.1	-1.2
24	DASH	167.6	149.3	-18.3
25	DASH	150.6	151.8	1.3
26	DASH	172.9	157.6	-15.3
27	DASH	170.8	165.5	-5.3
28	DASH	150.7	148.9	-1.8
29	DASH	157.7	151.6	-6.1
30	DASH	170.7	150.2	-20.5
31	DASH	143.7	135.8	-7.8
32	DASH	152.0	146.7	-5.3
33	DASH	146.9	146.9	-0.1
34	DASH	154.0	150.0	-4.0
35	DASH	152.2	148.8	-3.4
36	DASH	164.0	165.1	1.1
37	DASH	157.1	143.5	-13.7
38	DASH	139.3	126.4	-12.9
39	DASH	156.9	148.9	-8.1
40	DASH	140.1	134.8	-5.3
41	DASH	136.1	139.4	3.2
42	DASH	159.3	144.4	-14.9
43	DASH	135.5	120.6	-15.0
44	DASH	153.7	151.9	-1.8
45	DASH	140.0	128.8	-11.2
46	DASH	167.0	157.3	-9.7
47	DASH	158.5	155.4	-3.2

Chapter 3

Introduction to Data and R

At its core, science is a process of seeking knowledge by leveraging observations of empirical data. Given a research question, scientists need to make decisions about how to collect and analyze data to address that question. In this chapter, we begin learning how to work with data. We begin with an overview of the structure of datasets and different types of variables, then I'll introduce R software for storing and manipulating data.

3.1 An introduction to data

In scientific research, **data** consist of information collected via observation or measurement. That's a pretty broad definition! For example, as I write I'm looking out a window and can observe snow covering pine trees in my backyard. That's clearly an observation, so does it qualify as data? I usually make pizza for my family on Friday nights, and to do so I have to measure the weight or volume of flour, water, salt, and yeast. Are those measurements data?

For our purposes, these kinds of observations and measurements don't really qualify as data. In scientific research, data has the distinction of consisting of multiple observations or measurements that can be used to draw conclusions. At my university, courses in statistics address a general education goal on **quantitative reasoning**. Ultimately data are structured measurements or observations that can be used in reasoning, such as making a decision about a hypothesis.

Let's start by examining an example dataset. A **dataset** is simply a collection of data, often with multiple types. This example dataset is about a phenomenon in biology called tail autotomy, which is the ability of organisms like salamanders and lizards to drop their tail when attacked by a predator. The tail continues to move after it's severed, which is thought to be an adaptation to avoid being

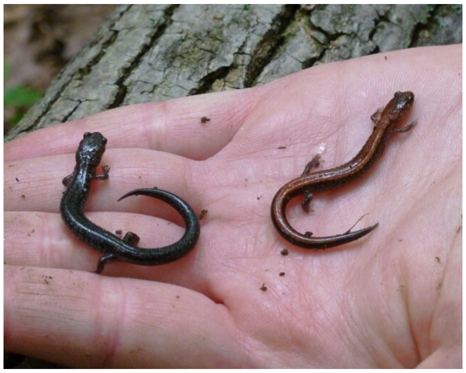


Figure 3.1: Unstriped (left) and striped color morphs of red-backed salamanders.

eaten. In this case, tail autotomy data was collected for red-backed salamanders (*Plethodon cinereus*), including two different color morphs (striped and unstriped) that are known to differ in behavioral and physiological traits (Figure 3.1). The dataset was collected by a former undergraduate student in my lab, Dr. Banan Otaibi, now a surgeon at the University of Arizona. Dr. Otaibi's research question focused on whether tail autotomy behavior differs between color morphs.

3.1.1 Variables and observations

The first thing to notice about the tail autotomy dataset is that it's organized as a table, where each column represents a unique **variable** and each row represents a unique **observation**. A variable is simply a particular type of data, where the observations or measurements can vary. There are eight variables and 40 observations for each variable in the example dataset. Each variable has a unique name. The number of observations for a variable is called the **sample size**, which is denoted n or N.

Data are often organized in this tabular format, whether it is in spreadsheet software like Google Sheets or Microsoft Excel, or as we'll see later in this chapter, statistical software like R. Other names for the tabular format of datasets include a **data matrix** or **dataframe**. They all have the common structure of variables in columns and observations in rows. The observations are also referenced under different names, such as measurements, cases, or individuals.

3.1.2 Types of variables

Variables can be distinguished between two general types, each with subtypes:

Table 3.1: Dataset on tail autotomy in red-backed salamanders.

individual	morph	tail.sec	tail.vel	mass.g	length.cm	easting	northing
16O300	striped	110	3.0	0.8	3.9	350827	4699989
16O301	striped	160	2.3	0.7	4.0	350827	4699989
16O302	striped	250	2.8	0.9	3.9	350831	4699988
16O303	striped	360	3.3	1.0	4.0	350831	4699988
16O304	striped	220	4.6	0.6	3.4	350831	4699988
16O306	striped	120	3.0	0.6	3.3	352128	4702166
16O308	striped	310	3.5	1.2	4.2	352121	4702172
16O309	striped	220	2.6	0.8	3.7	352122	4702174
16O310	striped	360	2.4	0.7	3.6	352116	4702169
16O311	striped	410	2.6	0.9	3.8	352119	4702170
16O312	striped	190	2.2	1.0	3.8	352110	4702167
16O314	striped	460	2.9	0.8	3.5	352089	4702122
16O315	striped	420	2.2	0.5	3.4	352090	4702135
16O316	striped	440	4.0	0.6	3.3	352088	4702129
16O317	striped	400	3.4	0.9	4.0	352071	4702177
17O300	striped	180	3.5	1.1	4.4	351010	4700176
17O302	striped	270	3.8	0.5	3.2	350989	4700122
17O303	striped	50	1.7	0.6	3.6	350962	4700106
17O304	striped	340	3.2	0.8	4.0	350946	4700091
17O305	striped	300	3.2	1.0	4.1	350939	4700088
16O305	unstriped	10	0.5	0.4	3.1	352130	4702168
16O307	unstriped	0	0.0	0.9	3.6	352119	4702157
16O313	unstriped	20	0.8	0.4	2.9	352090	4702123
16O318	unstriped	10	1.1	0.6	3.3	352075	4702046
17O301	unstriped	70	1.9	0.5	3.3	350990	4700114
17O306	unstriped	70	2.0	0.3	2.8	350910	4700059
17O307	unstriped	10	0.9	0.8	4.2	350910	4700057
17O308	unstriped	30	1.7	0.8	4.0	350887	4700003
17O309	unstriped	50	1.2	1.0	3.9	350889	4699994
17O310	unstriped	50	1.3	0.6	3.4	350893	4700009
17O311	unstriped	0	0.0	0.5	3.5	350855	4699997
17O312	unstriped	0	0.0	0.8	4.0	350871	4700003
17O313	unstriped	50	1.6	0.8	3.9	350838	4700072
17O314	unstriped	20	0.6	0.4	3.0	350982	4700003
17O315	unstriped	20	0.6	0.6	3.4	350957	4699969
17O316	unstriped	50	1.8	0.5	3.5	350795	4699993
17O317	unstriped	50	2.0	0.6	3.6	350789	4700005
17O318	unstriped	40	0.8	0.5	3.4	350811	4700018
17O319	unstriped	50	1.9	0.4	3.1	350806	4700007
17O320	unstriped	80	1.4	0.5	3.3	350812	4700030

1. ***Quantitative variables*** Quantitative variables are defined by the observations taking on a range of numeric values. In fact, a synonymous name for a quantitative variable is **numeric**. Some quantitative variables have observations that only take on **discrete** values, such as the number of amino acids composing a protein. In the example dataset, `tail.sec`, the number of seconds an autotomized tail moved, is discrete because seconds was recorded as integer values. Other variables are measured on a **continuous** scale down to any number of decimal places. For example, mass (`mass.g`), length (`length.cm`), and tail velocity (`tail.vel`, the maximum observed number of tail oscillations) are continuous variables. Although there are quantitative variables that by definition can only be discrete (e.g., number of amino acids, number of salamanders in a woodland), sometimes the difference between discrete and continuous is just a matter of measurement. For example. `tail.sec` could have been measured as a continuous variable (e.g., 3.4 sec), but in this case the measurements were rounded to the nearest second, making it discrete.

2. ***Qualitative variables*** Qualitative variables are defined by the observations being classified into different categories. Indeed, qualitative variables are often referred to as **categorical**, or **factor** variables. Like quantitative variables, there are different types of qualitative variables. These subcategories reflect whether or not the categories of a variable have an inherent order. When the categories do not have an order, the variable is called **nominal**. In the example dataset, color morph (`morph`) is a nominal variable because the categories, striped and unstriped, do not have an order. Other categorical variables have order, such as the life stages of ticks (larva, nymph, adult). Life stage is an **ordinal** variable, reflecting the ordering in which individuals move through the different life stages. Qualitative variables can also be differentiated by the number of categories making up the variable. Variables with only two categories, such as `morph` (striped and unstriped) are called **binary** variables.

Sometimes it's hard to classify a variable definitively. For example, consider `individual` in the example dataset. This is simply an alphanumeric code assigned as a unique id for each individual in the dataset. The last few digits of the code are indeed ordered, but the order isn't meaningful.

3.1.3 Relationships between variables

When we ask research questions about causality, we can generally define two types of variables: the variables inducing a causal effect, and the variable receiving the effect. In the scientific literature, people refer to these different types of variables with different names, so it's a good idea to become familiar with some of the most common terms.

A variable inducing a causal effect is often referred to as an **explanatory variable**. For example, we might conduct a study on how different types of medication affect blood pressure. In this case, medication type is the explanatory variable. Another common term people use for an explanatory variable is the **exposure variable**. Here the idea is that the type of medication an individual is exposed to has a causal effect on blood pressure.

The variable receiving causal effects is often referred to as the **response variable**. In the example on medication and blood pressure, blood pressure is the response variable. Another common term for a response variable is an **outcome variable**.

I will generally use these terms throughout the book, but note these are not the only terms you'll see when reading about statistical analyses. For example, some people refer to explanatory variables as **independent variables** and response variables as **dependent variables**. I'm going to avoid using those terms as I tend to think they're confusing because sometimes there are relationships among explanatory variables (i.e., they are *not* independent).

Like questions about causality, questions about prediction are also focused on relationships between variables, with the caveat that those relationships may or may not be causal. The research question on tail autotomy is a good example. In that case, color morph is the explanatory variable, and we measured two different response variables, total tail movement time (`tail.sec`) and initial tail velocity (`tail.vel`). Color morph may well predict tail autotomy, but that relationship is not likely to be directly causal. Perhaps there's a common genetic variant that causes both color morph and the degree of tail autotomy via physiological pathways. We'll explore examples like this in later chapters.

3.1.4 Variable naming conventions and metadata

It's good practice to use a consistent naming style for your variables. Notice that each of the variable names in the example database are lower-case. Some names have multiple words or abbreviations separated by a period (e.g., `tail.vel`). Spaces are not always handleded well in statistical software, so it's good practice to avoid them. Other conventions to separate components of a variable name work just fine; for example one could use an underscore (`tail_vel`), or capitalizing the first letter of each part `tailVel`). The most important thing is to be consistent in your approach.

It's also a good idea to get in the habit of generating metadata for your datasets. Metadata is a description of the dataset, including variable definitions, their units of measure, and more. Here's some metadata describing the variables in the example dataset:

Table 3.2: Variable descriptions for the tail autotomy dataset.

Variable	Description
individual	Unique identification code for each individual salamander.
morph	Color morph (striped or unstriped) for each salamander.
tail.sec	Total tail movement time (seconds).
tail.vel	Initial tail movement velocity, measured as the maximum observed oscillations per second
mass.g	Mass in grams
length.cm	Snout-vent-length in cm
easting	Longitudinal geographic coordinate in UTM zone 18 projection (meters).
northing	Latitudinal geographic coordinate in UTM zone 18 projection (meters).

3.2 Introduction to R

Statistical software is a means to an end. Ultimately our goal is to do good science, and to do that, we must collect, analyze, and interpret data. In this book, I use the statistical platform R to conduct analyses and create graphical outputs of data. You can find many types of software to perform basic data analyses commonly taught as part of an introductory statistics course. Some are free, and some are not. Some have a graphical user interface where you can point and click to select analyses (e.g., SPSS, JMP), some are spreadsheet-based (e.g., Excel, Google Sheets), and others are code-based, requiring the user to write their own computer code scripts to perform analyses. R is 100% free, code-based, and widely used in science. In this section, I will introduce you to the basic data processing skills needed to use R.

3.2.1 Installing R and RStudio

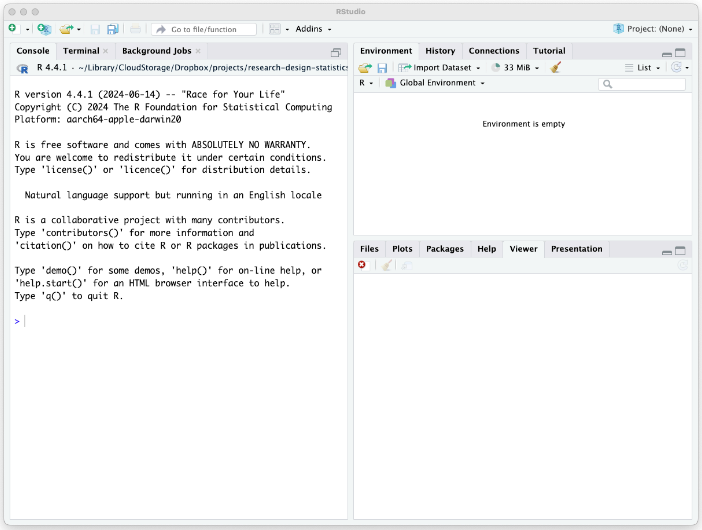
R is a code-based engine for data processing and analysis, and **RStudio** is an interface often used to work in R. Although you can use R by itself, RStudio makes it much easier, allowing you to efficiently write, save, and execute scripts of R code. So while all of the code in this book can be executed directly in R, I strongly encourage working with R via RStudio. Like R, RStudio is free.

If you want to install R and RStudio on your own computer, start by downloading and installing R from https://www.r-project.org/. Click the “CRAN” or “download R” link, and then choose any mirror to access a download link for your platform (e.g., macOS, Windows, Linux). Once you have installed R, head over to from https://posit.co/download/rstudio-desktop/ and click the the link to Install RStudio, then follow the instructions.

There are also ways of using R and RStudio via cloud-based platforms, such as Posit Cloud. With Posit Cloud, you don’t need to install R or RStudio to your hard drive, and your code can be easily shared with others. Although it is often handy to have R and RStudio directly on your hard drive, many courses setup spaces for students to with with R and RStudio via Posit Cloud, so it's worth mentioning here.

3.2.2 The RStudio Interface

When initially opened, the RStudio interface includes three sections (Figure **??**). The console pane is on the left side, which is where you work with the R environment by typing in or executing code. For numerical data processing, output of your code will generally be displayed in the console (though not always automatically). Graphical output will be displayed in the output pane on the bottom right when viewing the Plots tab. Other tabs in the bottom right allow one to navigate to files on your hard drive, examine help documents, and more. The section at the top right is the Environment pane, which shows the objects that you have loaded in R. We'll dig in first on how to compose and execute code in the R Console.



```
R version 4.4.1 (2024-06-14) -- "Race for Your Life"
Copyright (C) 2024 The R Foundation for Statistical Computing
Platform: aarch64-apple-darwin20

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

  Natural language support but running in an English locale

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

> 
```

Figure 3.2: RStudio interface showing the R console (left), working environment (top right), and output (bottom right) panes.

3.2.3 Basic data manipulation in R

3.2.3.1 R as a Calculator

Let's start simple by using R as a calculator. When you start R, you'll see some background on the version of R you're using and some related information. Below all that you'll see a greater than sign (>). This is the command prompt, and it's where you will type in code that you want to execute. For example, you can do a basic arithmetic such as 2+2 by simply typing in 2+2 and pressing enter.

```
2+2
```

```
## [1] 4
```

In the book, I've highlighted R code and like that above in a gray block. Every time I include a chunk of R code, you'll see the command prompt and code to be executed in gray, followed by the output (also in gray), which is the result of the code. So for a simple computation of 2 + 2, you see the command prompt, the 2 + 2 code, and then the output of 4. R prints a number next to the output, here being [1] for the first line of output.

If you want to include notes in your R code that are not executed as code, all you have to do is add a hash sign (#) before the text of your note, like this:

```
2+2 #simple addition
```

```
## [1] 4
```

Here I've added the note `"simple addition"`, which is ignored by R because I included a number sign before the text "simple addition". See what happens if you don't include the number sign to specify your note:

```
2+2 simple addition
```

```
## Error in parse(text = input): <text>:1:5: unexpected symbol
## 1: 2+2 simple
##         ^
```

The dreaded red warning text! When output is highlighted in red, that's usually R's way of telling you there's a problem. Here the problem is that R doesn't know what to do with the word "simple". There's no number sign signifying "simple addition" as a note that should be ignored, so R assumes it's part of your code. Also note that R doesn't even get to "addition", it stops executing your code at "simple" because it doesn't know how to proceed.

OK, so that's some background on how to execute a simple arithmetic function and include a note. Now go ahead and perform some basic arithmetic computations:

```
4*3 #simple multiplication
```

```
## [1] 12
```

```
20/4 #simple division
```

```
## [1] 5
```

```
log(42) #the natural log function
```

```
## [1] 3.73767
```

```
sqrt(4) #the square root function
```

```
## [1] 2
```

```
4^3 #raising 4 to the third power
```

```
## [1] 64
```

We're on our way! In the preceding examples, you can see that some text in R is actually meaningful and does not lead to an error. For example, `log(42)` computes the natural log of 42, and `sqrt(4)` computes the square root of four. In this case, `log` and `sqrt`" are built-in functions in R. More on that soon!

3.2.3.2 Objects

As you can see, it's pretty straightforward to use R as a calculator, but R can do much more. One of the most useful aspects of R is storing data as objects. As a very simple example, suppose we want to assign a value of 4 to a variable that we'll call `x`:

```
x <- 4
```

Here we have defined `x` as 4 with the left arrow, which is a less than sign followed by a dash. The arrow indicates the flow of information, with x on the left being defined as 4. There's nothing special about "x" here. For example, I could have defined "y" as 4:

```
y <- 4
```

Or, you can define the object `Taylor.Swift` as 4:

```
Taylor.Swift <- 4
```

Note that I didn't include a space between "Taylor" and "Swift". Try it with a space and see what happens:

```
Taylor Swift <- 4
```

```
## Error in parse(text = input): <text>:1:8: unexpected symbol
## 1: Taylor Swift
##             ^
```

No good. R doesn't like spaces, so when I define objects that involve more than one word, I usually either include a period or underline between them, or I don't include any characters between them but distinguish components of a variable name by capitalization.

```
Taylor.Swift <- 4
Taylor_Swift <- 4
TaylorSwift <- 4
taylorSwift <- 4
```

Notice that R is case sensitive! The objects `TaylorSwift` and `taylorSwift` are unique!

OK, great. So I've defined a whole bunch of objects as the value 4. Wait. How do I know each of these objects is 4? Well, type the object name into the R prompt and execute it, and you should see the result is 4:

```
x
```

```
## [1] 4
```

```
Taylor.Swift
```

```
## [1] 4
```

Cool! So R will store numeric values - data - symbolically. Quick note: R is case-sensitive. See what happens when you don't capitalize the T and S in `Taylor.Swift`:

```
taylor.swift
```

```
## Error:
## ! object 'taylor.swift' not found
```

The red text of death. Bummer. This is going to be a major source of frustration as you develop your coding skills. The smallest errors, like a lower-case character when it should be upper-case, will cause your code to fail. If you're hoping to work on your attention to detail, this is going to be a good way to do that!

When we define objects, we can start to use them in functions. For example, what's the square root of `Taylor.Swift`?

```
sqrt(Taylor.Swift)
```

```
## [1] 2
```

Clearly it's 2, because `Taylor.Swift` is 4, and the square root of 4 is 2. How about `x + y`?

```
x + y
```

```
## [1] 8
```

Right on - `x` and `y` were both defined as 4, so `x + y` is 8. Note that we can define these output of these arithmetic functions as new objects:

```
z <- x + y
z
```

```
## [1] 8
```

Here we defined "z" as `x + y`, so `z` is 8. Note that objects can be used to pretty much any degree of complexity:

```
z**(x+y)/(Taylor.Swift - x**y*z)
```

```
## [1] -8208.031
```

Finally, objects don't have to be numbers. There are different types of objects in R. All of the objects we just defined are called numeric, because they are just numbers. We can also define objects as characters, which are simply strings of text and defined by quotation marks:

```
> ## Character objects have quotation marks
> singer <- "Taylor Swift"
> singer
[1] "Taylor Swift"
```

Another type of object that we'll encounter a lot in statistics are factor variables, which are categorical variables. I will describe those later.

3.2.3.3 Functions

R has built-in functions that allow you to quickly perform calculations. We've already seen this, such as when we quantified the square root of 4 with the `sqrt` function. Remember that functions are usually called in R by specifying the name of the function followed by round brackets that enclose the arguments of the function:

```
> sqrt(4)
[1] 2
```

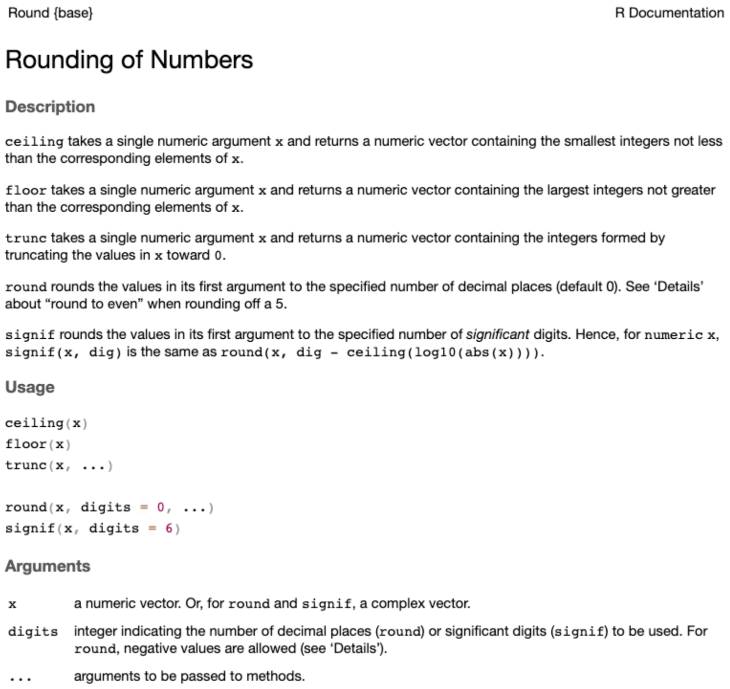
The `sqrt` function is very simple in that it has a single argument, which is the numerical value for which you want to quantify the square root. Other functions have multiple arguments. For example, suppose you want to round the value of 3.147 to a single decimal place. You can use R's built-in `round` function to do that, which has two arguments. The first argument is the value `x` you want to round, and the second argument is the number of digits to round to. When you apply a function with multiple arguments, the arguments are separated by a comma:

```
> round(x = 3.147, digits = 1)
[1] 3.1
```

How did I know which arguments are included in the `round` function? Built-in functions have supporting documentation that you can read to learn about the function and its arguments. To read the documentation, simply add a question mark in front of the function name, and execute that code:

```
?round
```

When you execute the code, the bottom right panel in RStudio will show you the "Help" document for the round function. You'll see an "Arguments" section, and there you'll see that the round function has two arguments (Figure 3.3.



Round {base} R Documentation

Rounding of Numbers

Description

`ceiling` takes a single numeric argument `x` and returns a numeric vector containing the smallest integers not less than the corresponding elements of `x`.

`floor` takes a single numeric argument `x` and returns a numeric vector containing the largest integers not greater than the corresponding elements of `x`.

`trunc` takes a single numeric argument `x` and returns a numeric vector containing the integers formed by truncating the values in `x` toward `0`.

`round` rounds the values in its first argument to the specified number of decimal places (default 0). See 'Details' about "round to even" when rounding off a 5.

`signif` rounds the values in its first argument to the specified number of *significant* digits. Hence, for `numeric` `x`, `signif(x, dig)` is the same as `round(x, dig - ceiling(log10(abs(x))))`.

Usage

```
ceiling(x)
floor(x)
trunc(x, ...)

round(x, digits = 0, ...)
signif(x, digits = 6)
```

Arguments

`x`	a numeric vector. Or, for `round` and `signif`, a complex vector.
`digits`	integer indicating the number of decimal places (`round`) or significant digits (`signif`) to be used. For `round`, negative values are allowed (see 'Details').
`...`	arguments to be passed to methods.

Figure 3.3: Documentation for the round function.

So we see here that the `round` function requires a numerical value `x` to round and the number of `digits` to round to. When you execute code with a function, you can go ahead and use the name of each argument and then specify the value of the argument following an equal sign, like I did above. If you are naming the arguments, the order in which you present the arguments does not matter:

```
> round(x = 3.147, digits = 1)
[1] 3.1
> round(digits = 1, x = 3.147)
[1] 3.1
```

You don't have to name the arguments when executing a function, but there's a catch. When you apply a function and specify the value of the necessary arguments, you have to specify the arguments in the order in which the function expects (x and then digits):

```r
round(3.147, 1)
```

```
## [1] 3.1
```

If you use the reverse order, you get a different answer:

```r
round(1, 3.147)
```

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

Note that arguments for a function often have default values. For example, the `digits` argument in the `round` function defaults to 0 if not specified, meaning that `round(3.147)` will round 3.147 to 0 decimal places. The default values can be found in the documentation under the Usage section.

```r
round(3.147)
```

```
## [1] 3
```

There are many, many functions in R. For example, the function `class` makes R show what type of object you are dealing with:

```
> class(y)
[1] "numeric"
>
> class(singer)
[1] "character"
```

3.2.3.4 Vectors

We know that datasets are made up of one or more variables, each with multiple observations. In R, we can store variables with multiple observations as **vectors**. A vector in R usually corresponds to a single variable (one column of your dataset). For example, suppose we have data on body temperature for five individuals of different ages:

We can use the concatenate function, `c`, to create vector for age and temperature:

Table 3.3: Example vectors for age and body temperature.

Age	Temperature (Fahrenheit)
22	98.2
28	99.1
34	99.3
43	98.4
50	98.9

```
>
> # The c() function combines individual values into a single vector
> age <- c(22, 28, 34, 43, 50)
> age
[1] 22 28 34 43 50
>
> ## and now a vector for temperature
> temp <- c(98.2, 99.1, 99.3, 98.4, 98.9)
> temp
[1] 98.2 99.1 99.3 98.4 98.9
```

We can ask R to show us all the values of a vector by simply calling the name of the object, as I've done above. We can also ask R to report particular observations of a vector by using square brackets. For example, to have R report the fifth observation of the temperature vector:

```
> temp[5]
[1] 98.9
```

Note that the temp vector has five observations. What happens if we ask for the sixth observation?

```
temp[6]
```

```
## [1] NA
```

Here R reports "NA", which means "not available". This is simply R's way of telling you that the value you asked for doesn't exist.

If you want to view multiple observations for a vector, you can specify multiple observations with the concatenate function, or you can use a colon to show consecutive observations:

```
>
> ## Show the 3rd and 5th observation:
> temp[c(3,5)]
[1] 99.3 98.9
>
> ## Show the 3rd through the 5th observation:
> temp[3:5]
[1] 99.3 98.4 98.9
```

If you want to view observations while excluding particular observations, you can do that by adding a minus sign in front of the observation(s) you want to exclude:

```
>
> ## Show all but the third observation:
> temp[-3]
[1] 98.2 99.1 98.4 98.9
>
> ## Show all but the third and fifth observation
> temp[-c(3,5)]
[1] 98.2 99.1 98.4
```

We can easily apply arithmetic functions to vectors. For example, suppose I wanted to know how many degrees each individual's body temperature is above or below what's considered the "normal" value of 98.6 °F. Here I'll just ask R to subtract 98.6 from each value in `temp`, and then save that output as a new object called `temp.deviation`:

```
>
>
> temp.deviation <- temp-98.6
>
> temp.deviation
[1] -0.4  0.5  0.7 -0.2  0.3
```

Here we can see that the first individual's temperature is 0.4 degrees below normal, the second individual is 0.5 degrees above normal, and so forth.

3.2.3.5 Matrices

Recall that we have body temperature data for individuals of different ages. So far we have created separate vectors for each variable, but each vector has the same structure (5 observations). Remember that datasets of multiple variables are usually organized in a tabular format, with columns representing different variables and rows representing the multiple observations of each variable. In R, we can combine vectors into a single data table called a **matrix**. Matrices are made of rows and columns, just like a spreadsheet, where the rows represent observations, and the columns represent different objects. Let's make a matrix for our age and temperature data using the `cbind` function, which combines objects into multiple columns. We'll save this new table and name it `patient.table`:

```
>
> patient.table <- cbind(age, temp)
>
> patient.table
     age temp
[1,]  22 98.2
[2,]  28 99.1
[3,]  34 99.3
[4,]  43 98.4
[5,]  50 98.9
```

We can use square brackets to access observations in a matrix, but now we have to consider that we have two dimensions of data: rows and columns. To extract observations from a matrix, we use a square bracket with row and column values separated by a comma. For example, let's say we want to view the age of the second individual. The second individual is in row 2, and age is the first column in our matrix:

```
>
> ## show the age of the second individual
> patient.table[2,1]
age
 28
```

What if we wanted to view the age AND temperature for the second individual. In this case, we can just leave the column value blank, which R interprets as requesting all the columns:

```
>
> ## show all data for the second individual
> patient.table[2,]
 age temp
28.0 99.1
```

We can also view all the observations for a single variable. Let's say we want to view just the temperature data again. Here we will leave the row value blank but specify the second column for temperature:

```
>
> ## show the temperature values
> patient.table[,2]
[1] 98.2 99.1 99.3 98.4 98.9
```

What if we want to add more variables? Let's say we have the body mass index (BMI) for each individual and want to add it to the matrix. We could just create a vector `bmi`, and then combine it with the other two vectors, or combine it with the `patient.table` matrix. Either will produce the same output:'

```
>
> ## create a bmi vector
> bmi <- c(20, 24, 21, 23, 24)
>
> ## combine each vector to create a table
> patient.table.new <- cbind(age, temp, bmi)
>
> ## or just combine the original table with the new bmi vector
> patient.table <- cbind(patient.table, bmi)
>
> patient.table
     age temp bmi
[1,]  22 98.2  20
[2,]  28 99.1  24
[3,]  34 99.3  21
[4,]  43 98.4  23
[5,]  50 98.9  24
```

Sometimes we don't have complete data for every variable. In R, remember that missing data are recorded as `NA` ("Not Available"). For example, suppose we have data on height (inches) for all but the third individual in our dataset. We would specify `NA` for that individual:

```
>
> ## create a height vector
> height <- c(65, 71, NA, 68, 66)
>
> ## add to the matrix
> patient.table <- cbind(patient.table, height)
>
> patient.table
     age temp bmi height
[1,]  22 98.2  20     65
[2,]  28 99.1  24     71
[3,]  34 99.3  21     NA
```

```
[4,]  43 98.4  23     68
[5,]  50 98.9  24     66
```

3.2.3.6 Data types in R

Recall that we can generally differentiate variables by their type, either quantitative or qualitative. Each of the four variables we've created so far (`age`, `temp`, `bmi`, `height`) are quantitative. R defines variable types by their **class**, and you can ask R to return each object's classification with the **class** function;

```
>
> class(age)
[1] "numeric"
> class(temp)
[1] "numeric"
> class(bmi)
[1] "numeric"
> class(height)
[1] "numeric"
```

We can see R identifies each of these vectors as **numeric**, which is synonymous with quantitative. Qualitative variables in R are usually classified as **character** or **factor** variables. For example, let's create a nominal qualitative variable **sex**, defining each individual in our dataset as male or female:

```
## create a character for sex
sex <- c("female", "female", "male", "female", "male")
class(sex)
```

```
## [1] "character"
```

Notice that `sex` is classified as a character because the observations consist of text rather than numbers. The other common way of defining and analyzing nominal variables in R is as a factor. We can change the object `sex` from a character to a factor variable by using the `as.factor` function:

```
## create a character for sex
sex <- as.factor(sex)
class(sex)
```

```
## [1] "factor"
```

When an object is stored as a factor, you can ask R to show you all the levels (i.e., categories) for the variable with the `levels` function:

```
levels(sex)
```

```
## [1] "female" "male"
```

Factor variables in R can also consist of levels that are numbers. For example, suppose that we assigned each individual in our dataset to one of three treatments, named “1”, “2”, and “3”. When we initially define the treatment object, R will interpret the class as numeric, but we can change it to a factor:

```
## create treatment variable with numeric codes
trt <- c(1,2,3,1,3)
class(trt)
```

```
## [1] "numeric"
```

```
## now coerce to factor
trt <- as.factor(trt)
class(trt)
```

```
## [1] "factor"
```

```
levels(trt)
```

```
## [1] "1" "2" "3"
```

When a variable is defined as a factor (or character), functions that treat the variable as numeric will not work. For example, we can’t numerically add different factor levels of `trt`, even though those levels are stored as numbers. R will tell us that mathematical functions applied to factor data doesn’t make sense:

```
trt[1] + trt[2]
```

```
## [1] NA
```

3.2.3.7 Data frames

Remember that datasets are organized in tabular format, as we’ve seen with a matrix in R. One of the downsides of using matrices in R is that they are restricted to a single type of data object, for example all numeric or all character objects. If we want to add the nominal variable `sex` to our data table, we can use a **data frame**, which are flexible enough to include objects of different class which is a categorical variable made of characters ("male", "female"). Data frames in R are useful for storing full datasets, including multiple variables of different types. Here we can use the function `cbind.data.frame` to combine our matrix of numeric vectors with factor object defining the sex of each individual:

```
>
> ## add to the matrix
> patient.table <- cbind.data.frame(patient.table, sex)
>
> patient.table
  age temp bmi height    sex
1  22 98.2  20     65 female
2  28 99.1  24     71 female
3  34 99.3  21     NA   male
4  43 98.4  23     68 female
5  50 98.9  24     66   male
```

There are some useful functions to inspect the data frame. For example, if we want to see a list of variables in the data frame, their class, and the first few observations, use the `str` function. You can see R reports that we have five variables, including four numeric and one factor, and it reports the name of each variable and the first five observations.

```
>
> str(patient.table)
'data.frame':   5 obs. of  5 variables:
 $ age   : num  22 28 34 43 50
 $ temp  : num  98.2 99.1 99.3 98.4 98.9
 $ bmi   : num  20 24 21 23 24
 $ height: num  65 71 NA 68 66
 $ sex   : Factor w/ 2 levels "female","male": 1 1 2 1 2
```

If you just want to see the first few observations in tabular format, use the `head` function:

```
>
> head(patient.table)
  age temp bmi height    sex
1  22 98.2  20     65 female
2  28 99.1  24     71 female
3  34 99.3  21     NA   male
4  43 98.4  23     68 female
5  50 98.9  24     66   male
```

One of nicest things about data frames is that we can call particular objects from the data frame by using their names. This is done by using the dollar sign, $. For example, if we want to view the observations of height, specify the name of the data frame, then a dollar sign, then the variable name:

```
>
> patient.table$height
[1] 65 71 NA 68 66
```

Note this is the same as calling the fourth column in a matrix as we did before, just more user friendly:

```
>
> patient.table[,4]
[1] 65 71 NA 68 66
```

We can also use the bracket notation specifying variables by names:

```
>
> ## view the values for the variable height
> patient.table[,"height"]
[1] 65 71 NA 68 66
>
> ## view the values for age and height
> patient.table[,c("age", "height")]
  age height
1  22     65
2  28     71
3  34     NA
4  43     68
5  50     66
```

We can do anything we did before with matrices, such as extracting a subset of observations, or doing arithmetic:

```
>
> ## view the values for age and height for the second through fourth individual
> patient.table[2:4,c("age", "height")]
  age height
2  28     71
3  34     NA
4  43     68
>
> ## view all the values for each variable for the second individual
> patient.table[2,]
```

```
  age temp bmi height    sex
2  28 99.1  24     71 female
>
> ## quantify deviations from 98.6 for the temperature values:
> patient.table[,"temp"]-98.6
[1] -0.4  0.5  0.7 -0.2  0.3
```

3.2.4 Scripting

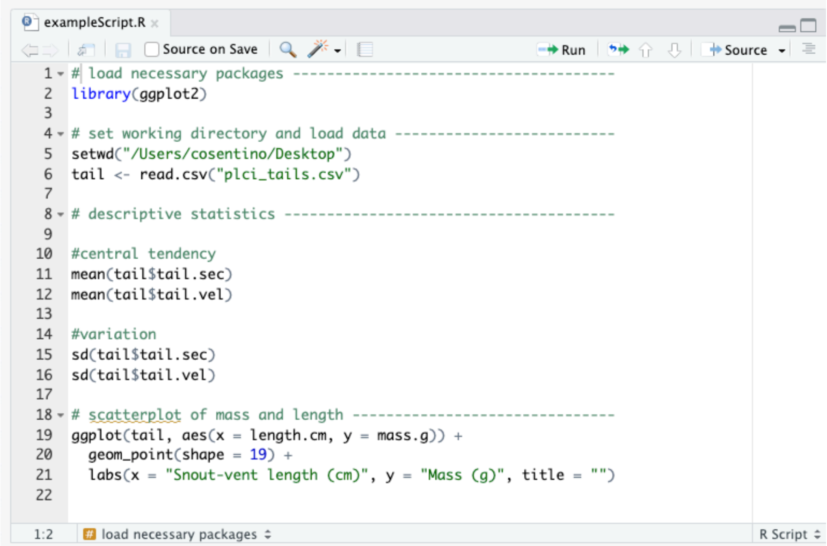
One last R skill that I'd like to briefly cover is scripting. Doing analysis by coding is great for a variety of reasons, but one of the best reasons to code is to make your science reproducible with a script. When you execute a set of functions for an analysis, you can save the code to execute those functions in a script, which is basically a text file (with an .R extension).

Suppose you're working on an analysis for an hour and then the power goes out. Well, if you were scripting your analysis, all you have to do is load your script and execute all the functions up to the point where you left off. A script allows you go back and easily make changes, and it allows others to see *exactly* how you performed an analysis. Indeed, most scientific journals are now requiring that authors publish their code used to conduct analyses to be completely transparent about how the analyses were performed.

How do you make a script? It's really easy in RStudio? In your toolbar, just click File, New File, and then R script. That will open a text editor in the top left panel of RStudio. From there you can start writing your code. I strongly recommend that you add notes to your code to describe what the code is doing. These notes don't have to be extensive, but it's very useful to help reorient yourself, or orient someone else for the first time, to what the code is doing.

For example, Figure 3.4 shows a simple script to perform some basic descriptive statistical analysis on the salamander tails data. I included some notes to delineate different sections of the script. The dashes are not necessary, but I often use them because they allow me to visually demarcate the different sections of the script, and RStudio recognizes those sections and allows you to navigate among the different sections using the drop-down in the bottom left corner of the script panel.

How do you execute the code in a script? It's easy! Just highlight the code that you want to execute, then click the “Run” button in the top right corner of the script panel. You'll see the code executed in the R console in the bottom left panel, along with any generated output.



```
# load necessary packages --------------------------------------
library(ggplot2)

# set working directory and load data --------------------------
setwd("/Users/cosentino/Desktop")
tail <- read.csv("plci_tails.csv")

# descriptive statistics ---------------------------------------

#central tendency
mean(tail$tail.sec)
mean(tail$tail.vel)

#variation
sd(tail$tail.sec)
sd(tail$tail.vel)

# scatterplot of mass and length -------------------------------
ggplot(tail, aes(x = length.cm, y = mass.g)) +
  geom_point(shape = 19) +
  labs(x = "Snout-vent length (cm)", y = "Mass (g)", title = "")
```

Figure 3.4: Unstriped (left) and striped color morphs of red-backed salamanders.

Chapter 4

Describing data

Now that we have a basic understanding of data, in this chapter we look at elementary approaches for describing data graphically and numerically with statistics. At a basic level, a statistic is just a numerica description of data. Description may not make for the most thrilling science, but simple descriptions are essential for decision-making. For example, politicians may use the poverty rate to inform decisions about budget priorities, including funding for food assistance, educational programming, and other services. Without simple descriptive statistics like the poverty rate, we risk decisions being made based on **anecdotes**, which are individual observations often made by a single person. A single politician who lives in a wealthy part of town may think the poverty rate is low based on his personal experience. Statistics, when based on sound study designs, provide a more accurate description than any one individual's experience.

4.1 Defining the population

Even when the research goal is something simple like description, remember that it remains critical to define the population of interest when analyzing and interpreting the data. The population defines who the data should be collected from, and who we can draw inferences about. In this chapter we will investigate descriptive research questions where the population of interest is small enough that we can collect data from everyone in the population.

For example, consider a small municipality with 1000 residents. The population of interest is this single municipality, and it is small enough that everyone in the population can be measured when describing quantities of interest. In cases like this, the research question can be answered with near certainty. If 87 of 1000 residents meet the definition for poverty, then the poverty rate in this municipality is simply

$$\frac{87}{1000} = 0.087$$

Now we can't be *completely certain* that 0.087 is the correct proportion. Perhaps the records used to determine the poverty status of residents are inaccurate. But, setting aside the issue of measurement error, we can often be confident that the degree of uncertainty about descriptive measures is limited when everyone in the population is measured. As we'll see in the coming chapters, there is much more uncertainty when we can't measure everyone in a population, and we will look at methods to minimize and quantify that uncertainty. For now, we will keep things simple and focus on the different ways we can describe data.

4.2 Loading data into R

In 2013 there were 336,776 flights that departed airports in New York City (NYC) to destinations in the United States (Wickham 2022). We'll begin by describing some basic information about these departing flights, such as their departing times, departure delays, and more. Because the dataset includes all departing flights of interest in 2013, the descriptive statistics we quantify essentially represent the truth about this population.

To explore the data, the first thing we need to do is load the dataset into R. There are multiple types of files that can be loaded into R. A common file type is the **comma separated values (CSV)** file. CSV files have data organized in columns separated by commas, with separate rows for each row in a data table. CSV files can be generated with a simple text editor, or in spreadsheet software like Excel or Google Sheets.

I created a CSV of the NYC flight data called `nycflights13.csv`. To load the CSV into R, you need to tell R where the file is stored. This location is called the **working directory**, which is simply a folder on your computer. For example, suppose I stored the CSV directly on my desktop. I would begin by setting the working directory to my desktop, using the `setwd` function:

```
> setwd("/Users/cosentino/Desktop")   #Mac example
> setwd("C:/Users/cosentino/Desktop") #PC example
```

Once the working directory is set, I can then load the CSV with the `read.csv` function, naming the results data frame `d`:

```
> d <- read.csv("nycflights13.csv")
```

If you prefer not to set a generic working directory for a script, you can always load files by specifying the entire directory:

```
> d <- read.csv("/Users/cosentino/Desktop/nycflights.csv")   #Mac example
> d <- read.csv("C:/Users/cosentino/Desktop/nycflights.csv") #PC example
```

Note for Posit Cloud users

If you're working in RStudio on Posit Cloud, the working directory is automatically set to the root of your project when you open it. That means you usually do **not** need to use `setwd()`. As long as your data files are saved inside the project (for example in a `data/` folder), you can just use a relative path: `sq <- read.csv("data/nycflights13.csv")`

Another common data format you can load into R is an **.RData** file. This type of file is specific to R and has one or more data object stored directly in the file. The ability to store multiple objects is a nice feature of RData files.For example, an RData file could include two different dataframes. This can be handy when working with multiple datasets that are related to each other as part of an analysis. I generated an RData file that has the `nycflights13` dataset stored as a data frame named `d` data frame, which you can load in the following way using the `load` function:

```
> load("nycflights13.RData")
```

Once you've loaded the RData file in this way, the `d` data frame should appear in your working environment.

4.3 Inspecting the dataset

Once you've loaded a file and named a data frame for it, it's a good idea to inspect the structure of a new data frame to confirm that everything loaded properly and as you expect it:

```
> str(d)
tibble [336,776 x 19] (S3: tbl_df/tbl/data.frame)
 $ year          : int [1:336776] 2013 2013 2013 2013 2013 2013 2013 2013 2013 2013 ...
 $ month         : int [1:336776] 1 1 1 1 1 1 1 1 1 1 ...
 $ day           : int [1:336776] 1 1 1 1 1 1 1 1 1 1 ...
 $ dep_time      : int [1:336776] 517 533 542 544 554 554 555 557 557 558 ...
 $ sched_dep_time: int [1:336776] 515 529 540 545 600 558 600 600 600 600 ...
 $ dep_delay     : num [1:336776] 2 4 2 -1 -6 -4 -5 -3 -3 -2 ...
 $ arr_time      : int [1:336776] 830 850 923 1004 812 740 913 709 838 753 ...
 $ sched_arr_time: int [1:336776] 819 830 850 1022 837 728 854 723 846 745 ...
 $ arr_delay     : num [1:336776] 11 20 33 -18 -25 12 19 -14 -8 8 ...
 $ carrier       : chr [1:336776] "UA" "UA" "AA" "B6" ...
 $ flight        : int [1:336776] 1545 1714 1141 725 461 1696 507 5708 79 301 ...
 $ tailnum       : chr [1:336776] "N14228" "N24211" "N619AA" "N804JB" ...
 $ origin        : chr [1:336776] "EWR" "LGA" "JFK" "JFK" ...
 $ dest          : chr [1:336776] "IAH" "IAH" "MIA" "BQN" ...
 $ air_time      : num [1:336776] 227 227 160 183 116 150 158 53 140 138 ...
 $ distance      : num [1:336776] 1400 1416 1089 1576 762 ...
 $ hour          : num [1:336776] 5 5 5 5 6 5 6 6 6 6 ...
 $ minute        : num [1:336776] 15 29 40 45 0 58 0 0 0 0 ...
 $ time_hour     : POSIXct[1:336776], format: "2013-01-01 05:00:00" "2013-01-01 05:00:0
```

We see there are 336,776 observations and 19 variables. Some of the variables are numeric, such as the departure delay (`dep_delay`), and others are characters, such as the destination airport (`dest`). We also see some new data classes in this file type. For example, multiple variables are defined as integer (`int`). These are simply discrete quantitative variables. The difference from a numeric variable in R is largely inconsequential; it's just R's way of differentiating between quantitative variables that do or do not have decimals. We also see the variable `time_hour` is stored as `POSIXct`, which is a format for dates and times.

4.4 Describing single variables

4.4.1 Qualitative variables

Describing single variables in isolation is really a simple exercise. What are the possible values the variable could take on, and what is the relatively likelihood of those values? In other words, describing a single variable is an exercise in characterizing its **distribution**. What values are most likely, which values are least likely, and how much variation is there among the values?

Flights from NYC depart from three major airports: Newark Liberty International Airport (EWR), LaGuardia Airport (LGA), and John F. Kennedy International Airport (JFK). The origin airport is defined by the variable `origin`, which we see is classified as a character (`chr`). How might we describe this variable? The first thing we could do is look at the different levels present in the `origin` variable. The `unique` function will show you each unique level present in a character or factor variable:

```
> unique(d$origin)
[1] "EWR" "LGA" "JFK"
```

Here we can see the airport codes for each of the three NYC airports. Now we might want to know how many flights are departing from each airport, or whether most departing flights from one of the airports. In R we can see the **frequency distribution** of a categorical variable with the `table` function.

```
> table(d$origin)

   EWR    JFK    LGA 
120835 111279 104662 
```

A frequency distribution is simply the raw count of each observation. We can see each airport had between 104,000 and 121,000 departing flights in 2013. These are staggering numbers. EWR has the most departing flights, but not be much. The distribution of departing flights appears relatively even among the three airports.

We can quantify the *relative* distribution of observations more directly with a **relative frequency distribution**. In other words, how common are departing flights from one airport relative to the other airports. Relative frequency of a qualitative variable is called a **proportion**, quantified as

$$p = \frac{x}{n},$$

where p is the proportion, x is the observed number of observations for the category of interest (often called **successes**), and n is the total number of observations in the dataset. Let's quantify the proportion of flights departing from EWR. We know there were 336,776 departing flights(n), and 120,835 were from EWR (x), so the proportion of flights from EWR in 2013 was

$$p = \frac{120835}{336776} = 0.359,$$

We see that just over one third of departing flights came from EWR. Proportions are probabilities, so one way of interpreting the outcome here is that, of all departing flights from NYC in 2013, the probability of any one flight departing from EWR was 0.359. Proportions can be converted to percentages by multiplying the proportion by 100. Of the departing flights in 2013, 35.9% were from EWR.

What about the proportion of flights departing from JFK and LGA? Rather than computing the proportions for each airport separately, we can ask R to compute the proportions using the `prop.table` function:

```
> prop.table(table(d$origin))

      EWR       JFK       LGA 
0.3587993 0.3304244 0.3107763 
```

Here we wrapped the `table` function of the frequency of each coat color in the `prop.table` function, and now we can see the entire relative distribution of primary fur color. Most flights originated from EWR (35.8%), followed by JFK (33.0%) and LGA (31.1%).

Patterns in data are often easier to see and communicate graphically than numerically. Rather than reporting a table of frequencies or relative proportions, we could create a bar graph. R has built-in functions that can be used to create graphs, but in most cases for this book, we will use functions from a specialized package called `ggplot2` developed specifically for creating graphs. When you want to use functions from a specialized package in R, you first have to install the package with those functions with the `install.packages` function. The main argument of the `install.packages` function is a character vector of the packages you want to install. If you only want to install a single package, just write the name of the package as a character:

```
install.packages("ggplot2")
```

When you execute the code, you'll see R will download the package and install it. After you've installed a package, you don't have to install it again, but to use functions from the package you do need to load the package with the `library` function:

```
library(ggplot2)
```

Once you execute this code to load the package, you can then use any function from `ggplot2` just as you'd use functions from base R. The main function for creating graphics in `ggplot2` is called `ggplot`. Let's walk through the basic steps of using the `ggplot` function, in this case to create a frequency distribution plots of departing flights by NYC airport in 2013.

1. The first thing we need to do is create a dataframe organizing the data that we want to plot. In this case we're plotting the frequency of departing flights originating from NYC airports, so we'll create a data frame of these frequencies:

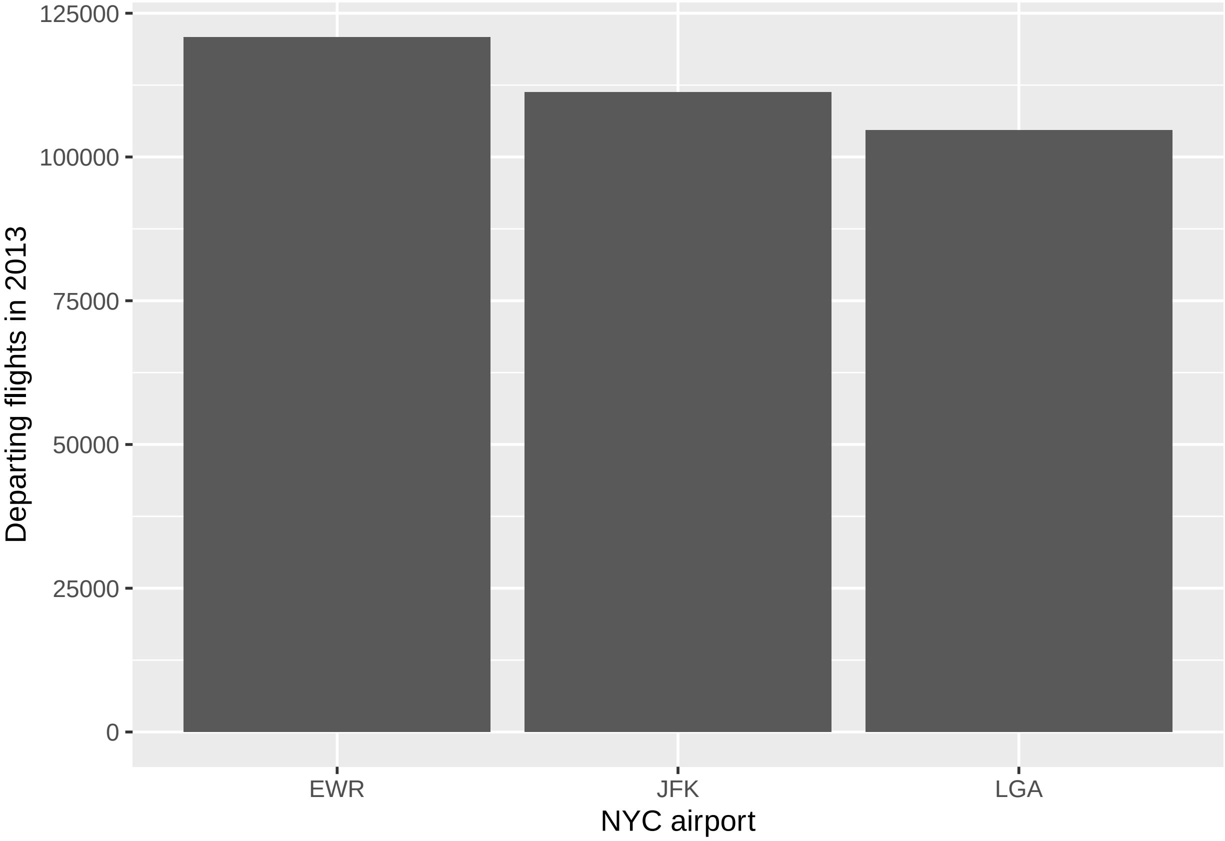
```
origin_freq <- as.data.frame(table(d$origin))
colnames(origin_freq) <- c("origin", "n")
origin_freq #display to see how the data are organized
```

```
##   origin      n
## 1    EWR 120835
## 2    JFK 111279
## 3    LGA 104662
```

2. Next we use the `ggplot` function to code our plot. The first argument in the `ggplot` function should specify the data frame containing the data

(origin_freq), and we usually include a second argument mapping variables to axes. After the ggplot function, we add a set of hierarchical layers, each specified with the + operator. Here's the full code needed for a basic bar graph:

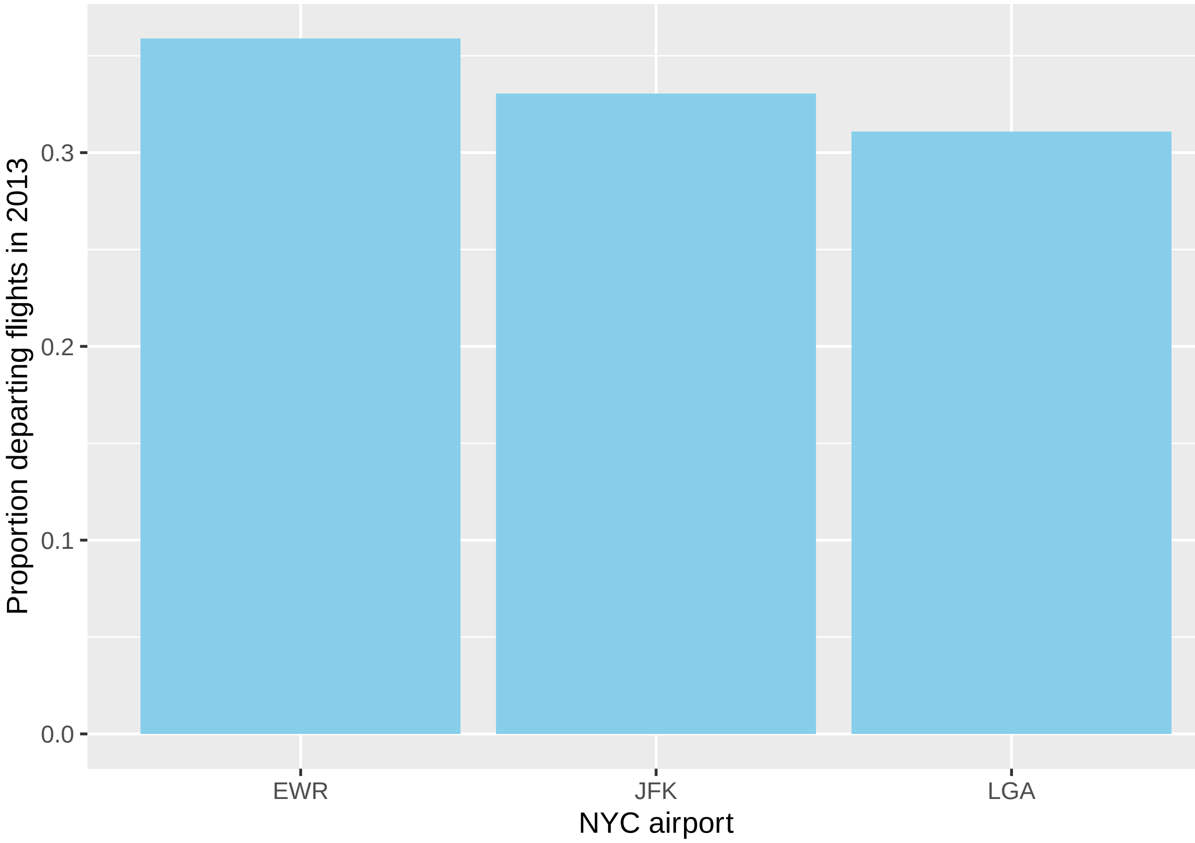
```
ggplot(origin_freq, aes(x = origin, y = n)) +
  geom_col() +
  labs(x = "NYC airport",
       y = "Departing flights in 2013")
```



We see the code created a graph with two axes, the horizontal x-axis and the vertical y-axis. The variables are mapped to the x and y axes by the aes function. The geom_col layer is included to specify a bar chart where the actual frequencies of each category are displayed on the y-axis, and the labs layer lets you define the x- and y-axis labels. Note that the aes function can be added as a layer outside the ggplot function. Including aes as an argument within the ggplot function is more common because it applies the axis mapping as the default for all subsequent layers.

What if you want the y-axis to show proportions instead of frequencies? Easy. Simply quantify the proportions when creating your dataframe:

```
origin_prop <- as.data.frame(prop.table(table(d$origin)))
colnames(origin_prop) <- c("origin", "p")

ggplot(origin_prop, aes(x = origin, y = p)) +
  geom_col(fill="skyblue") +
  labs(x = "NYC airport",
       y = "Proportion departing flights in 2013")
```



Note that I added a fill argument here in geom_col to specify a different color for the bars. There are many more arguments you can use to customize these figures. Throughout the book I'll add various options like this to plots and point them out.

4.4.2 Quantitative variables

Let's turn our attention to describing quantitative variables. For this section, we'll narrow our focus on departing flights operated by Alaska Airlines departing Newark, so first we will subset the dataset to flights where the `carrier` is `WN` and `origin` is `EWR`:

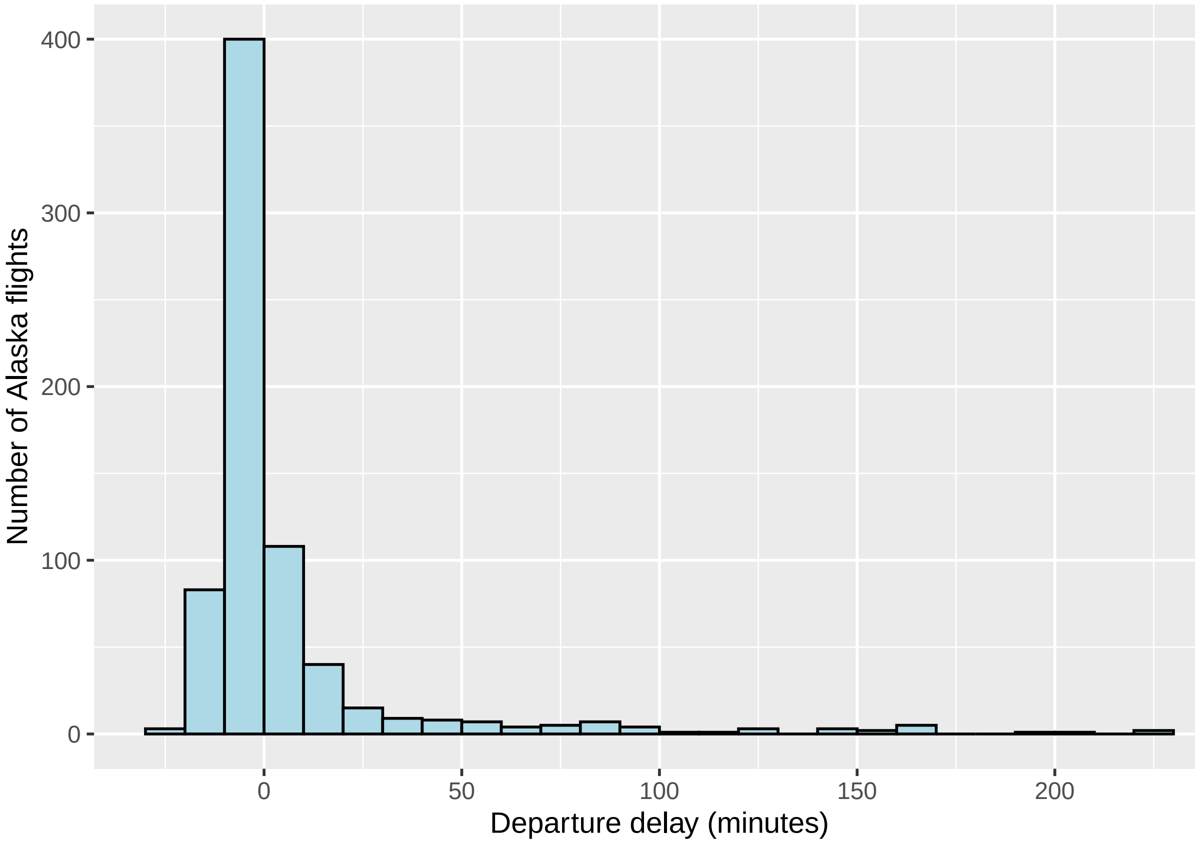
```
d.as <- subset(d, carrier == "AS" & origin == "EWR")
```

There 714 Alaska flights that departed Newark. If you're taking an Alaska flight out of Newark - perhaps to visit family out west - you might be interested in whether or not their flights typically depart on time. Let's take a look at the distribution of departure delays. The `dep_delay` variable is the departure delay in minutes. The value of `dep_delay` is positive when a flight was delayed, negative when the flight left early, and 0 when the flight departed on time.

Characterizing the distribution of a quantitative variable is more involved than for a qualitative variable. When we were describing flight departures by airport, we easily computed the proportion of all NYC departures by airport. That was an easy task because categories are discrete units. Quantitative variables, on the other hand, are more tricky because often the data are not discrete, and even when they are discrete, there are so many possible values that there will be few observations for each particular value. For example, there won't be many Southwest flights with a departure delay of exactly 2.45364 minutes.

One of the most common graphs used to illustrate the distribution of a single quantitative variable is a **histogram**, so let's start there:

```
ggplot(d.as, aes(x = dep_delay)) +
  geom_histogram(binwidth = 10,  # 10-minute bins
                 boundary = 0,
                 color = "black", fill = "lightblue") +
  labs(x = "Departure delay (minutes)",
       y = "Number of Alaska flights")
```



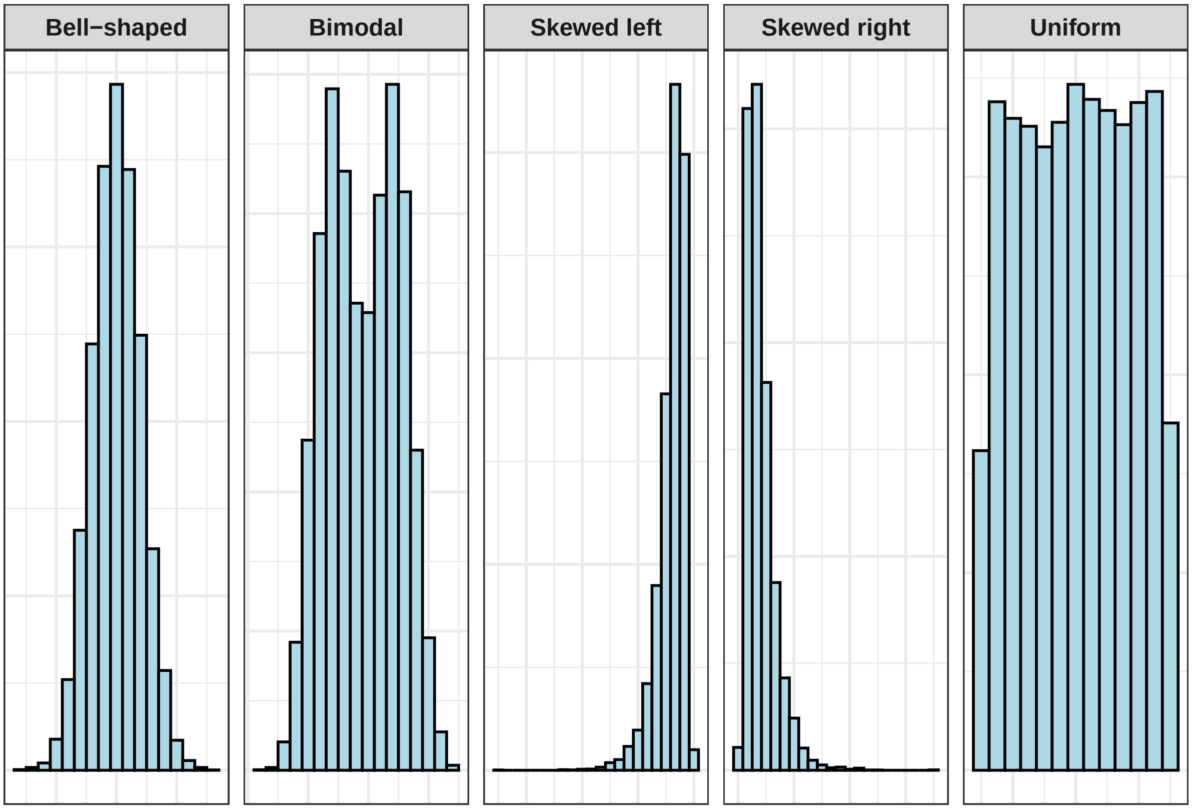
A histogram combines multiple observations into bins of a particular size and shows the frequency of those observations. In this histogram, I set the `binwidth` to 10 minutes. I also set the `boundary` to 0, which forces 0 to be a boundary between bins, allowing us to clearly differentiate between early- and late-departing flights. You don't need to set these values, but sometimes it helps when you want to make the bin size meaningful. Just beware that setting bin sizes too small or too large will make it impossible to see the shape of the distribution.

What can we learn from our histogram of departure delays among Alaska flights from Newark? Impressively, we see there were about 400 Alaska flights that departed 0-10 minutes early, the most common observations in the dataset (which is called the **mode**). We also see a significant number of flights (just over 1500) that departed 10-20 minutes early, or 0-10 minutes late. And then of course there are the occasional flights have long delays, sometimes an hour or more.

It's worth pointing out the shape of the histogram. The majority of observations of departure delay are clustered around zero, but there is a long *tail* of observations to the right, corresponding to longer delays. We can describe this kind of distribution as *skewed to the right*, or as one having *positive skew*. So although the vast majority of flights depart slightly early or slightly late, there's a small chance of departure delays that last for hours. In contrast, we never see flights that depart hours early, which of course makes sense given that people don't arrive many hours before a flight anticipating an early departure.

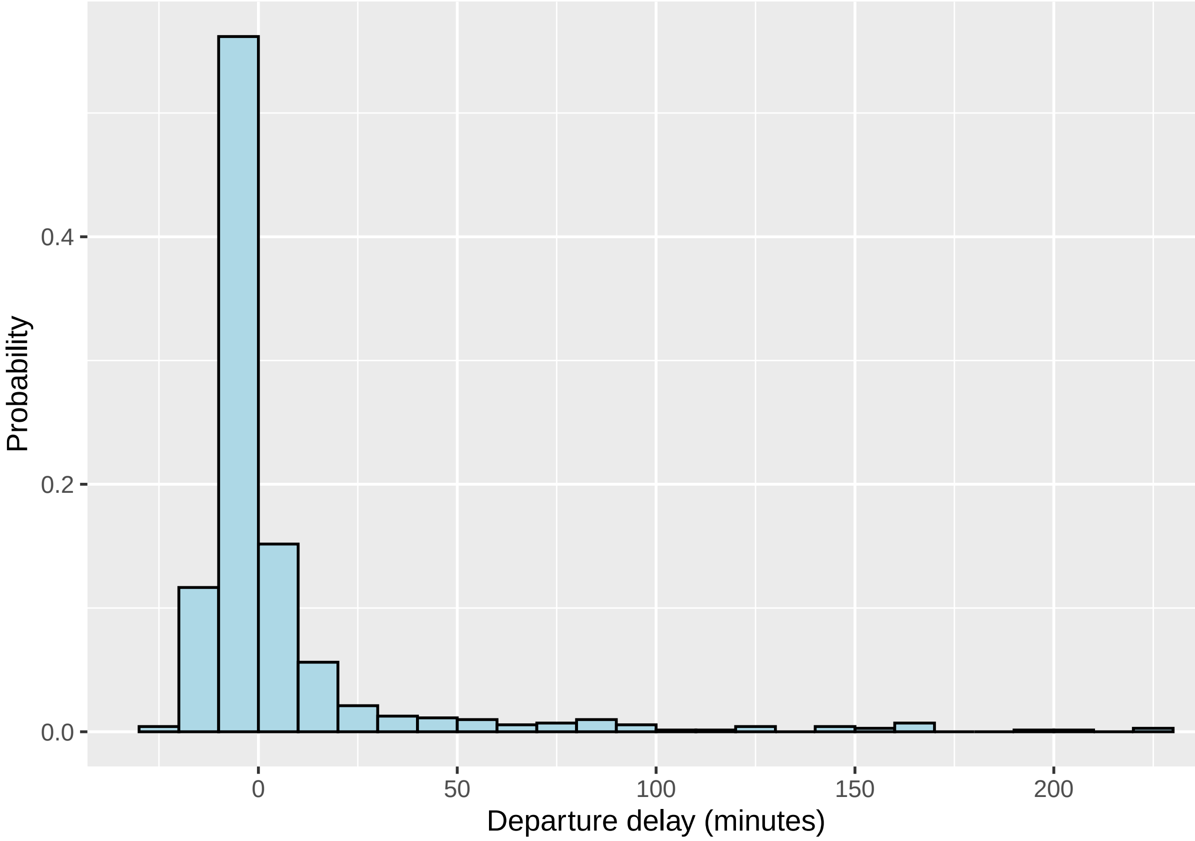
The shape of distributions of quantitative variables can vary substantially from variable to variable, or population to population. The figure below shows the most common shapes:

-*Bell-shaped* distributions have a single peak around the most typical values, with a similar number of of observations above and below the most typical values -*Bimodal* distributions have two peaks that can be distinguished with a gap of fewer observations in between. -*Skewed* distributions have a long tail of observations extending above (skewed to the right) or below (skewed to the left) the most typical values. -*Uniform* distributions have a similar nubmer of observations in each bin.



Like qualitative data, we can summarize quantitative data using the absolute count of observations as we've done, or we can display the relative frequency (probability) of each bin. To do this we use the same code as our initial histogram, but add a y-axis computing proportions for each bin with the `after_stat` function.

```
ggplot(d.as, aes(x = dep_delay,
                 y = after_stat(..count.. / sum(..count..)))) +
  geom_histogram(binwidth = 10,  # 10-minute bins
                 boundary = 0,
                 color = "black", fill = "lightblue") +
  labs(x = "Departure delay (minutes)",
       y = "Probability")
```



We don't learn anything new about the shape of the distribution, but now we can see, for example, that over half of Alaska flights depart on time or early, and just over half depart late. With so many observations, it was difficult to eyeball those proportions when plotting the absolute counts.

4.4.2.1 Numerical descriptions of central tendency

Distributions of quantitative variables contain a lot of information, and so it can be helpful to summarize the distributions numerically. One aspect of a distribution is its **central tendency**. In other words, what are the typical values of a variable? One numeric summary of central tendency is the **mean**:

$$\mu = \frac{\sum_{i=1}^{n} x_i}{n},$$

where μ is the mean of the variable, x_i is each observed value of the variable, and n is the total number of observations. The Greek letter sigma is mathematical notation for summation. The formula says to quantify the mean, you need to sum the value of each observation i in the population, then divide by the total number of observations. The i is simply a subscript representing each observation of the variable, from the first ($i = 1$) to the last ($i = n$). For example, if we have a dataset of $\{3, 5, 7, 9, 10\}$, the mean is quantified as

$$\mu = \frac{3+5+7+9+10}{5} = 6.8$$

In R, the mean can be quantified with the `mean` function:

```
(3+5+7+9+10)/5
```

```
## [1] 6.8
```

```
mean(c(3,5,7,9,10))
```

```
## [1] 6.8
```

Let's quantify the mean departure delay for Alaska flights. One thing to be aware of is that the departure delay variable has some missing observations, which are coded as `NA` in R. These are for flights that were cancelled. The `mean` function in R will not work unless you tell it how to handle the missing observations. We can simply remove those missing observations to quantify the mean by setting the `na.rm` argument to `TRUE`:

```
mean(d.as$dep_delay, na.rm=TRUE)
```

```
## [1] 5.804775
```

We see the mean departure delay for United flights was 5.8 minutes. Let's compare that to another common metric of central tendency: the **median**. The median is the middle value of a distribution. Consider again the made-up dataset of {3, 5, 7, 9, 10}. The value in the middle is 7, so that's the median. If there's an even number of observations in a dataset, the median can be computed mean of the two middle numbers. For example, if a dataset had observations of {3, 5, 7, 8, 9, 10}, the median is the average of 7 and 8: 7.5. Let's compute the median departure delay for United flights, again telling R to remove missing observations:

```
median(d.as$dep_delay, na.rm = TRUE)
```

```
## [1] -3
```

Fascinating. The median departure delay is -3 minute, or a departure time of three minutes early. How is it possible that these two different statistics paint different pictures of the typical departure delay? It turns out that the mean and median characterizd central tendency in different ways. Let's break this down. The mean is a ratio of two numbers: the sum of all observations divided by the number of observations. In our example, the mean is a ratio of the total number of minutes flights were delayed over the number of flights. Notice that the magnitude of every observation in the dataset can influence the mean. Indeed, if we change one of the values, the mean should change. Let's try that. There were 30 Alaska flights that departed exactly on time (`dep_delay` = 0). Let's quantify the mean as if one of them had left 500 minutes late.

```
#actual mean departure delay
mean.numerator <- sum(d.as$dep_delay, na.rm = T)
mean.denominator <- sum(!is.na(d.as$dep_delay))
mean.numerator/mean.denominator
```

```
## [1] 5.804775
```

```
#mean if one on-time flight left 500 minutes late
mean.numerator <- sum(d.as$dep_delay, na.rm = T) + 500
mean.denominator <- sum(!is.na(d.as$dep_delay))
mean.numerator/mean.denominator
```

```
## [1] 6.507022
```

Just by changing one departure delay value from 0 to 500, we see the mean departure delay changes from 5.8 to 6.5 minutes. The mean is sensitive to the magnitude of each observation, and thus it's particularly sensitive to extreme observations. Because the mean is sensitive to extreme observations, it's not always a useful indicator of typical values. Indeed, we can see in the histogram that it's much more likely for an Alaska flight to leave 0-10 minutes early than 0-10 minutes late.

Let's contrast with the median. The median is simply the value in the dataset where 50% of observations are greater and 50% of observations are lower. The magnitude of the values above or below the mean don't matter. Let's prove this with our example of a single departure delay changing from 0 to 500 minutes:

```
#actual median
median(d.as$dep_delay, na.rm=T)
```

```
## [1] -3
```

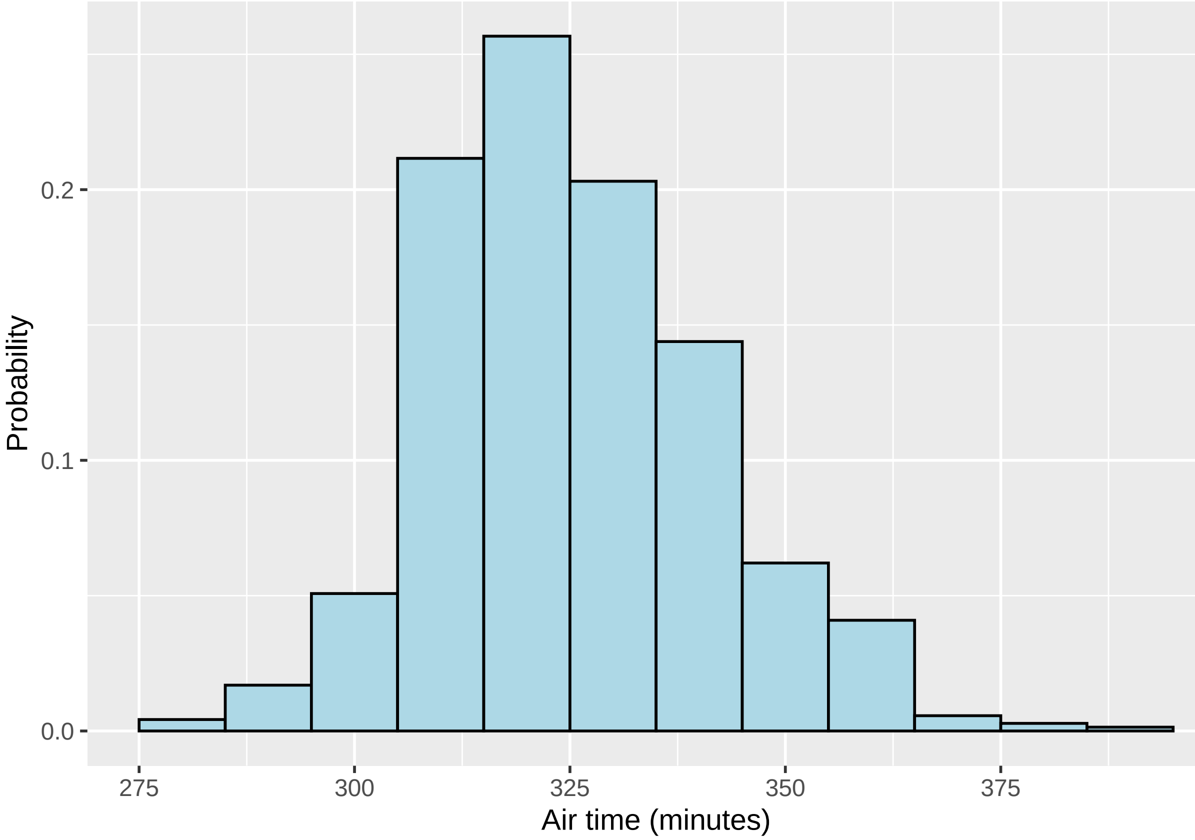
```
#median if one on-time flight left 500 minutes late
x <- d.as$dep_delay #create a vector x of the actual departure delays
x[which(x == 0)[1]] <- 500 #change the first on-time flight to 500 min late
median(x, na.rm=T)
```

```
## [1] -3
```

The median doesn't budge at all. We could have changed that late flight's departure delay to 1000, 5000, or an entire month late, and the median would stay the same. In this way, the median is robust to extreme observations, and so it's a handy statistic to characterize the central tendency of distributions that are strongly skewed (i.e., those with extreme observations), or those where we are interested in the most typical observations.

One last point about the mean and median. When the distribution of a quantitative variable is bell-shaped, the mean and median will be similar. For example, the distribution of air time for Alaska flights is approximately bell-shaped:

```
ggplot(d.as, aes(x = air_time,
                 y = after_stat(..count.. / sum(..count..)))) +
  geom_histogram(binwidth = 10,  # 10-minute bins
                 color = "black", fill = "lightblue") +
  labs(x = "Air time (minutes)",
       y = "Probability")
```



Because there are not extreme values with strong skew towards one end of the distribution, the mean and median will be similar.

```
mean(d.as$air_time, na.rm=T)
```

```
## [1] 325.6178
```

```
median(d.as$air_time, na.rm=T)
```

```
## [1] 324
```

Here the mean and median air time are only one minutes apart. Indeed, if the distribution was *perfectly* bell-shaped, the mean and median (and mode!) would all be identical. In contrast, when a variable has a skewed distribution, the mean and median diverge from each other, with the mean being drawn out in the direction of the skew.

4.4.2.2 Numerical descriptions of variation

Statistics of central tendency are useful for describing the most typical values of a quantitative variable, but sometimes we're interested in the variaation among observations. Two distributions can have the same central tendency, but different levels of variation. For example, datasets X = {4, 5, 5, 5, 6} and Y = {1, 2, 5, 8, 9} both have a mean and median of 5, but clearly the observations in Y are more variable than the observations in X. One way of quantifying variation among observations is the **variance**:

$$\sigma^2 = \frac{\sum_{i=1}^{n}(x_i - \mu)^2}{n}$$

Let's take a look at what the variance is doing. We take each observation x, subtract the mean from that value, and then square the term. This value is called a **squared deviation**, and the variance sums those squared deviations and then divides by the sample size. Effectively what's happening here is that the variance is quantifying the mean of the squared deviations from the mean. The more the observations are far away from the mean, the greater the squared deviations, the greater the mean. The variance with our made-up dataset, {3, 5, 7, 9, 10}:

$$\sigma^2 = \frac{(3-6.8)^2 + (5-6.8)^2 + (7-6.8)^2 + (9-6.8)^2 + (10-6.8)^2}{5} = 6.56$$

This way of quantifying the variance assumes the dataset includes all observations from the pouplation. In practice, most statistical software assumes you don't have all observations from the population, and in those cases, the variance is quantified with the denominator *n-1* instead of *n*. The quantity $n-1$ is a bias correction factor when quantifying the variance with a subset of observations from the population, and we'll explore this topic further when we introduce the concept of sampling in a later chapter.

For practical purposes, we can use the more typical calculation of variance with the denominator *n-1*, which is exactly what the `var` function applied in R. Let's go ahead and quantify the variance in air time among flights:

```r
var(d.as$air_time, na.rm=T)
```

```
## [1] 261.3608
```

We see the variance in arrival time is 251.4. Because we're squaring the deviation of each observation from the mean, the units of variance is the square of the original unit. Thus, the variance of air time is 261.4 min^2. Squared units can be difficult to comprehend, but we can put this metric back on the scale of the original units by taking the square root of the variance, which is called the **standard deviation**:

$$\sigma = \sqrt{\frac{\sum_{i=1}^{n}(x_i - \mu)^2}{n}} = \sqrt{\sigma^2}$$

The standard deviation can be quantified in R with the `sd` function. Like the variance, R's `sd` function uses the bias correction factor in the denominator.

```
sd(d.as$air_time, na.rm=T)
```

```
## [1] 16.16666
```

We see now that the standard deviation of air time is 16.2 *min*. The standard deviation is back in the original units of the variable, which is good, but it's still not obvious how we should interpret this number. For practical purposes, you can think of the standard deviation as the average difference in an observation from the mean. In other words, if you randomly picked values from the dataset, on average how far would those value be from the mean air time of 325.6 *min*? For Alasaka airline flights, the average difference in air time from the mean is 16.2 *min*.

Because standard deviations (and variances) are quantified based on the mean, they have the same weaknesses as the mean, namely they are sensitive to extreme observations. Consider, for example, the standard deviation in departure delay for Alaska flights:

```
sd(d.as$dep_delay, na.rm=T)
```

```
## [1] 31.36303
```

The meaning here is that we should expect an average difference of 31.4 min from the mean departure delay of 5.8 min. That suggests quit a bit of variation. Indeed, based on the mean and standard deviation alone, we shouldn't be surprised to see departure delays around 35-40 minutes late, or 30-35 minutes early. Yet if we look again at the histogram for departure delay, we see that Alaska flights rarely leave more than 35 mintues late or 30 minutes early. Indeed, most of the variation in departure delays is clustered between -20 and 20 minutes. The standard deviation here seems too high to describe the typical variation in the dataset, and that's because the standard deviation is being increased by the rare but extreme cases of very long delays.

A more flexible way of describing variation in a quantitative variable is to use **percentiles**. A percentile is the observation in the dataset for which a specified percentage of the data falls below that value. We've actually already seen a percentile. The median is the 50th percentile because 50% of the observations fall below that value. But you can quantify a percentile for any level of percentage. In R, percentiles are quantified with the `quantile` function . Here are some percentiles for the departure delay data:

```
## Report just the 50th percentile (i.e., the median)
quantile(d.as$dep_delay, probs = 0.5, na.rm=TRUE)
```

```
## 50%
##  -3
```

```
## Report the 0th, 25th, 50th, 75th, and 100th percentiles:
quantile(d.as$dep_delay, probs=c(0, 0.25, 0.5, 0.75, 1), na.rm=TRUE)
```

```
##   0%  25%  50%  75% 100%
##  -21   -7   -3    3  225
```

So we can see that there are no observations below -21 *min*, 25% of the observations are below -7 *min*, 50% of the observations are below -3 *min*, and so on. Note that the 0th and 100th percentiles are the **minimum** and **maximum** observations in the data, respectively. The percentiles nicely describe variation despite the strong skew in the dataset. For example, we can see from the percentiles that the bulk of the observations - 75% in fact - are between -21 and 3. So if you have a departing flight with Alaska Airlines, you can interpret this as meaning there's a 75% chance that your flight will depart anytime from 21 minutes early to 3 minutes late. The remaining 25% of observations are between 3 and 225 minute delays.

The percentiles are sometimes used to quantify numerical indices of variation. Two common metrics are the **range** and the **interquartile range (IQR)**. The range is simply the difference between the maximum and minimum observation, whereas the IQR is the difference between the 75th and 25th percentiles. Here are some relevant R functions that you'll find of use:

```
## report the minimum and maximum
range(d.as$dep_delay, na.rm=TRUE)
```

```
## [1] -21 225
```

```
## report the range as max - min
max(d.as$dep_delay, na.rm=TRUE) - min(d.as$dep_delay, na.rm=TRUE)
```

```
## [1] 246
## IQR
IQR(d.as$dep_delay, na.rm=TRUE)
```

```
## [1] 10
```

```
## IQR calculated with the quantile function
q.75 <- quantile(d.as$dep_delay, na.rm=TRUE, probs=0.75)
q.25 <- quantile(d.as$dep_delay, na.rm=TRUE, probs=0.25)
as.numeric(q.75 - q.25)
```

```
## [1] 10
```

The range is commonly reported for variables to define the bounds of the observed data, but note that the range will be strongly sensitive to extreme observations. For distributions with extreme observations or strong skew, the IQR can be a useful summary statistic to describe variation. Both the range and the IQR have the same units as the original variable.

4.4.2.3 Summary function

The summary function is handy for quickly evaluating the quantiles and mean of a numeric variable. You can apply it to a single vector, or to an entire data frame to summarize all the objects in the data frame at once. If there are characters in the data frame, the summary function will simply indicate that the object is a character of a specific length.

```
## Summary for a single variable
summary(d.as$dep_delay)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
## -21.000  -7.000  -3.000   5.805   3.000 225.000       2
```

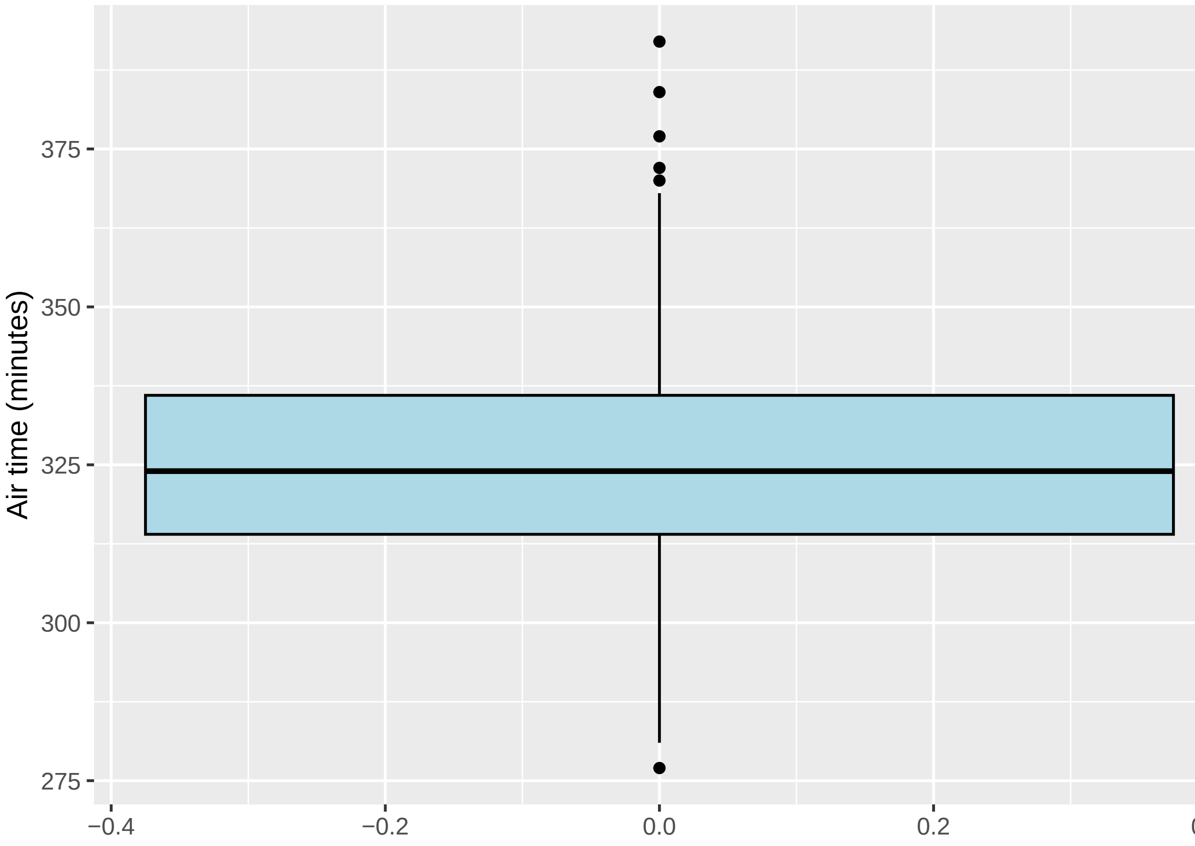
```
## Summary for the entire data frame
summary(d.as)
```

```
##       year          month             day           dep_time
##  Min.   :2013   Min.   : 1.000   Min.   : 1.00   Min.   : 651.0
##  1st Qu.:2013   1st Qu.: 3.000   1st Qu.: 8.00   1st Qu.: 717.8
##  Median :2013   Median : 6.000   Median :16.00   Median :1805.0
##  Mean   :2013   Mean   : 6.415   Mean   :15.79   Mean   :1294.6
##  3rd Qu.:2013   3rd Qu.: 9.000   3rd Qu.:23.00   3rd Qu.:1825.0
##  Max.   :2013   Max.   :12.000   Max.   :31.00   Max.   :2205.0
##                                                   NA's   :2
##  sched_dep_time   dep_delay         arr_time    sched_arr_time
##  Min.   : 705   Min.   :-21.000   Min.   :   3   Min.   :1015
##  1st Qu.: 720   1st Qu.: -7.000   1st Qu.:1003   1st Qu.:1025
##  Median :1815   Median : -3.000   Median :2043   Median :2125
##  Mean   :1285   Mean   :  5.805   Mean   :1565   Mean   :1595
##  3rd Qu.:1825   3rd Qu.:  3.000   3rd Qu.:2128   3rd Qu.:2145
##  Max.   :1835   Max.   :225.000   Max.   :2355   Max.   :2158
##                 NA's   :2         NA's   :2
##    arr_delay        carrier              flight         tailnum
##  Min.   :-74.000   Length:714         Min.   :  5.00   Length:714
##  1st Qu.:-32.000   Class :character   1st Qu.:  7.00   Class :character
##  Median :-17.000   Mode  :character   Median :  7.00   Mode  :character
##  Mean   : -9.931                      Mean   : 12.19
##  3rd Qu.:  2.000                      3rd Qu.: 15.00
##  Max.   :198.000                      Max.   :915.00
##  NA's   :5
##     origin             dest              air_time        distance
##  Length:714         Length:714         Min.   :277.0   Min.   :2402
##  Class :character   Class :character   1st Qu.:314.0   1st Qu.:2402
##  Mode  :character   Mode  :character   Median :324.0   Median :2402
##                                        Mean   :325.6   Mean   :2402
##                                        3rd Qu.:336.0   3rd Qu.:2402
##                                        Max.   :392.0   Max.   :2402
##                                        NA's   :5
##       hour           minute        time_hour
##  Min.   : 7.00   Min.   : 5.00   Min.   :2013-01-01 07:00:00
##  1st Qu.: 7.00   1st Qu.:20.00   1st Qu.:2013-03-31 09:45:00
##  Median :18.00   Median :25.00   Median :2013-06-28 12:30:00
##  Mean   :12.62   Mean   :22.18   Mean   :2013-06-29 05:05:27
##  3rd Qu.:18.00   3rd Qu.:25.00   3rd Qu.:2013-09-25 15:15:00
##  Max.   :18.00   Max.   :35.00   Max.   :2013-12-31 18:00:00
##
```

4.4.2.4 Other graphical approaches to display quantitative variables

I want to introduce two additional types of graphs that are useful for displaying the distributions of quantitative variables, specifically a **box plot** and **stripchart**. The graph below shows the distribution of air time using both plots. Box plots show the distribution of quantitative variables by displaying numerical summaries.

```
ggplot(d.as, aes(y = air_time)) +
  geom_boxplot(color = "black", fill = "lightblue") +
  labs(x = NULL,
       y = "Air time (minutes)")
```



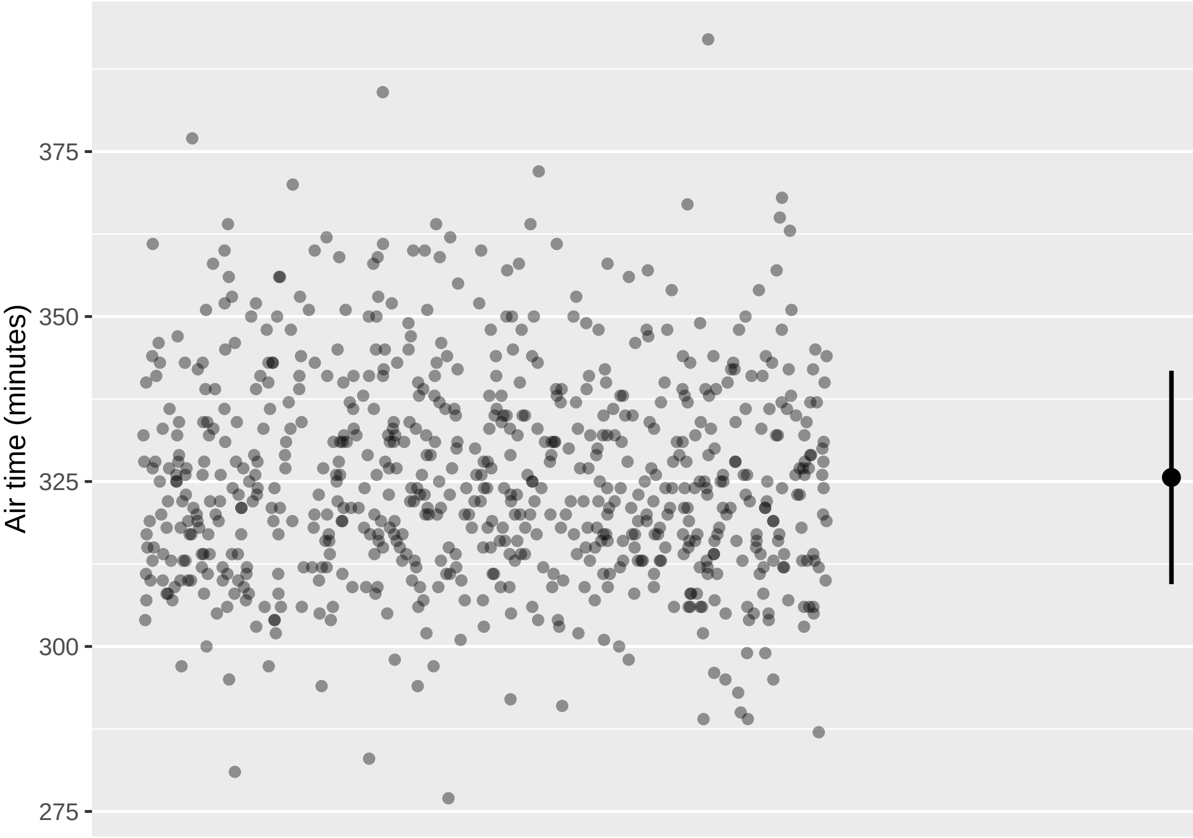
- Median: represented by the thick horizontal line as a metric of central tendency.
- IQR: represented by the edges of the box, where the lower edge is the 25th percentile and the upper edge is the 75th percentile. The box displays the bounds of the middle 50% of the data, giving you a sense for variation.
- Variation in the tails: Values in the tails of the distribution are displayed by the whiskers, which extend up to 1.5 times the IQR from the box. Any values outside the whiskers are defined as **outliers** and are displayed by points.

The shape of a distribution can be inferred from a box plot based on the symmetry of the box and whiskers. If the box and whiskers are largely symmetrical

- as is the case for air time - then the distribution has a bell shape. If the box and whisker extends far from the median to one end of the distribution, then the distribution is skewed.

A stripchart shows each individual observation as a cluster of points along the x- or y-axis (I've chosen the y-axis in this case). It's a nice option if you want to display all the data while still getting a sense for the shape of the distribution. In this case we can see clustering of the arrival times around 325 minutes with fewer observations about equally represented above and below the mean, suggesting a bell shape. Stripcharts can be supplemented with numerical descriptions as well. In this case, the circle with error bars around it represents the mean and one standard deviation above and below the mean.

```r
ggplot() +
  geom_jitter(
    data = d.as,
    aes(x = 0, y = air_time),
    width = 0.01, height = 0, alpha = 0.4
  ) +
  stat_summary(
    data = d.as,
    aes(x = 0.02, y = air_time),
    fun.data = function(y) {
      m <- mean(y, na.rm = TRUE)
      s <- sd(y, na.rm = TRUE)
      data.frame(y = m, ymin = m - s, ymax = m + s)
    },
    geom = "linerange",
    linewidth = 0.8
  ) +
  stat_summary(
    data = d.as,
    aes(x = 0.02
        , y = air_time),
    fun = function(y) mean(y, na.rm = TRUE),
    geom = "point",
    shape = 16,
    size = 3
  ) +
  scale_x_continuous(NULL, breaks = NULL) +
  labs(y = "Air time (minutes)")
```



4.5 Describing relationships between variables

Whether your goal is causal explanation or prediction, much of science involves examining relationships between variables. So far we have looked at different ways of describing a single variable. But perhaps the distribution is associated with another variable. Those associations may or may not be causal, and teasing out causal vs. non-causal associations will be a major topic later in the book. At this point we simply need some basic approaches to describe associations between variables.

4.5.1 Associations between quantitative variables

Does learning something about the departure delay of a fligth tell us anything about the arrival delay? The first step to examine the association between two quantitative variables is to create a **scatterplot**, where each point in the graph displays the values of the two variables for each individual in the dataset.

```
ggplot(d.as, aes(x = dep_delay, y = arr_delay)) +
  geom_point(alpha = 0.4) +
  labs(x = "Departure delay (minutes)",
       y = "Arrival delay (minutes)")
```

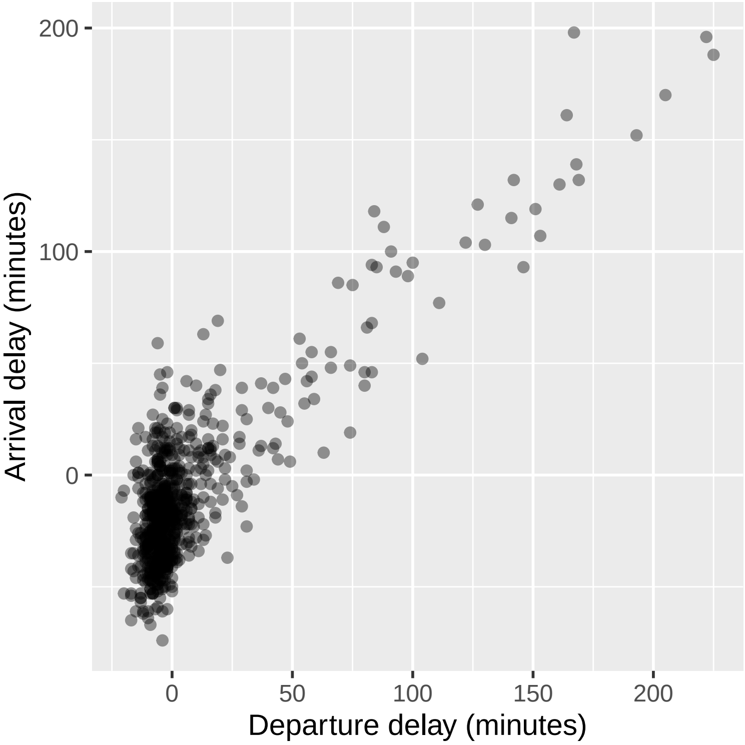


Figure 4.1: TODO: caption

The scatterplot shows a clear and expected pattern. When a flight has a long departure delay, it tends to have a long arrival delay. In other words, we see a positive relationship between the two variables. Scatterplots can also reveal a negative relationships, where the value of one variable tends to decrease as the value of the other variable increases.

The figure below shows a range of possible patterns that one might encounter in scatterplots. Note that in addition to describing the nature of the relationship between variables, we can also describe the magnitude of the relationship. For example, the graph shows two positive relationships in the top row, but the pattern looks much stronger in the scatterplot on the left than the scatterplot on the right.

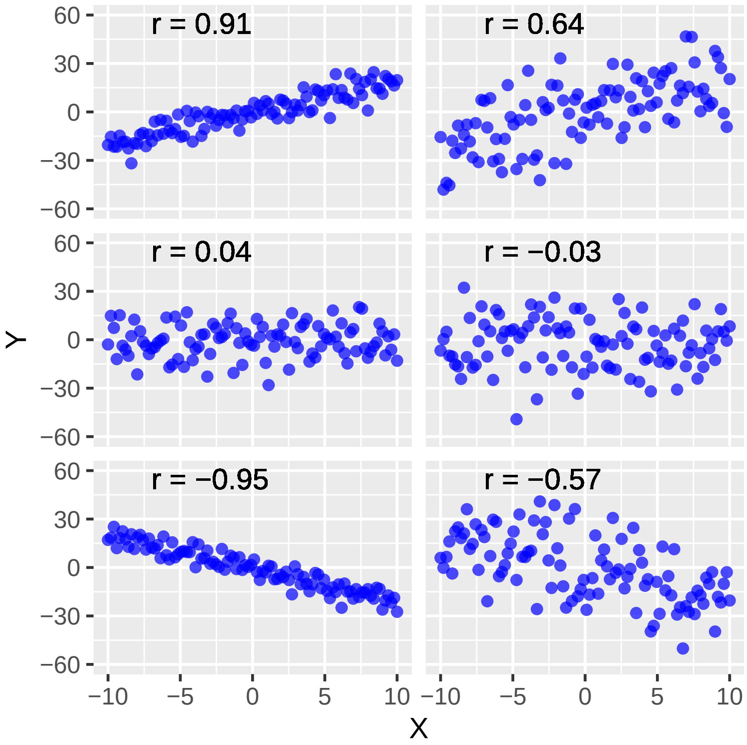


Figure 4.2: TODO: caption

The **correlation coefficient (r)** is a numerical metric used to describe both the direction and strength of an association. It has a bit of a messy formula:

$$r = \frac{\frac{1}{n-1}\sum_{i=1}^{n}(x_i - \mu_x)(y_i - \mu_y)}{\sigma_x \sigma_y}$$

At a basic level, the correlation coefficient is based on the **covariance**, which is the numerator in the formula for the correlation coefficient. The covariance is a cross product of the deviation between each value of variables x and y from their mean. When x and y values have a common relationship to their mean, the covariance is positive. This would reflect a positive association, where high values of x correspond to high values of y, and vice versa. When x and y values have opposite relationship to their mean, the covariance is negative. This would reflect a negative association, where high values of x correspond to low values of y, and vice versa. When there's no association between x

and y, the covariance is 0. Covariance has awkward units. The correlation coefficient divides the covariance by the standard deviations of each variable, which standardizes the association to be between -1 and 1. The values $r < 0$ indicate negative associations, $r > 0$ indicate positive values, and $r = 0$ reflects no association. The strength of the association is reflected by how close the value of r is to 1 or -1. Stronger associations have values of r close to 1 or -1. You can see this in the example scatterplots, where stronger associations reflect less scatter among the points. When r is exactly 1 or -1, the points all fall on a perfect line.

Does it matter which variable we place on each axis when examining the association between two quantitative variables? It can. If your goal is causal explanation, the general practice is to place the explanatory variable on the x-axis and the response variable on the y-axis. The departure delay is almost certainly a cause the the arrival delay, so it makes sense that we placed departure delay on the x-axis. But don't assume that an association between two variables is causal! When we describe associations between variables, all we are doing is describing patterns. We'll need additional tools to say whether or not the patterns we find in the data reflect some kind of causal process. More on that in later chapters.

4.5.2 Associations between quantitative and qualitative variables

Let's look at how to compare quantitative data between categories of a qualitative variable. The idea here is to visually inspect how the distribution of the quantitative variable differs between categories of the qualitative variable. As an example, let's compare the distance of flights departing from each of the three airports in New York.

```r
ggplot(d, aes(x = distance)) +
  geom_histogram(binwidth = 300, boundary = 0,
                 color = "black", fill = "lightblue") +
  facet_wrap(vars(origin), ncol=1) +
  labs(x = "Flight distance (miles)",
       y = "Number of flights")
```

What can we learn from this figure. First, we can see that the shape of the distribution of flight distance varies among airports. LGA has a largely bell-shaped distribution with no skew, whereas the EWR and JFK distributions are more bimodal (perhaps trimodal for JFK). The peak of flight distances between 2000-3000 miles for EWR and JFK suggests those airports specialize more in long-haul flights in comparison to LGA. Indeed, the most common flight distance out of JFK is about 2500 miles.

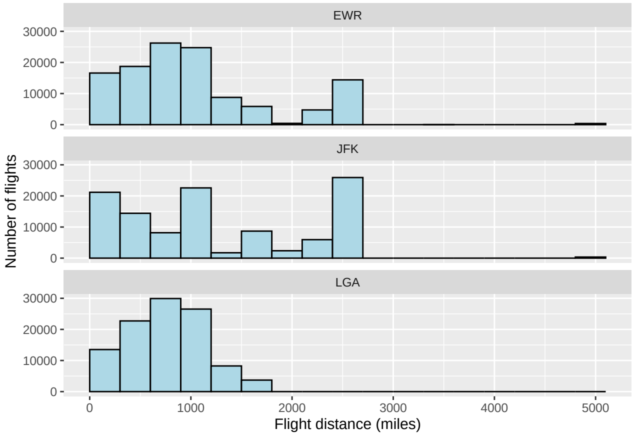


Figure 4.3: TODO: caption

Although in this case I've plotted a stacked panel of histograms to compare flight distance among categories of airport, we could have used other types of figures. Here's the same association displayed as a set of boxplots. The box plots show the same general shape of the distributions, but they additionally show some useful quantitative summaries. For example, we can see the median distance is greater for JFK than EWR and LGA. It's also clear from this plot that variation in distance among flights is greatest for JFK and lowest for LGA.

4.5.3 Associations between qualitative variables

What about associations between two categorical variables? One type of plot for this purpose is a **mosaic plot**, which shows the relative frequencies of each category for one variable by the categories of the other variable. For example, the plot below shows the percentages of each carrier with departing flights from each airport. To simplify the plot, I've combined small carriers (defined as those with <1000 departing flights) into an "other" category.

It's quite apparent that the distribution of carriers differs quite a bit among airports. The top carrier at EWR is United Airlines (UA), but ExpressJet (EV) wasn't close behing. ExpressJet was an airline that flew contracted flights for other carriers. At JFK we see JetBlue (B6) is clearly the most common carrier. Delta is the most common carrier departing LGA, but it appears there's a much more even distribution of carriers operating at LGA compared to EWR and JFK.

```
#collapse small carriers (<1000 flights)
n_carrier <- table(d$carrier)
small_carriers <- names(n_carrier[n_carrier < 1000])
```

```
# collapsed carrier variable
d$carrier2 <- ifelse(d$carrier %in% small_carriers,
                     "Other", as.character(d$carrier))

# plot
ggplot(d, aes(x = origin, fill = carrier2)) +
  geom_bar(position = "fill", color = "black") +
  scale_fill_brewer(palette = "Set3") +
  scale_y_continuous(labels = function(x) paste0(100 * x, "%")) +
  labs(x = "Origin", y = "Proportion of flights", fill = "Carrier")
```

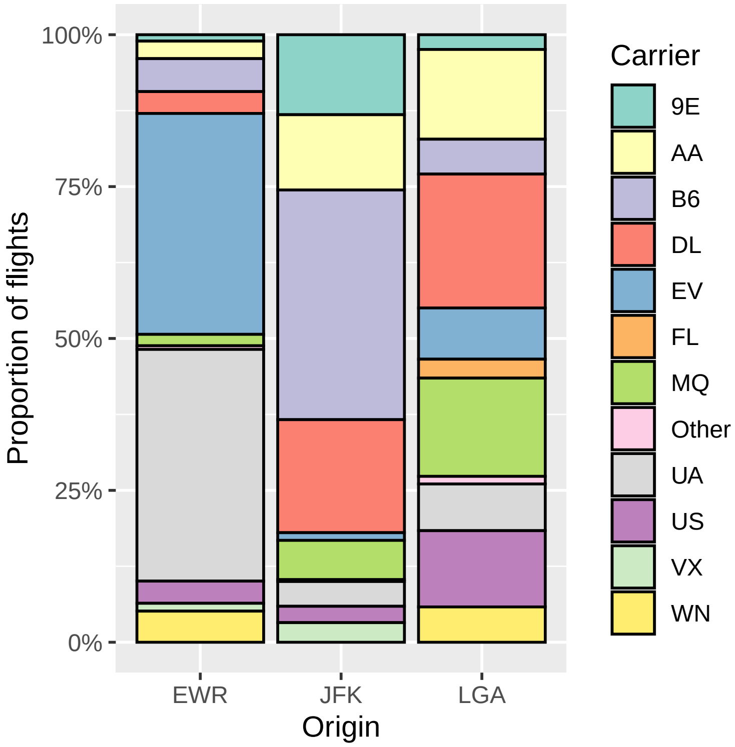


Figure 4.4: TODO: caption

4.6 Next steps

In this chapter we've barely scratched the surface of ways to describe and display data. The approaches in this chapter are common, but certainly not the only options, and we will encounter new types of statistics figures in the coming chapters.

Chapter 5

Uncertainty from sampling

The last two chapters introduced the basic structure of datasets and approaches to describe and display data. We explored a dataset where all observations were made from the population of interest, namely all flights out of NYC airports in 2013. Using a complete dataset like this allowed us to ignore the **problem of uncertainty**. If all observations from the population of interest enter the dataset (and are measured accurately), then any summary statistic quantified (e.g., mean, standard deviation, proportion) might be considered the truth. In practice, even "complete" datasets that include all observations from a population can still have sources of uncertainty, such as measurement or processing errors, but we'll set those issues aside for now.

If only life was so simple! The truth is that in the vast majority of scientific studies, it isn't feasible to measure every individual in a population of interest. What proportion of people in the United States had Lyme Disease last year? What fraction of forests are occupied by endangered spotted owls? What is the mean cholesterol level of people diagnosed with Type 2 diabetes? What's the average radon concentration in households in Chicago? By how much does a vaccine reduce the probability of getting a viral infection? Good luck measuring all the individuals in each of the populations of interest represented by these questions.

What happens when we can't measure every individual in the population? Consider the question about radon concentration in Chicago. If I wanted to know the true mean radon concentration, I would need to measure the air in *every* household in Chicago. There are over a million households in Chicago, and it simply isn't feasible to measure them all. So what do we do? We measure radon in a *subset* of the households. Suppose we select 1000 households for measurement, and we find the mean radon level is 3.0 pCi/L. Is that the true mean radon concentration? Probably not! It's very likely that the mean radon level in our sample will differ from the truth *just by chance*. In other words, measurements from subsets of populations are *estimates*, and there is *uncertainty* about how good those estimates are relative to the truth.

Uncertainty is the fundamental reason why we need a discipline called *statistics*. If we could just quantify values of interest with certainty, then the analysis of data would ultimately be an exercise in mathematics. Measure a bunch of individuals, quantify the value of interest with a mathematical formula, and voila, you have your answer! It just isn't that easy.

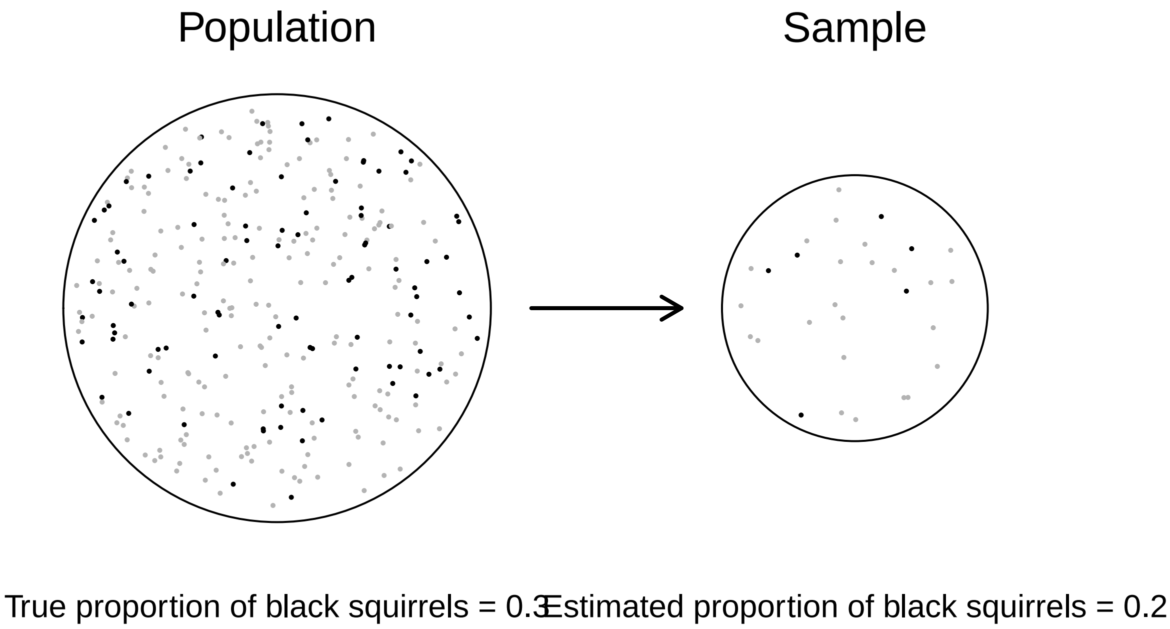
If we're going to do science the right way, we have to wrestle with the problem of uncertainty. What creates uncertainty in our statistics? How do we quantify the magnitude of uncertainty? How do we design studies to limit uncertainty? How do we make decisions in light of uncertainty? Fundamentally these are the sorts of questions that are at the heart of statistical analysis. In this chapter, we will look at how uncertainty arises and how we can control it. Then the next chapter will focus on how we can quantify uncertainty with probability.

5.1 Sampling requires estimation

I'm an evolutionary biologist interested in how environmental change causes evolution in wildlife populations. One of our projects focuses on how urbanization and climate change affect the evolution of coat color in eastern gray squirrels (*Sciurus carolinensis*). Coat color is typically gray or black, which is determined by a single allele at a gene called *Mc1R* (melanocortin-1 receptor). The *Mc1R* gene controls how pigment is deposited in tissue in vertebrates. One of the fundamental measurements we make is how frequent each color morph is in a population, and how that frequency changes over time.

We have both color morphs of gray squirrels in my own backyard in Rochester, NY, so let's start there. *What proportion of eastern gray squirrels in my backyard are black*? Notice my question is very specific about *who* I want to make conclusions about. What proportion of the gray squirrels in *my backyard* are black? The target population is very specific and small. My backyard is pretty small, and let's suppose there are only 10 squirrels that live there. I spot each of them and count six black morphs and four gray morphs. Thus, the proportion of black morphs is 0.6, the proportion of gray morphs is 0.4, and that's that! Life is simple when our target population is very specific, small, and easily measured.

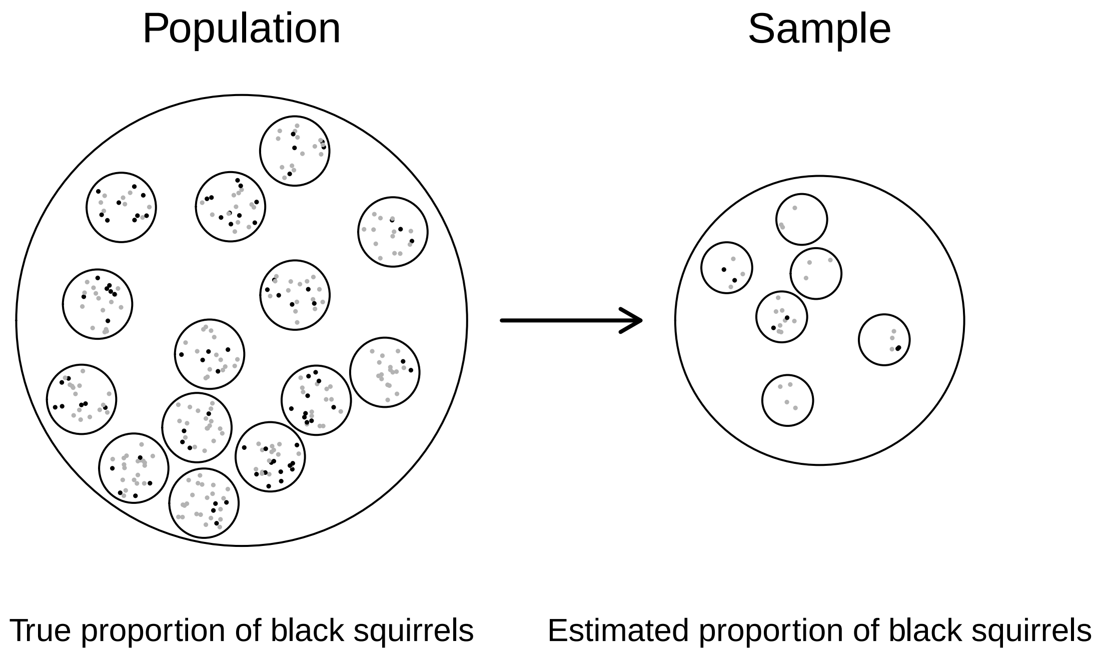
But let's expand now. What if I lived on a property with 20 acres of woods (if only I were so lucky), and let's say there are 300 eastern gray squirrels in my woods. Measuring the proportion of black squirrels isn't so simple now. I can't track down all 300 squirrels, but I could measure the coat color in a subset of squirrels. The subset of individuals measured from the population of interest is called the **sample**. Again, for the sake of argument, let's assume that of the 300 squirrels in my woods, 210 are gray and 90 are black, meaning the true proportion of black squirrels is $\frac{90}{300} = 0.3$. Now I don't know this in reality, which is the entire problem we face here. I can't find all 300 squirrels, but I can *estimate* the proportion black with a sample of them. Suppose I take a sample of 30 squirrels and find 6 are black. Based on my sample, I quantify the proportion of black squirrels ss $\frac{6}{30} = 0.2$.


Population
Sample
True proportion of black squirrels = 0.3
Estimated proportion of black squirrels = 0.2

```
## Population black count:90of300 (0.3)
```

```
## Sample black count:6of30 (0.2)
```

In this example, note that the sampling process is about individual squirrels. But sometimes sampling can be hierarchical. For example, let's rephrase the research question and ask *What proportion of squirrels in all backyards in Rochester, NY are black*? There are too many backyards in Rochester to visit them all, so I will need to select a sample of backyards. Moreover, I can't guarantee that I will observe every single squirrel in every back yard, so there's additional sampling going on within each backyard. In other words, the population is nested. Of the population of all backyards in Rochester, I can only visit a subset of them. And among the backyards I visit, I can only measure a subset of squirrels. There are two, nested layers of sampling here that can cause deviations between the true proportion of black squirrels in Rochester backyards and the estimate from my sampling.


Population
Sample
True proportion of black squirrels
Estimated proportion of black squirrels

```
## Population black count: 90 of 300
```

```
## Sample black count: 6 of 30
```

Sampling brings us face to face with the fundamental problem of statistical analysis. The estimates we make of quantities of interest from a sample will not be the same as the truth. This uncertainty can make decision-making very difficult. Suppose I'm interested in whether natural selection is causing an increase or decrease in the proportion of black squirrels in a population of interest. I measure the proportion every year for 20 years. How much of the change that I see is due to natural selection (if any), and how much is due to the fact that I'm sampling a subset of the pouplation? In other words, how much of the change from year-to-year is a signal of the process of interest (selection), and how much of it is noise (errors from sampling)? The more noise there is, the harder it is to see the signal. So if we're going to do science well, we need tools to minimize the amount of noise from sampling and to quantify the degree of uncertainty about an estimate from any given sample.

5.2 Parameters, estimates, estimands

Like most technical disciplines, statistics is full of jargon. Sorry. Before we take a look at the causes of uncertainty and how we can control them, let's define some key terms related to sampling. Consider again my goal of estimating the proportion of black squirrels in my woods.

Quantities with unknown values are called **parameters**. In my example, the parameter is the true proportion of black squirrels in my woods. The parameter is a probability in this case, but parameters can be means, variances, rates, components of a complex statistical model, or any number of other quantities. The defining feature of parameters is that they are unknown and require estimation to answer your research question. If I collect data on 30 squirrels and find 6 are black, then my **estimate** is 0.2 for the proportion black. In practice, we can distinguish estimates from parameters by using a "hat". For example, the true proportion of black squirrels is $p = 0.3$, whereas the estimate from my sample is $\widehat{p} = 0.2$.

Sometimes we have to estimate multiple parameters in a statistical analysis, but not all parameters are of direct interest for our research question. The quantities of interest for our research question are called **estimands**. In my example on the proportion of black squirrels in my woods, the question and analysis is simple enough that there's only a single parameter, and it's the parameter I'm interested in. The true proportion of black squirrels is the estimand.

In a more complex statistical analyses, we will often have to estimate multiple parameters, but not every parameter is of interest. For example, when I want to measure the change in the proportion of black squirrels over time, I could fit a model that includes two parameters, specifically the proportion of black squirrels at the starting time point, and the annual change in the proportion of black squirrels. If I'm interested in whether natural selection is causing a change in the proportion fo black squirrels, the estimand is the annual change. In these cases of complex statistical analyses, it is important to identify which parameters estimands.

5.3 Sources of uncertainty from sampling

Now that we have a basic sense for the process of sampling, let's get more specific about why estimates from samples deviate from the truth. There are two fundamental processes that cause estimates from a sample to be different from the truth:

1. **Sampling error**: When I take a sample of 30 squirrels out of 300, by chance there can be relatively more or fewer black squirrels in my sample than in the true population. This kind of variability is a consequence of selecting a finite number of individuals into a sample. There's simply randomness in who happens to be selected. The deviation in a sample estimate from the truth due to this chance variation of who enters the sample is called sampling error.

2. **Systematic error (bias):** How did the 30 squirrels I observed enter my sample? If each of the 300 squirrels in my woods had an equal chance of entering the sample, then we have no problem. But let's say the sampling process was not random. What if black squirrels are easier to spot than gray ones, such that each black morph has a greater chance of entering my sample than the gray ones? My estimate of the proportion black would be greater than the truth. The tendency for the sampling procedure to produce estimates that are too high or too low on average is called systematic error, or bias.

Sampling error and bias are the two basic causes of uncertainty when estimating quantities from samples, but it's important to note that they are not the only sources of uncertainty in practice. For example, **measurement error** is a source of uncertainty that can be present whether we use a sample or exhaustively measure every individual in a population. **Measurement error** is a deviation in the observed value for a particular individual in the sample compared to the truth. Those deviations may be random. For example, a scale used to measure mass could produce measurements that are a few milligrams too high or low. If the deviations are equally likely to be high or low (zero on average), then, then the measurement error will add variability to the estimate, but not bias it. On the other hand, if the scale consistently overestimates or underestimates mass, then measurement error will introduce bias.

The takehome point here is that when we rely on samples, we can expect uncertainty in our estimates from random sampling error and biases with respect to how individuals enter the sample. Additional uncertainty can come from measurement error, data entry errors, misclassification, and more. These latter sources of error are important but often more specific to the data collection methods of particular disciplines, so I acknowledge them here but will keep the focus of this chapter on sampling error and bias associated with samples.

5.4 Accuracy and precision

When we estimate quantities from a sample, there's uncertainty about the value of that estimate. We will explore two different ways of characterizing the quality of an estimate: how close our estimate should be to the truth based on our sampling design, and the variation in the possible values of the estimate we might obtain based on our sampling design. These two concepts are referred to as **accuracy** and **precision**, and estimates are of highest quality when they are both accurate and precise. Let's break these terms down.

5.4.1 Accuracy

On average, how close is an estimate from the truth? Let's reconsider our squirrel sampling process. There are 300 squirrels in my woods, and 90 are black, so the true proportion of black squirrels is 0.3 The sampling process consists of observing a subset of 30 squirrels and recording whether or not each is black. In this case, the parameter (and estimand) is the true proportion of black squirrels, and the estimate is the proportion of black squirrels in the sample. If we record 6 black squirrels out of 30 our estimate for the prportion black is 0.2. On its face, that particular estimate doesn't appear to be very accurate. The estimate is obviously too low. So imagine we resample the population, again drawing a sample of 30 and this time finding that 11 are black. In this second sample, the proportion black is $\frac{11}{30} = 0.37$. Now the estimate is too high! So again we resample the population, drawing another 30 squirrels, eight being black, so the proportion black is too low: $\frac{8}{30} = 0.27$.

Are these estimates accurate? In the strict sense that the estimates are not identical to the truth, you might be tempted to say they are not accurate. But that's not how we define accuracy in a statistical sense. To describe whether a sample estimate is accurate, we have to consider the entire distribution of possible estimates from a sample of 30 squirrels. In other words, imagine that I repeated this process of sampling 30 squirrels 1000 times, each time estimating the proportion black. The figure below shows a histogram of the estimates: (5.1.

Notice that this distribution is centered right on 0.3 (dotted vertical line), which is exactly what the truth is. In other words, when considering our sampling process, repeatedly conducted many times, we get a sense for the entire distribution of possible estimates, and we see that our estimate is - **on average** - accurate. Any particular estimate can be low or high. The point here is that there is no *systematic* difference between the estimate and the truth, and in that sense, we would conclude the sampling process is accurate. This distribution of possible outcomes of an estimate based on our sampling process is called a **sampling distribution**. If the sampling distribution is centered on the true value of a parameter, then the estimates are by definition accurate.

Of course estimates from samples are not always accurate. Remember that black

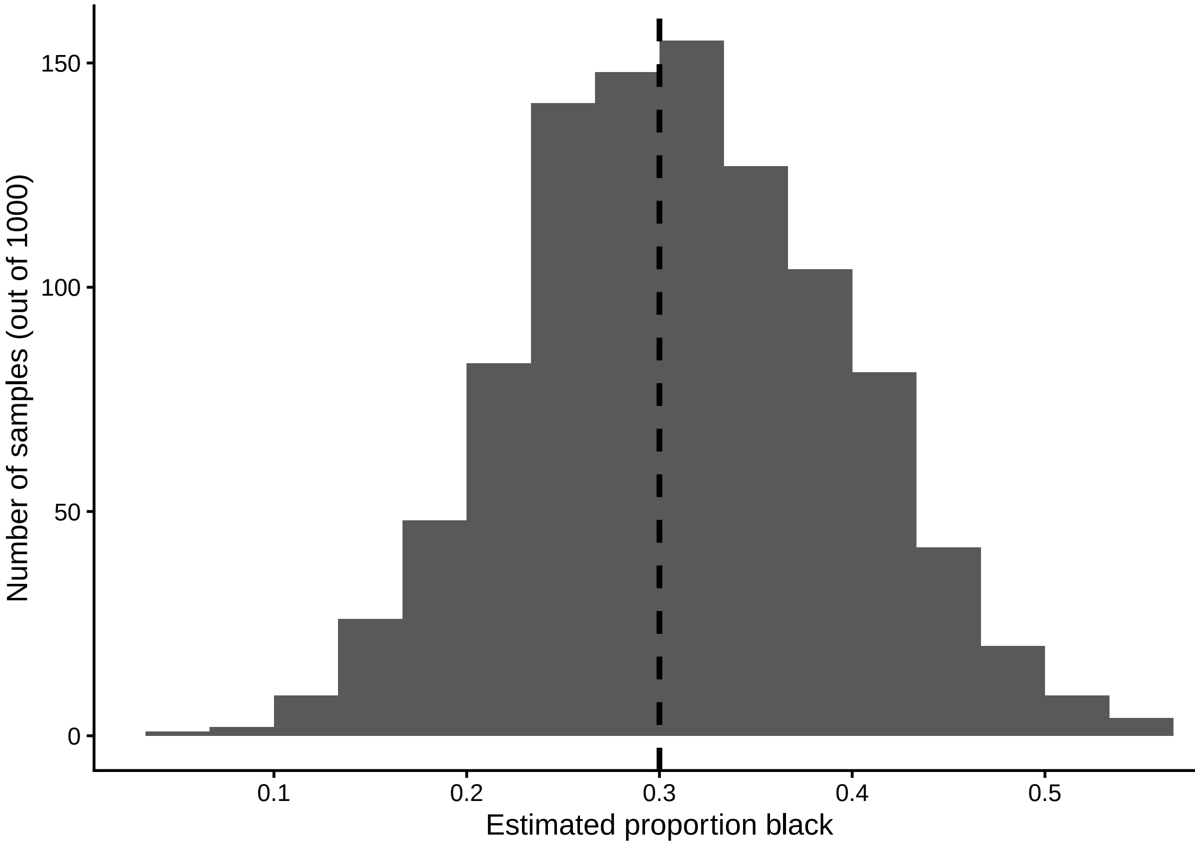


Figure 5.1: Distribution of estimates of the proportion of black squirrels from 1000 unique samples each with 30 squirrels.

squirrels are easier to spot in the woods than the gray morph, which increases the probability of black squirrels entering the sample. Figure 5.2 shows one example of what a sampling distribution might look like when black squirrels are more likely to be detected than the gray morph.

```r
set.seed(4)

p_true <- 0.30
n <- 30
reps <- 1000
w_black <- 1.25  # 25% higher detection for black vs gray (gray weight = 1)

# Effective probability a *detected* squirrel is black
p_det <- (w_black * p_true) / (w_black * p_true + 1 * (1 - p_true))

# Sampling distribution of p-hat based on detected squirrels (n fixed at 30)
phat <- rbinom(reps, size = n, prob = p_det) / n
d <- data.frame(phat = phat)
mean_phat <- mean(phat)

ggplot(d, aes(x = phat)) +
  geom_histogram(binwidth = 1/n, boundary = 0, closed = "left") +
  geom_vline(xintercept = p_true, linetype = "dashed", linewidth = 1) +
  geom_vline(xintercept = p_det, linetype = "solid",  linewidth = 1) +
  annotate(
    "text",
    x = p_true, y = Inf,
    label = paste0("Truth = ", sprintf("%.2f", p_true)),
    vjust = 1.2, hjust = 1.2
  ) +
  annotate(
    "text",
    x = mean_phat, y = Inf,
    label = paste0("Mean estimate = ", sprintf("%.2f", mean_phat)),
    vjust = 1.2, hjust = -0.2
  ) +
  labs(
    x = "Estimated proportion black",
    y = "Number of samples (out of 1000)"
  ) +
  theme_classic()
```

Here we can see there's a clear problem in our sampling process. There's variation in the possible estimates just like the first case, but notice that the estimates are more consistently overestimates than underestimates. On average, our estimates are too high. If the estimates tend to systematically differ from the truth,

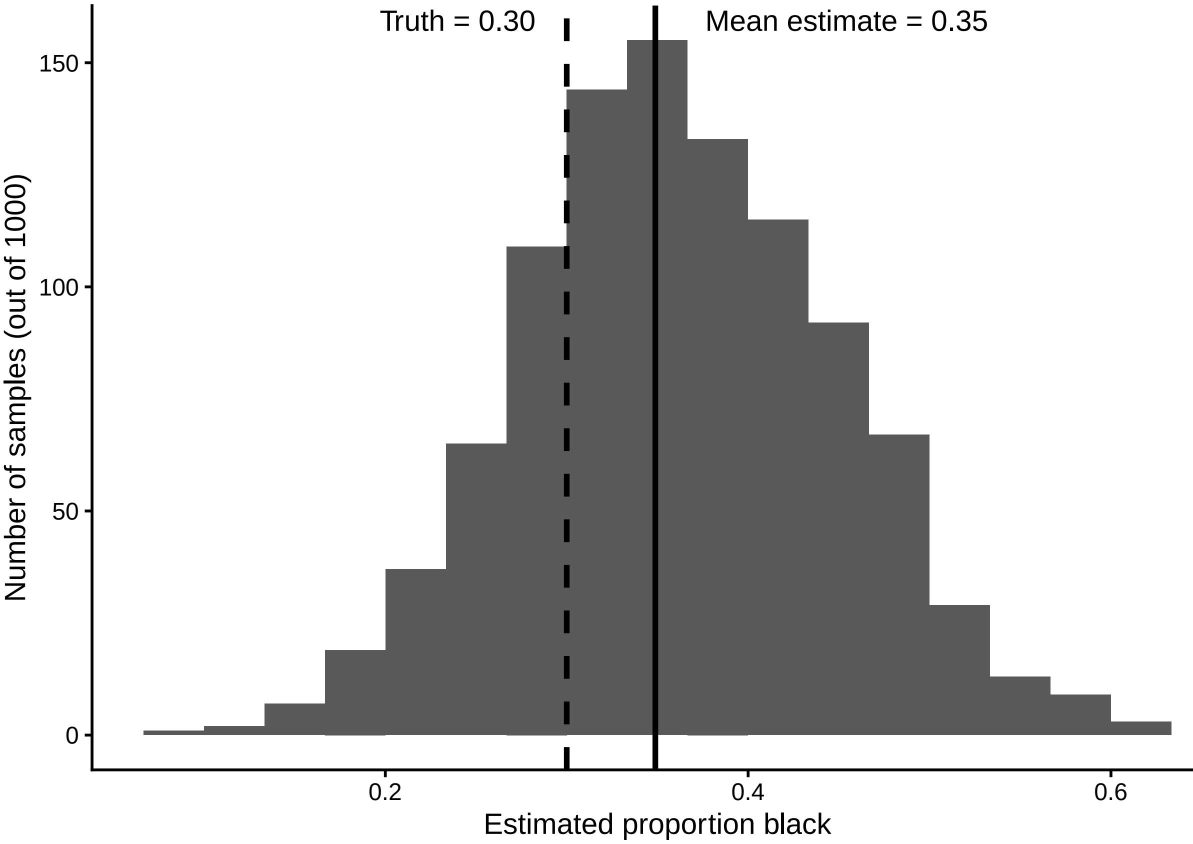


Figure 5.2: Example of a sampling distribution showing biased estimates.

then the estimates are ***biased***. Bias is a consistent discrepancy between the estimates and the true value of the parameter. Estimates may be biased high or biased low. When we design a study and sampling process, it is important to use strategies to maximize accuracy (minimize bias). More on that in a bit.

5.4.2 Precision

When we examine the range of possible estimates based on a sampling design, we can examine how much variation we expect to see in the estimates from sample to sample. When we repeatedly sample 30 squirrels, we know our estimates of the proportion black will vary because the squirrels that enter our sample have an element of chance to it. Sometimes you end up with 8 black morphs, other times you sample 11 black morphs, even though the expectation is 9 black squirrels based on a true proportion of 30%. We defined this kind of chance variation in which individuals enter a sample **sampling error**, and it leads to deviations of the sample estimate from the parameter value. The variation in sample estimates from sample to sample is called **precision**. Precision is a measure of how consistent estimates should be when we repeatedly sample from a population, using the same sampling methodology each time. If the estimates are consistently around the same value, then the estimates are precise. If the estimates vary wildly, then the estimates are imprecise.

From a coarse perspective, we can gauge the precision of an estimate by examining the width of a sampling distribution. To illustrate this, consider two sampling processes for estimating the proportion of black squirrels, one where we sample 30 squirrels each time, and another where we sample 60 squirrels each time. I've simulated each sampling process 1000 times and show the results in (Figure 5.3. What do you notice that's different between these sampling distributions?

These sampling distributions differ in a couple important ways. First, notice that the sampling distribution for N = 30 squirrels is wider than the sampling distribution for N = 60 squirrels. There is a clear difference in precision between the two sampling approaches. Sampling with N = 30 squirrels leads to less precise estimates of the proportion black than sampling with N = 60 squirrels. With fewer squirrels, you should expect to see much more variation in the potential estimates. If you're having trouble understanding why, consider a more extreme example. What if I had only sampled N = 2 squirrels? The only possible estimates of the proportion black would be 0, 0.5, and 1. Conversely, suppose I sample all N = 300 squirrels? In that case, I would get 0.3 every time. Increasing sample size increases precision.

Second, notice that regardless of whether the sample size is N = 30 or 60 squirrels, the most likely value of the estimate is 0.3, which is the truth. In other words, although sample size increases precision, it doesn't affect accuracy. The estimates are accurate whether the sample size is N = 30 or N = 60.

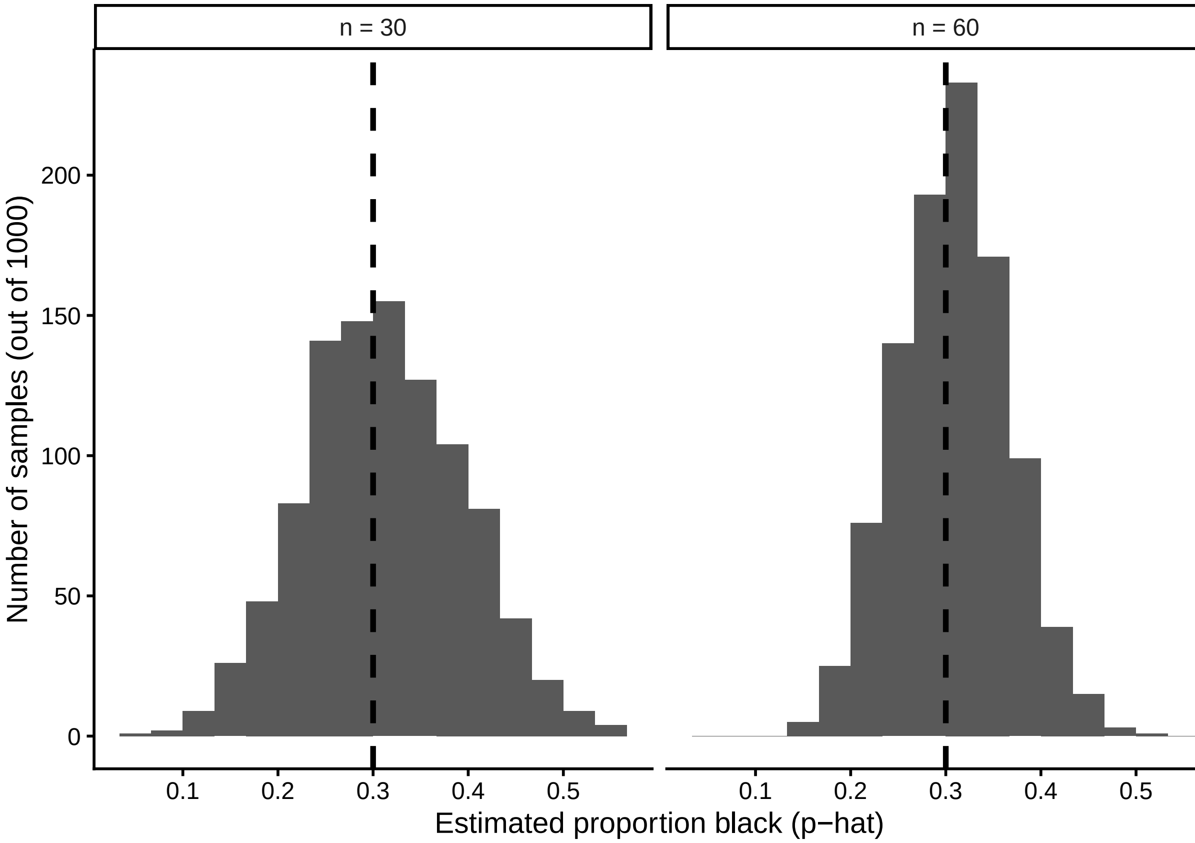


Figure 5.3: Sampling distributions for estimates of the proportion of black squirrels based on samples of size 30 (left) vs. 60 (right).

5.4.3 Considering accuracy and precision together

When designing a study to sample individuals from a population of interest, your goal should be to design sampling schemes that maximize accuracy and precision of estimates. Both matter, and one doesn't guarantee the other. In other words, you can have samples that are accurate and precise accurate and imprecise, inaccurate and precise, or inaccurate and imprecise. The figure below shows each of these four outcomes for our example of estimating the proportion of black squirrels. The ideal situation is obtaining an estimate that is accurate and precise, in other words consistently getting the right answer. The worst outcome is inconsistently getting the wrong answer (inaccurate and imprecise).

5.5 Maximizing accuracy and precision of estimates

High quality estimates are both accurate and precise. When designing a scientific study, it is critical to design the data collection scheme in a way that will maximize accuracy and precision. At this point I'd like to intorduce two elements of study design that affect accuracy and precision.

5.5.1 Random sampling

The most effective strategy to minimize bias is to take a ***random sample*** of individuals from the population of interest. A random sample means that every individual in the population has an equal chance of being included in the sample. For our study estimating the proportion of black squirrels, that means black and gray morphs must have an equal probability of entering the sample. Often this is easier said than done and often requires some domain knowledge. For example, I know that black squirrels are more visible than gray ones, but I also know from experience that the black morph tends to be less active than the gray morph. I either need to design my sampling to mitigate those potential biases, or I need to explicitly account for them when I estimate the proportion of black squirrels.

In other words, it's much better to be aware of potentially biases in a sample than to ignore them. Consider an epidemiologist interested in estimating the proportion of people in a city with influenza. One might be tempted to use medical records from people who visit a doctor to estimate flu prevalence, but it's important to consider that records of individuals who visit the doctor are not representative of the broader population. People who went to the doctor likely had significant - potentially even severe - symptoms. People who get the flu but who have mild symptoms are likely underrepesented in medical records, and so relying on those records would lead to an overestimate of flu prevalence. Data

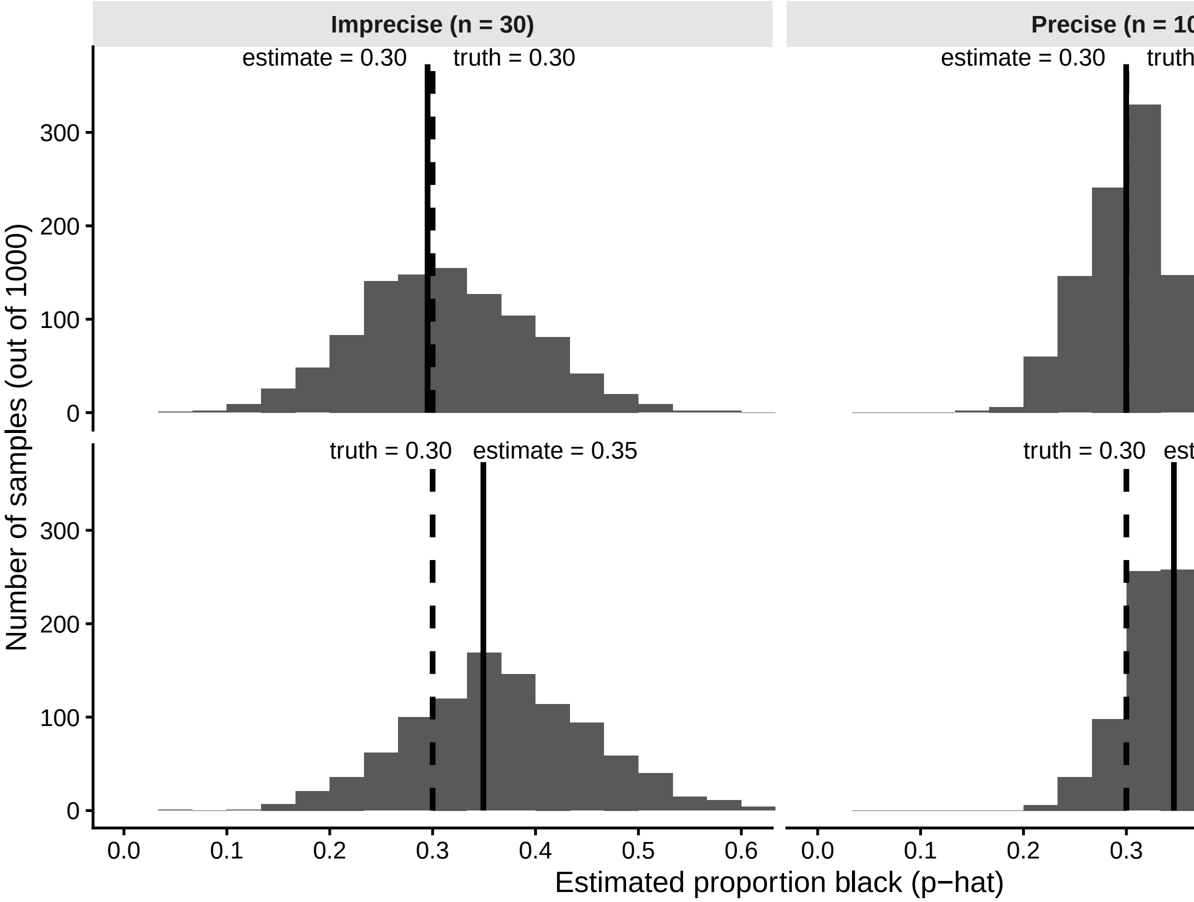


Figure 5.4: Four combinations of accuracy and precision. Precision increases with sample size (n = 100 vs n = 30). Inaccuracy is illustrated by a 25% detection bias favoring black squirrels.

from sources like medical records are often called a **sample of convenience** because the data are easy to access, but rarely are the data representative of the population of interest.

Bias in estimates is similarly caused by **volunteer samples**. The classic case is survey design. Suppose residents in a small town are debating whether to purchase land to build a park for recreation. The park will include a playground, a splash pad, fields for soccer, baseball, and football, and some tennis courts. The tennis courts and fields will all have lights, allowing people to play in the evenings. The town board wants to see if residents support the idea of building the new park, so they create a survey and mail it to residents. Who's most likely to respond? In this case, volunteer bias would occur if the people most likely to respond are ones who have very strong opinions. For example, residents who are really passionate about having recreation fields may be most likely to respond, causing an overestimate in the proportion of residents who favor creating the park. Conversely, imagine there's a large neighborhood next to the land being considered for purchase, and those residents don't want bright lights on at night in their backyard. Support for the park may be underestimated if the residents in that neighborhood are more inclined to respond to the survey.

5.5.2 Replication

The clearest way to maximize precision (i.e., minimize sampling error) is to maximize replication, the number of independent sampling units in a study. The number of independent sampling units is also referred to as the **sample size**. Remember that sampling error is the chance variation in estimates due to sampling only a subset of individuals from a population. If you could include all individuals from the population in your study, there would be no sampling error. Sampling error decreases as you increase the sample size, which is exactly what we showed earlier in the chapter with the squirrelcolor example. There was much more variation in estimates of the proportion of black squirrels when sampling 30 squirrels than when sampling 60 squirrels. Precision increases with sample size.

Note that I defined replication as the number of **independent** sampling units. Independence is really important, because it affects how you count sampling units and ultimately determine the sample size in a study. Replicates are independent if the measurements from one replicate have no influence on the measurements from other replicates. Sometimes individuals in a study are not independent. For example, what if gray and black squirrels are not randomly distributed in space? Perhaps they prefer different types of habitats. In that case, if I see one black squirrel, that increases the chance of seeing another black squirrel. In a situation like this, we would need to consider the way in which squirrels are clustered in larger sampling areas, like backyards or parks.

As another example, consider again the survey about building a park. Should the researcher include multiple people from the same household in the survey?

They certainly could, but the responses of individuals from the same household are likely correlated. In other words, individuals from the same household may have similar preferences about the park. If you included these individuals in your study and counted them as independent replicates, your estimates will be less precise than you think. Counting individuals as unique replicates when they are not independent is referred to as ***pseudoreplication*** and can lead to incorrect inferences. When in doubt about independence, a useful rule of thumb is to ask yourself **What is the unit that was selected for inclusion in the sample?**. That can be a useful place to start for thinking about sample size. In our backyard squirrel example, we're first and foremost selecting backyards. The individual squirrels are nested within those yards.

Important note: Sample size affects precision, but it does not affect accuracy of estimates. Remember that accuracy is how close, **on average**, your sample estimates are to the true value of the parameter. There's no inherent effect of sample size on accuracy. In other words, low sample size does not cause bias.

5.5.3 Take-home points

- Most scientific studies rely on samples, not full populations, so the empirical values we quantify are estimates, not the truth.
- Sampling distributions describe how estimates vary across repeated samples.
- Accuracy is about whether estimates are correct (consistent with the truth) on average. Estimates that are consistently wrong are biased. Random sampling is a key study design strategy to reduce bias.
- Precision is about how much estimates vary from sample to sample due to sampling error. Increasing sample size is a key study design strategy to increase precision.

Chapter 6

Probability as the Language of Uncertainty

One of the central goals of epidemiology is to monitor the prevalence of disease. Basic information on the prevalence of disease is used by public health officials to guide decision-making on interventions. Perhaps there's no better example of this than the Covid-19 pandemic that started in late 2019 and led to dramatic measures intended to reduce the spread of the virus, such as social distancing, mask wearing, travel restrictions, school shutdowns, and quarantine. These interventions place significant limitations on the freedoms we tend to take for granted, and so the benefits of these measures to the public should outweigh the costs.

Consider this scenario. Imagine you are the lead epidemiologist in a community of 10,000 people, working with public health officials to make decisions about interventions to minimize the spread of a new viral disease. Based on a cost-benefit analysis, public officials determined that interventions will be enforced if the prevalence of the disease reaches 10%. As the lead epidemiologist, your goal is to monitor the prevalence of the disease. Fortunately a test for the infection is available, and it's completely fool proof. When someone has the virus, the test is always positive, and when someone doesn't have the virus, the test is always negative. Diagnostic tests are of course rarely perfect in this way, but we'll make this simplifying assumption for now.

Knowing you can't possibly test everyone in your population of 10,000, you decide on a strategy to sample individuals from the population for testing. Having studied statistics, you are well aware that the fundamental problem of statistics is the inevitable uncertainty about estimates made from samples. If you test only a sample of the individuals in the population, the proportion infected in the sample will differ from the true proportion infected due to sampling error or systematic biases. To minimize bias, you decide on a random sampling approach. In truth, random sampling in public health is hard to do, but it's a useful starting point.

Eager to get your first estimate, you test a random sample of individuals and find that 6.7% are infected. You share the finding with your public health colleagues, who are generally happy to see the number is under 10%. But one astute colleague looks at the estimate and asks you "How good is the estimate?"

How good is your estimate? The question implies that your estimated proportion infected differ from the truth. Even though only 6.7% in the *sample* were infected, it's still possible that more than 10% in the *population* are infected. Even with random sampling, we expect a difference between teh sample estimate and the truth because of sampling error. In this chapter, we directly confront this problem by examining how we can quantify uncertainty about our estimates in the language of **probability**.

6.1 Defining probability

Let's start with some terms. When we test an individual for infection, the test is called a *trial*. A trial is simply any process that produces a probabilistic *outcome*. There are only two possible outcomes of the test: positive or negative. These outcomes are **mutually exclusive**, meaning that an individual cannot test positive *and* negative for the infection at the same time. The set of all possible mutually exclusive outcomes is called the *sample space*. The sample space is often denoted Ω and defined in brackets. For example, the sample space for infection status is $\Omega = \{\text{positive, negative}\}$.

When examining a probability, we are interested in the probability of a particular *event* that we a define. The event could be as simple as the occurrence of a single outcome, such as an individual testing positive. But events can also be defined as sets of outcomes. For example, consider tossing a six-sided die, where the sample space is $\Omega = \{1,2,3,4,5,6\}$. Here we could define the event as rolling an even number, which includes three of the possible outcomes.

Numerically, the values of probability are bounded between 0 and 1. A probability of 1 means that the event of interest is certain, whereas a probability of 0 means the event of interest is impossible. Mathematically probability is abbreviated as P, so we can say a certain event has $P = 1$ and an impossible event has $P = 0$. The events of interest are usually appended parenthetically. For example, when you toss a coin with two sides, *head* and *tails*, you can be (reasonably) certain the probability of heads or tails is one:

$$P(\text{heads or tails}) = 1$$

Conversely, you can be (reasonably) certain that you won't see something other than heads or tails, which can be written as

$$P(\text{not heads or tails}) = 0$$

Events can be any statement of interest. For our epidemiology example, we're interested in the probability that an individual is infected: $P(\text{infected})$. For this event, we probably won't be as confident as we were for our coin tossing example, but we can express our confidence numerically as a probability. Because $P = 1$ implies certainty and $P = 0$ implies impossibility, all values of probability are bounded between 0 and 1. The higher the value of probability, the more likely it is that the event is true. The lower the value of probability, the more likely it is that the event is false. For example, the value $P(\text{infected}) = 0.7$ implies it is more likely than not that an individual is infected. The value $P(\text{infected}) = 0.1$ implies only a small chance of an individual being infected. What really matters here is that values other than $P = 1$ and $P = 0$ imply a *lack of knowledge* about an event, or in other words, uncertainty! $P = 1$ and $P = 0$ mean we are certain an event is true or false, and all other values mean we are uncertain, with the degree of certainty of the event being true scaling numerically with the value of probability.

Although these simple rules of probability may appear straightforward, philosophically there has been much discussion about the meaning of probability. My goal here is not to provide an exhaustive overview of the various interpretations, but I do want to introduce two common interpretations that will be themes throughout the remainder of the book.

6.1.1 Frequentist definition

The **frequentist** interpretation of probability is one that I suspect you might be familiar with: probability is the proportion of trials n where we observe the event of interest, X:

$$P(X) = \frac{X}{n}$$

Let's just assume for now that of the 10,000 people in the community, 500 are infected with the virus. Based on the frequentist definition, the probability of the infection (I) is

$$P(I) = \frac{I}{n} = \frac{500}{10000} = 0.05$$

The frequentist definition simply looks at the frequency of the event of interest relative to the total number of trials. A probability is a proportion, following the rules we've already established where values must be between 0 and 1. Probabilities can also be expressed as percentages. To do this, simply multiply the proportion by 100. Saying the probability of infection is 0.05 is the same as saying 5% of the population is infected.

Now in practice, how do we know the frequentist probability? As the formula implies, we could go track down all n individuals in the population and give them our fool proof test for the viral infection. But you know that tracking down every individual in the population is usually not feasible, so we usually need to estimate the probability of interest with a sample of data. For example, imagine you randomly sample $n = 15$ individuals and find one of the 15 tests are positive. In this case, the estimated probability of infection ($\hat{p}$) is

$$\hat{p}_{infected} = \frac{I}{n} = \frac{1}{15} = 0.067$$

Here the carrot symbol simply indicates that our quantity is an estimate based on a sample.

There is a way of logically deriving frequentist probabilities, but this approach is restricted to only the most simple of examples, such as tossing a coin or rolling a die, where you can count the number of times the event of interest occurs out of the sample space. When we flip a coin, there are two sides, heads and tails. Assuming we have a fair coin and flip, the probability of heads must be 0.5, because heads represents half of the sample space. Similarly, if we roll a six-sided die with the numbers 1, 2, 3, 4, 5, and 6, the probability of seeing a "four" must be $\frac{1}{6}$, because four represents one out of six possible outcomes. This approach to quantifying probabilities is called the **principle of indifference**, in which there is no reason to believe one possible outcome is any more or less likely than the other possible outcomes. Based on the principle of indifference, the probability of each outcome is equal, and so numerically the probability of each outcome is just one divided by the total number of possible outcomes.

But as I already mentioned, this logic of deriving probability is not widely applicable. Consider our test of whether an individual is infected with a virus. Here each individual is infected or not infected. In this limited sample space, infected represents one of two possible outcomes, so isn't the probability of infection 0.5? No! When we apply the prinicple of indifference to coin tossing or die rolling, we make important assumptions, namely that we have fair coins and dice and that the coin flips and dice rolls are conducted in a random way. In other words, deriving probabilities mathematically from the sample space requires assumptions about the external forces that can affect the probability of heads, or the probability of rolling a four. If I do not give the coin a fair toss, for example by just dropping the coin flat with the heads face up, then the probability of heads will very likely *not* be 0.5. There are a multitude of external forces that affect the likelihood of an individual being infected with a virus, such as exposure, immune function, public health measures, and even the prevalence of the infection itself. In other words, the principle of indifference is a poor model of how infection works.

6.1.2 Bayesian definition

The **Bayesian** way of thinking about probability is as a strength of belief. What do you believe is the probability of infection? We make these kind of subjective probability assessments all the time. If you look outside and see storm clouds on the horizon, you might be inclined to believe it's more likely than not that it will rain today. When you are deciding which route to take to work, you might notice it's the middle of rush hour and conclude there's a 90% chance that you'll end up in stop-and-go traffic on the interstate. When watching your favorite college basketball team take on the #1 ranked team, you might conclude your team has only a 10% chance of winning.

These subjective beliefs aren't empirically computed by frequencies across multiple trials or from opportunity from the sample space. Rather, they are subjectively computed in your mind based on your knowledge, experience, and intuitions. If you've never heard of anyone being infected with the viral infection being investigated, you might be inclined to conclude it's more likely than not that the prevalence of the virus is below 10%. In this case you're assigning a subjective probability statement (>50% strength of belief) about the frequentist value of a probability (the proportion of individuals infected being below 10%).

In *Doing Bayesian Data Analysis*, the author John Kruschke talks about how subjective beliefs about probability can be calibrated by comparing those beliefs with events that have known probabilities. For example, suppose I offer you two choices. You can win $20 if you flip a coin and the result is heads, or you can win $20 if your favorite college basketball team beats the #1 ranked team. If you choose the coin flip, you are implicitly concluding that the probability of your team winning is less than 50%.

Now, Bayesian probabilities aren't always completely subjective. Indeed, as we will soon find out, there methods for updating our subjective beliefs about a probability (which we will call a *prior probability*) with frequentist probabilities informed by data that we collect. More on that soon.

6.2 Probability rules

One can take an entire course on the mathematics and theory of probability, but here our goal is to apply some basic knowledge of probability to science and data analysis. That said, familiarity with some basic rules of probability is necessary to do applied statistics. We already know that probability is bounded between 0 and 1, but there are some additional rules we need to be familiar with in order to use probability for quantifying our uncertainty in data analysis. Importantly, these rules apply whether you interpret probability from a frequentist or Bayesian perspective. Let's take a look at those rules [^ch07-1].

6.2.1 Individual events

We first look at two rules that apply when one is interested in the probability of a single event of interest in isolation. For example, suppose you're driving into work and will go through one traffic light. Traffic lights in the United States can be red, yellow, or green. For the two rules that follow, we consider the probability of one of those outcomes in isolation, such as $P(\text{red})$.

Addition rule

$$P(A \text{ or } B) = P(A) + P(B)$$

If outcomes A and B are **mutually exclusive**, then the probability of either A or B occurring is the sum of their individual probabilities.

Let's apply the addition rule to an example. Consider the traffic light example, and let's assume the probability of red is 0.48, the probability of yellow is 0.04, and the probability of green is 0.48. Each outcome is mutually exclusive, because the traffic light cannot - for example - be red and green at the same time (barring a malfunction). When events are mutually exclusive in this way, we can apply the addition rule to quantify the probability of one event or another. For example, the probability of the light being green or yellow when driving through it is:

$$P(\text{green or yellow}) = P(\text{green}) + P(\text{yellow}) = 0.48 + 0.04 = 0.52$$

We can extend this rule beyond two events. For example, the probability of green, yellow, or red is

$$P(\text{green or yellow or red}) = P(\text{green}) + P(\text{yellow}) + P(\text{red}) = 0.48 + 0.04 + 0.48 = 1$$

Here we see that the probabilities of each possible mutually exclusive outcome must sum to 1.

Not rule

$$P(\text{not } A) = 1 - P(A)$$

The probability of an event not occurring is one minus the probability that it occurs.

The not rule simply states that once you know the probability of an event being true, you also know the probability that the event is false, computed as one minus the probability of the event being true. If the probability of the light being green or yellow is is 0.52, then the probability of it not being green or yellow is

$$P(\text{not green or yellow}) = 1 - P(\text{green or yellow}) = 1 - 0.52 = 0.48$$

6.2.2 Joint events

The first two rules apply to mutually exclusive events in isolation, but often we are interested in the joint probability of multiple events occurring at the same time. For this situation, let's return to the goal of estimating prevalence of a viral infection. The epidemiologist does not know the true prevalence, but let's assume it is 5%. Let's also assume that 10% of individuals in the population are left-handed. What is the probability of selecting an individual who tests positive and is left-handed? Here we are interested in a **joint probability**, namely the events "infected" and "being left-handed" at the same time.

6.2.2.1 Independent events

In cases where the joint events are **independent**, we can apply the following rule to quantify the joint probability:

Multiplication rule

$$P(A \text{ and } B) = P(A)P(B)$$

When events A and B are independent, the probability of A and B is the multiplication of the probabilities of A and B individually.

For events to be independent, one event must have no impact at all on the probability of another event. In the context of testing and handedness, this would mean that the handedness of a person must have no impact on infection status, and vice versa. If that's the case, then we can quantify the probability of individual being infected and left-handed as

$$P(\text{infected and left-handed}) = 0.05 * 0.10 = 0.005$$

When the prevalence of the infection is 5%, the probability that an individual is infected and left-handed is only 0.5%. A rare event indeed!

6.2.2.2 Dependent events (conditional probability)

In the last example, we assumed handedness gave us no information about the probability of infection (i.e., they are independent). Now let's consider a different case: infection status and vaccine status. Unlike handedness, vaccine status plausibly gives us information about a person's infection status. Indeed, vaccines are designed to reduce the likelihood of being infected, so we shouldn't expect infection status and vaccine status are independent events. In other words, the probability of infection likely depends on vaccine status. This is getting interesting! Indeed, non-independence is at the heart of many research

Table 6.1: <span style='color:black; font-weight:bold;'>Counts of Infection Status vs. Vaccination (N = 10,000)</span>

Status	Vaccinated Count	Unvaccinated Count	Total Count
Infected	210	290	500
Not Infected	6790	2710	9500
Total	7000	3000	10000

questions, particular those about causal explanation. Does vaccine status affect infection status? Such a question can be addressed quantitatively by examining joint probabilities under assumptions of independence vs. dependency between the events.

To analyze joint probabilities of multiple events when those events are not independent, we need to define **conditional probability**. A conditional probability can be defined as the probability of event A given that know event B is true. Let's formalize another rule:

Conditional probability

$$P(A|B) = \frac{P(A \text{ and } B)}{P(B)}$$

The vertical bar (|) means "given", so $P(A|B)$ reads "the probability of A given B". For example, the probability of infection status among vaccinated people can be written as $P(\text{infected} \mid \text{vaccinated})$.

Consider Table 6.1, which shows our population of 10,000 people broken down by infection status and vaccine status. First, notice that the overall probability of infection is 5% as assumed (500 infected out of 10,000), and the overall probability of vaccination is 70% (7,000 vaccinated out of 10,000). These are **marginal probabilities** because they are computed by aggregating across the levels of the other variable. In other words, to compute the marginal probability of infection, we have to consider the number of people infected among those who are vaccinated (210) and not vaccinated (290). Together there are 500 people infected, which is 5% of the total population.

Second, from these data we can compute the joint probabilities of each combination of events:

$$\begin{aligned}
P(\text{infected and vaccinated}) &= \frac{210}{10000} = 0.021 \\
P(\text{not infected and vaccinated}) &= \frac{6790}{10000} = 0.679 \\
P(\text{infected and unvaccinated}) &= \frac{290}{10000} = 0.029 \\
P(\text{not infected and unvaccinated}) &= \frac{2710}{10000} = 0.271
\end{aligned}$$

These joint probabilities make up the entire sample space for the joint event "infection status and vaccination status", so following our addition rule, they should sum to one.

Now let's examine the question of conditional probability. Are the events infection status and tattoo vaccination status independent or dependent events? If they are independent events, then the probability of infection should be the same for people are vaccinated and unvaccinated. Conversely, if they are dependent events, the probability of infection will differ between people who are vaccinated and unvaccinated. Let's compute the conditional probabilities following our rule:

$$P(\text{infected} \mid \text{vaccinated}) = \frac{P(\text{infected and vaccinated})}{P(\text{vaccinated})} = \frac{210}{210 + 6790} = 0.03$$

$$P(\text{infected} \mid \text{unvaccinated}) = \frac{P(\text{infected and unvaccinated})}{P(\text{unvaccinated})} = \frac{290}{290 + 2710} = 0.0967$$

Here we clearly see that the probability of infection differs by vaccine status. The probability of infection is over three times as likely for unvaccinated than vaccinated people. In other words, infection status is *not* independent of vaccine status.

How do we compute joint probabilities when events are not independent? There's a rule for that!

General multiplication rule

$$P(A \text{ and } B) = P(A|B)P(B)$$

This general multiplication rule is derived from the rule on conditional probability by simply isolating $P(A\ and\ B)$. As an example, if you knew that 3% of vaccinated people were infected and 70% of all people were infected, then you can quantify the joint probability of those who are vaccinated and infected as

$$P(\text{infected and vaccinated}) = P(\text{infected} \mid \text{vaccinated}) * P(\text{vaccinated}) = 0.03 * 0.7 = 0.021$$

Note that we can show infection status and infection status are not independent by testing the simple multiplication rule. The simple multiplication rule says the joint probability of independent events is the multiplication of their individual probabilities. Thus, if infection status and vaccine status are independent, the proportion of people who are infected and vaccinated should be $0.05 * 0.70 = 0.035$. But we ust showed that's not the case! The simple multiplication rule doesn't work here because it assumes independence, when in reality infection status is conditional on vaccine status.

6.2.3 General addition rule

Let's revisit the situation when we want to know the probability of an one event *or* another event. When the events are mutually exclusive, the addition rule tells us to simply add the probabilities of each event. What if the events are not mutually exclusive? For example, what if we want to quantify the probability that a person is infected or vaccinated. These events are not mutually exclusive because some infected individuals are vaccinated. If we simply the probability of $P(\text{infected})$ $P(\text{vaccinated})$, we will doulbe count the people who are both infected and vaccinated. To quantify the probability of event A or B when A and B are not mutually exclusive, we need to apply the **general addition rule**:

General addition rule

$$P(A \text{ or } B) = P(A) + P(B) - P(A \text{ and } B)$$

Let's apply this to the infection and tattoo example. Here I've abbreviated the events "infected" as "I" and "vaccinated" as "V":

$$P(\text{I or V}) = P(\text{I}) + P(\text{V}) - P(\text{I and V}) = 0.05 + 0.70 - 0.021 = 0.729$$

6.2.4 Quantifying marginal probabilities

When working with joint events, sometimes we want to work backwards from disaggregated probabilities to aggregated, or marginal probabilities. For example, suppose you knew the infection probability separately for vaccinated and unvaccinated individuals. What is the overall prevalence of infection? You might be tempted to compute the simple average of the two conditional probabilities $P(\text{infected} \mid \text{vaccinated}) = 0.03$ and $P(\text{infected} \mid \text{vaccinated}) = 0.0967$, but the mean (0.063) is not correct. Why not? The mean does not weight the conditions of vaccination status correctly. We know 70% of the population is vaccinated, whereas 30% of the population is unvaccinated. We need to account for the fact that more of the population is vaccinated. To quantify the marginal probability of infection (i.e., marginalizing the probability of infection over the levels of tattoo status), we basically quantify a weighted mean. This is the *law of total probability*:

Law of total probability

$$P(A) = \sum_{i=1}^{n} P(A|B_i)P(B_i),$$

where A is event A and B_i is condition i of event B.

Here we see the total (marginal) probability of event A is the weighted average of the probability of A across conditions of event B. Now let's apply this rule to quantify the probability of infection across conditions of vaccination (abbreviated each of the events):

$$P(\mathrm{I}) = P(\mathrm{I} \mid \mathrm{V}) * P(\mathrm{V}) + P(\mathrm{I} \mid \mathrm{UV}) * P(\mathrm{UV}) = 0.03 * 0.7 + 0.0967 * 0.3 = 0.05$$

6.3 Sampling from probability distributions

Now that we have a basic handle on probability rules, let's turn our attention to the topic that motivated this chapter. How can we use probability to describe uncertainty about estimates when we sample from populations? Recall that you're the lead epidemiologist trying to estimate the prevalence of an infection in a community of 10000 people. You have a fool proof test, when you tested 15 people at random, you found one was positive, leading to an estimated prevalence of 6.7%. How good is that estimate? Are you confident the prevalence is truly under 10%, the threshold that would trigger public health interventions? How confident?

Science is ultimately a process of making inferences about the world with data. Because the data we collect are almost always incomplete, being samples of the population of interest, those inferences must be made with uncertainty. In this section, I will outline how probability can be used to describe the uncertainty about the estimates we make with data.

6.3.1 Sampling from populations is probabalistic

Let's take a close look at how data are generated in the process of sampling a population for testing. We begin with a much simpler example than sampling a population of 10,000 people. In this example, assume we have a population of just four individuals, and we randomly select two individuals for testing. Each test returns a positive or negative result. To make matters more simple from a probability perspective, we assume individuals are selected from the population **with replacement**. This means each person is available to be selected for each of the two tests. This is analogous to randomly selecting multiple playing cards from a deck, but each time replacing the card you chose before you select a new one.

Suppose you go ahead and randomly select two individuals and find that one of the two tests was positive, and one of the two tests was negative. In probability terms, we know there was $X = 1$ positive test out of $n = 2$ trials. Those are the observed data. What insight do these data provide into proportion of individuals infected in the broader population of four people? Do we simply conclude the prevalence of the disease is 50% because one out of two tests were positive, or is there more to the story?

Let's examine the issue by assuming the true prevalence of the infection is 25% ($p_{infected} = 0.25$), meaning that just one of the four people in the population is infected. If only one of the four people are infected, how likely were we to see one positive out of two tests? To answer that question, let's look at all the possible outcomes when we conduct two tests, given that only one individual is infected. Figure 6.1 shows these possibilities in the format of a branching tree.

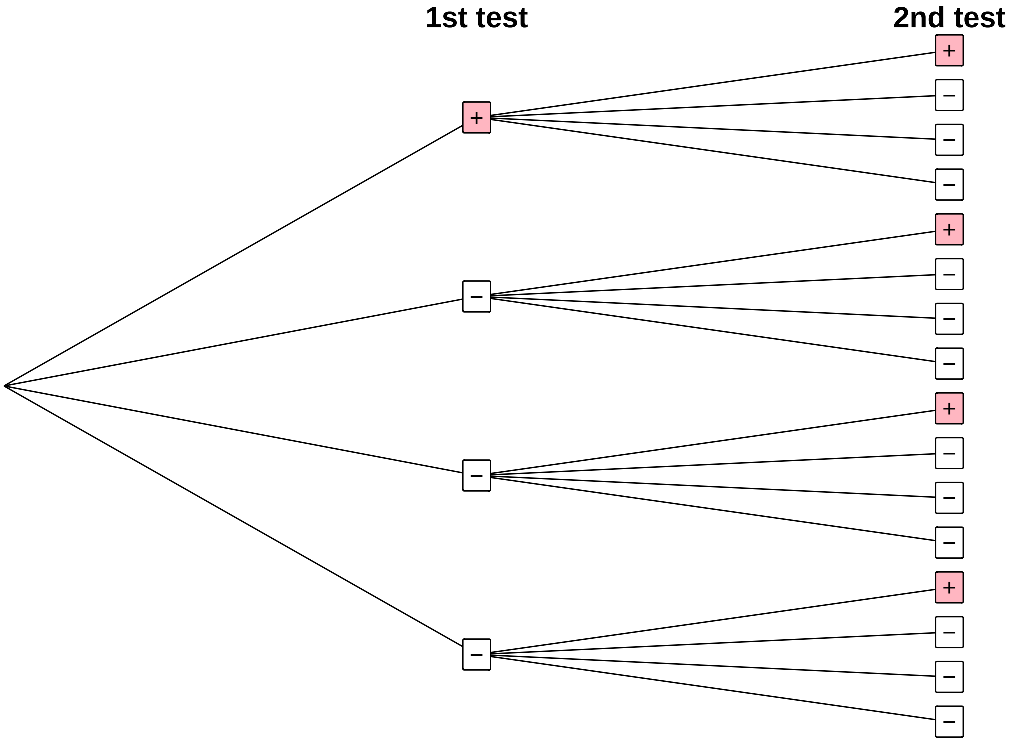


Figure 6.1: Probability tree showing the possible outcomes of testing when one of four people in the household are infected. Pink boxes with + indicate a positive test, and white boxes with - indicate a negative test.

The boxes in each branch of the tree represents the four individuals in the population. Only one of the four members is truly infected in this scenario, so for the 1st test, we see there's three more ways to see a negative test than a positive test. In other words, when we randomly select an individual for the first test, there's a 25% chance of a positive test and a 75% chance of a negative test when one of four individuals is infected. We then repeat that process for the 2nd test. Because we're sampling with replacement, the possibilities on the 2nd test are the same as the 1st test.

We see that when only one of the four people is truly infected, there are 16 possible outcomes when we conduct two tests: one outcome where both tests are positive, six outcomes where one test is positive and one test is negative, and nine outcomes when both tests are negative. Of these 16 possible outcomes, six outcomes are consistent with the data we observed, one positive test and one negative test. In other words, there was a 6 in 16 chance (37.5%) of getting the data we observed. Those outcomes are highlighted with bold lines in Figure 6.2.

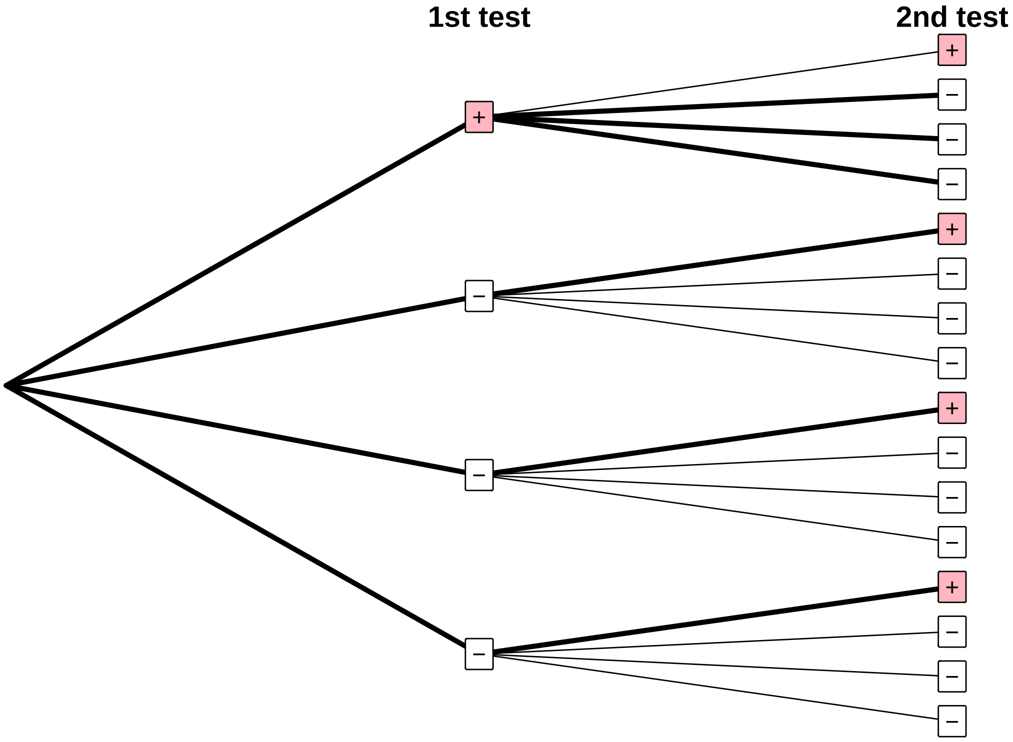


Figure 6.2: Probability tree showing the possible outcomes of testing when one of four people in the household are infected, highlighting outcomes consistent with the observed data.

Probability trees like this are really handy for looking at all the possible outcomes of joint events, but we can also apply our probability rules to compute the probability of one positive out of two tests when the probability of infection is 0.25. You might be tempted to apply the multiplication rule, because we want to know the probability of a positive test *and* a negative test. Applying that rule, we find $P(+\text{ and }-) = P(+)P(-) = 0.25 * 0.75 = 0.1875$. Why isn't this correct? The problem is not independence. We can reasonably assume the test results are independent if we are randomly picking people for each test. The problem is that there are multiple ways of observing exactly one positive test and one negative test! You can see this in the probability tree. It could be that the first test is positive and the second test is negative ("+-"), which can happen in three ways, or it could be that the first test is negative and the second test is positive ("-+"), which can also happen in three ways. We can separately quantify $P(+\text{ and }-)$ and $P(-\text{ and }+)$, and then apply our addition rule because these outcomes are mutually exclusive.

$$P(\text{One }+\text{ and One }-) = P(+)*P(-)+P(-)*P(+)P(\text{One }+\text{ and One }-) = 0.25*0.75+0.75*0.25 = 0.375$$

6.3.2 Discrete random variables

At this point I want to formalize some important concepts from the example of conducting two tests in a population of four. What we've shown is that **the process of sampling from a population is probabilistic**. When one of four people is infected and we take a sample of $n = 2$ tests, the outcome that we observe has an element of chance. The outcome X positive tests is called a **random variable**, where the term *random* implies the element of chance in terms of how we observe the variable. When we conduct $n = 2$ tests, we can observe one of three mutually exclusive outcomes: X = 0 positives, X = 1 positive, or X = 2 positives. Some of these outcomes are more likely than others, just like getting 5 heads when you flip a coin 10 times is more likely than getting one heads. But there's an element of chance, which is ultimately what creates much of the uncertainty we face when we estimate a quantity.

Random variable can be characterized by a **probability distribution**, which is the distribution of probabilities for each mutually exclusive outcome. Mathematically, we can define the probability that a random variable X takes on each possible value x as $(P(X = x))$. For the random variable $X =$ *number of positives out of two tests*, the probability distribution consists of $P(X = 0)$, $P(X = 1)$, $P(X = 2)$. First, we can quantify these probabilities by enumerating all of the possibilities in the sample space and count up the outcomes:

```
#P(X = 0 positives): 9 ways
9/16
```

```
## [1] 0.5625
```

```
#P(X = 1 positives): 6 ways
6/16
```

```
## [1] 0.375
```

```
#P(X = 2 positives): 1 way
1/16
```

```
## [1] 0.0625
```

Second, we could apply our probability rules:

```
#P(X = 0 positives) = P(negative and negative)
(0.75*0.75)
```

```
## [1] 0.5625
```

```
#P(X = 1 positives) = P(positive and negative)
(0.75*0.25) + (0.25*0.75) #2 ways this can happen
```

```
## [1] 0.375
```

```
#P(X = 2 positives) = P(positive and positive)
(0.25*0.25)
```

```
## [1] 0.0625
```

The probabilities for each possible of outcome, no matter how we quantify them, form a probability distribution. This particular example is a **discrete probability distribution** because the sample space is composed of discrete, mutually exclusive outcomes where we can quantify the probability of each as we have done. Figure 6.3 shows the probability distribution.

The probability of discrete outcomes is referred to as **probability mass**, but as Figure 6.3 shows, the y-axis of discrete probability distributions will often just be labeled "probability".

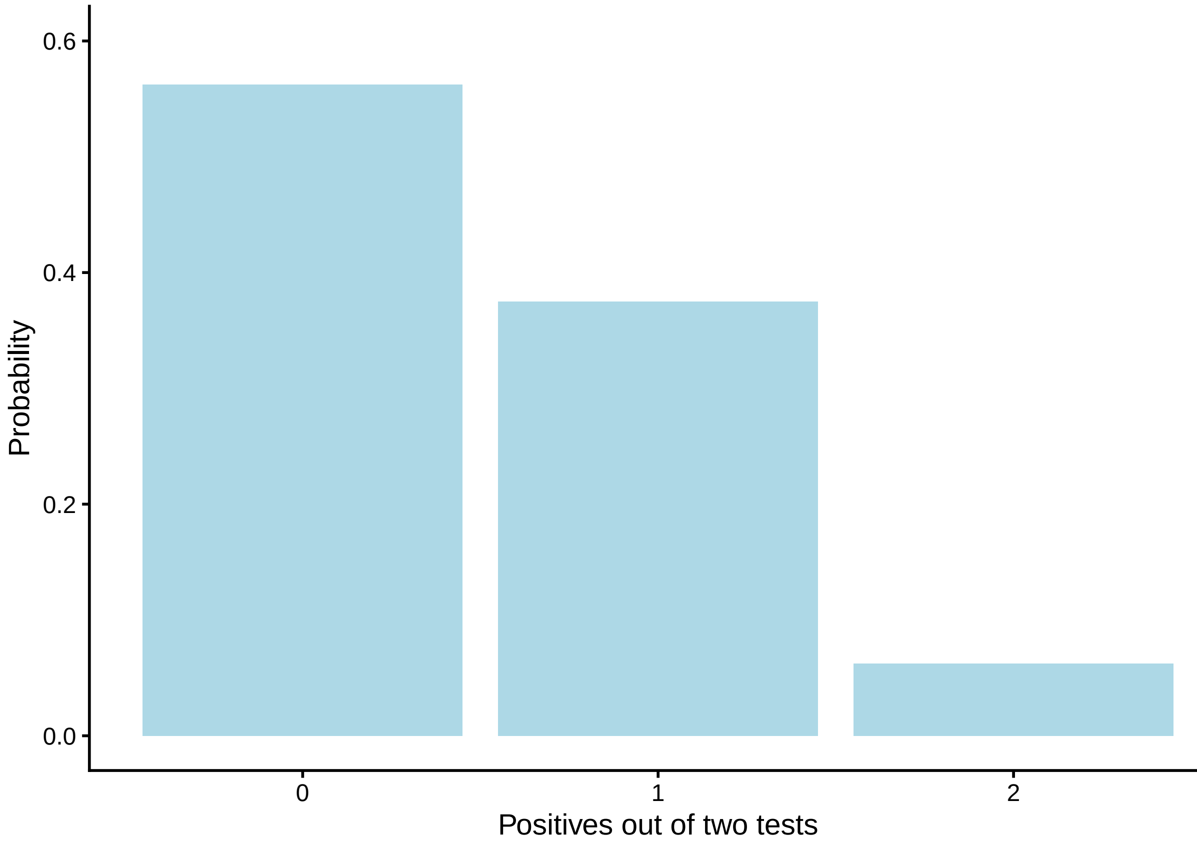


Figure 6.3: Discrete probability distribution of the number of positive tests out of two trials.

6.3.2.1 Binomial distribution

It was straightforward to quantify the probability distribution for the number of positives out of only two tests. The sample space is so small that we could easily quantify those probabilities by applying our probability rules. But now let's consider the larger sample of 15 tests from a community of 10,000. Manually enumerating the probability distribution for each possible number of positives would take considerable time! Fortunately we don't have to do that.

The random variable that we defined - the number of positives X out of n trials - is actually an example of a variable that follows a known mathematical function called the **binomial distribution**. A binomial distribution is a probability distribution for a binary variable where the outcome of that binary variable is examined across n trials, where each individual trial is called a *Bernoulli trial*. The sample space of each trial must be binary, for example heads and tails when flipping a coin, even or odd when rolling a die, and positive or negative when testing for a viral infection. The outcome being recorded is often generically referred to as a **success**. The n trials must be independent and have the same probability of success, p, which is the only parameter for the binomial distribution (e.g., probability of positive test). If those assumptions are met, the probability of x successes out of n trials can be computed with the binomial formula:

$$P(X = x) = \binom{n}{x} p^x (1-p)^{n-x}$$

The *binomnx* part of the formula reads "n choose x" and represents the number of ways x successes can occur out of n trials without regard to order (i.e., **combinations**). For example, we've already seen that 1 positive test can occur in two ways based on 2 trials. The number of combinations can be quantified as

$$\binom{n}{x} = \frac{n!}{x!(n-x)!}$$

Let's now consider the random sample of 15 people from our community of 10,000 where the prevalence of infection is 0.05. What's the probability of observing exactly one infection out of 15? Just apply the binomial formula:

$$P(X = 2) = \binom{15}{1} 0.05^1 (1-0.05)^{15-1} = 0.366$$

I'm not going to go into details here, but if you work through this formula, you'll see that all it's doing is applying the multiplication rule for independent events and the addition rule for mutually exclusive outcomes. And fortunately for us, we don't have to do these calculations by hand because R has a built-in function to compute the binomial probability: dbinom. For example, we can use the function to quantify the probability of 1 infections out of 15 trials:

```
dbinom(x = 1, size = 15, prob = 0.05)
```

```
## [1] 0.3657562
```

We can apply the dbinom formula to efficiently compute the binomial probability of all possible values of X positive tests out of 15 trials, when the probability of infection is 0.05, and then display the probability distribution in a graph (Figure 6.4):

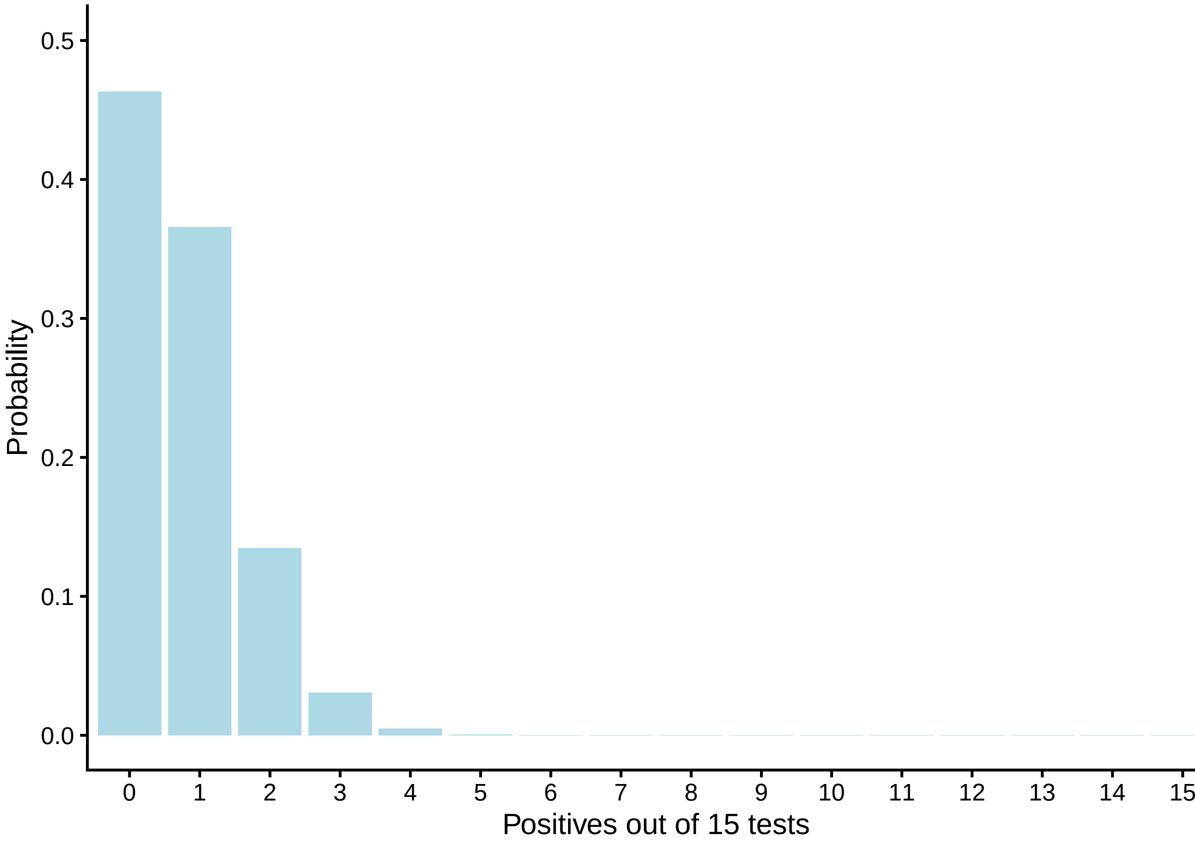


Figure 6.4: Discrete probability distribution of the number of positive tests out of 15 trials when 5% of the population is infected.

The probability distribution shows that the most likely outcome is no positives out of 15 when the prevalence is 0.05, and the probability of each outcome decreases as the number of positives increases. In fact, there's virtually no chance of observing six or more positives out of 15 tests when the true prevalence is only 5%. The probability distribution would look much different if 50% of the population was infected (Figure 6.5).

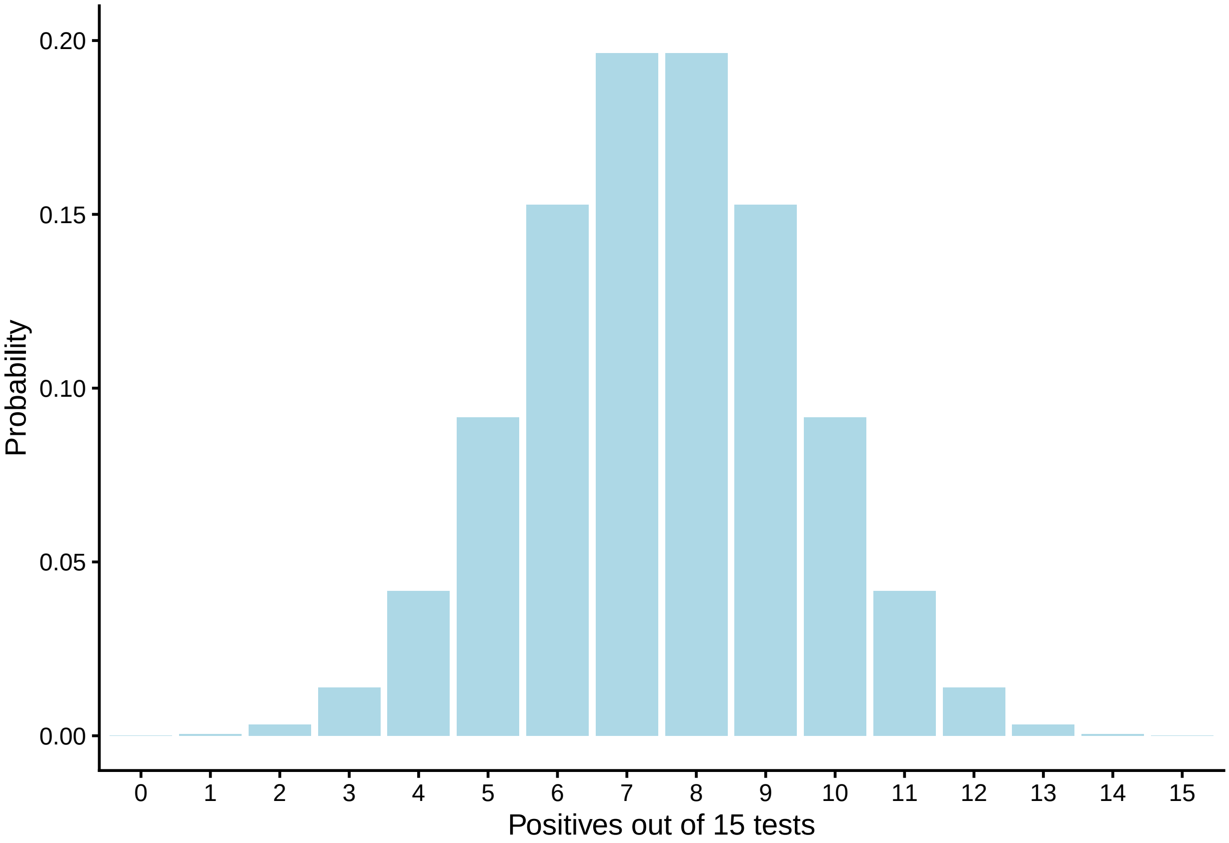


Figure 6.5: Discrete probability distribution of the number of positive tests out of 15 trials when 50% of the population is infected.

When 50% of the population is infected, the most likely outcomes are 7 or 8 tests out of 15, but note that many other values are plausible. In fact, it's more likely that we'll see a value other than 7 or 8 positives out of a sample of 15, even though X = 7 and X = 8 are the most likely outcomes:

```
1 - (dbinom(7, 15, prob=0.5) + dbinom(8, 15, prob=0.5))
```

```
## [1] 0.6072388
```

Here we see there's a 61% chance of observing a value other than 7 or 8 positive tests out of 15 when the prevalence of the disease is 50%. This is an excellent illustration of the problem of sampling error.

6.3.2.1.1 Mean for a binomial random variable

As we saw in Chapter 4, we can characterize the shape of distributions by their central tendency and variation. When examining probability distributions of random variables, central tendency is usually characterized by the mean, also known as the **expected value**. The expected value of each possible outcome X is

$$E[X] = \sum_X P(X)X$$

Expected value is simply weighing each possible outcome of the random variable X by its probability of occurring. Let's quantify the expected value of the number of positives out of 15 tests when the prevalence is 50%:

```r
#create a vector of outcomes
x <- c(0:15)

#sum the products of each outcome and its probability
sum(dbinom(x=x, size=15, prob=0.5)*x)
```

```
## [1] 7.5
```

We see exactly what we expect. When the prevalence is 50% and we conduct 15 tests, we expect 7.5 positives on average. What if the prevalence was 5%?

```r
sum(dbinom(x=x, size=15, prob=0.05)*x)
```

```
## [1] 0.75
```

When the prevalence is 5%, we expect 0.75 positives on average. Obviously you can't observe 7.5 or 0.75 positives for a discrete random variable, so you need to think about the expected value of discrete random variables as the long run outcome. That is, if you could take thousands of samples of 15 and compute the mean of the number of positives across samples, you'd expect the mean to be 7.5 positives if the prevalence was 50% and 0.75 positives if the prevalence was 5%. For binomial random variables, the mean of the probability distribution is equal to the true probability of success, so the expected value of a binomial random variable can be quantified simply as $E(X) = np$, where n is the number of trials and p is the probability of success. For example, when we take $n = 15$ tests from a population with $p_{infection} = 0.05$, the expected value can be computed as $E(X) = np = 15 * 0.05 = 0.75$.

6.3.2.1.2 Variance for a binomial random variable Recall that variance is a measure of variation in an outcome relative to the mean. The variance will be large when outcomes of the random variable can range widely around the mean, and it will be small when the outcomes show little variation around the mean. For a discrete random variable, the variance is quantified as

$$V[X] = \sum_{X} P(X)(X - E[X])^2$$

What does this mean? Just like we saw in the variance formula before, we're looking at how far each value of X is away from the mean, quantified as the squared difference between X and $E[X]$. We weigh each of those squared deviations by the probability of X, $P(X)$, and then we sum them up. Intuitively, you should see that the variance will increase as there are highly probable values of X far from the mean.

The variance formula for a binomial random variable can be simplified to the following:

$$V[X] = np(1-p)$$

From this formula we can see that the variance of a binomial random variable will be greatest when the probability of success is closer to 0.5. When $p = 0.5$ and $n = 15$, the variance is $V[X] = 15 * 0.5 * (1 - 0.5) = 3.75$. Any deviation from $p = 0.5$ leads to lower variance. For example, when $p = 0.05$ (e.g., 5% prevalence of infection), the variance is $V[X] = 15 * 0.05 * (1 - 0.05) = 0.7125$. At the extremes, when $p = 0$ or $p = 1$, the variance by definition must be 0. What this means is that probability distribution for binomial random variables will be somewhat bell-shaped with long tails when the probability of success is close to 0.5, and it will become skewed as the probability distribution deviates from 0.5.

6.3.3 Continuous random variables

Discrete probability distributions are straightforward because we can quantify the probability mass for each discrete outcome in the sample space. When we test an individual for a viral infection, the outcome is either positive or negative with probabilities $p_{positive}$ and $p_negative$. But not all random variables are so simple. Many variables form **continuous** probability distributions, such as height, weight, air temperature, and many more. The challenge of characterizing probability distributions for continuous variables is that there's an *infinite* number of potential outcomes in the sample space.

Consider the incubation period for a viral illness, which is the time between exposure to the pathogen and onset of symptoms. Incubation period is a continuous random variable because there's an infinite number of possibilities in the sample space. The incubation might be 2, 2.1, 2.15, 2.154, 2.1547 days, and so on. The probability of the incubation period being a particular value (e.g., 2.154794667 days) is 0 because there's an infinite number of possible values. If there's an infinite number of possible values in the sample space and each possibility had a non-zero probability, then the sum of the probability of all the mutually exclusive outcomes would be infinity, violating our rules of probability.

6.3.3.1 Probability density function

To characterize probability distributions for continuous variables, we use the concept of **probability density**. A **probability density function (pdf)** describes how probability is distributed across the possible values of a continuous random variable. Instead of assigning probabilities to individual points, the *pdf* reflects how "dense" the probability is at different values of the random variable.

To visualize this concept, we can approximate a continuous variable by discretizing the sample space into intervals. Figure @ref(fig:c06_f6) shows a simulated distribution of incubation times for 10,000 individuals, grouped into bins of 0.1 days. You can see this distribution has a slight positive skew, which makes sense given that time must be positive. The highlighted bin for 1.4-1.5 days contains 1,375 observations, meaning the probability *mass* for this interval is:

$$\frac{1234}{10000} = 0.1234$$

The probability density is defined as the ratio of the probability mass to the bin width, in this case:

$$\frac{0.1234}{0.1} = 1.234$$

Here we see that probability densities can be greater than 1. That's because probability density is a measure of how concentrated the probability is within a particular interval (a density!) rather than a probability of a particular observation. In this way probability density is similar to measuring human population density in a city. A city can have densities well over one person per square kilometer, even though the probability of finding one person in a randomly selected square kilometer is very low.

If we can quantify probability mass for a discrete interval, why do we use probability density? We use probability density because we want to describe the distribution continuously, without being constrained by arbitrary interval sizes. Narrower intervals result in smaller probability masses, approaching zero as the interval size approaches zero. The pdf allows us to describe the relative likelihood of observations without relying on a fixed bin size. Given a pdf, we can

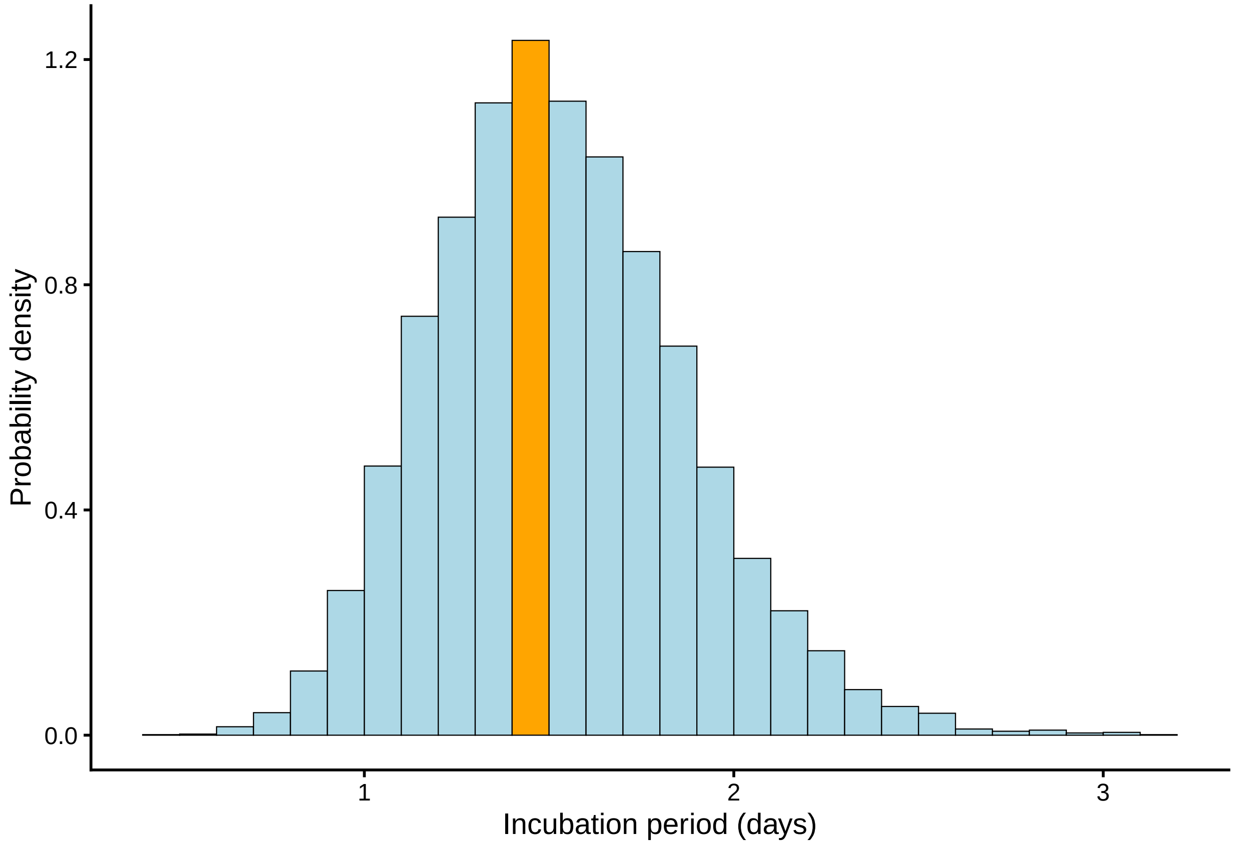


Figure 6.6: Probability distribution of incubation period with intervals of 0.1 days.

quantify the probability mass of *any* interval by applying integrals from calculus, which represents the area under the curve of the pdf over the interval of interest. The total probability over the entire sample space remains one, satisfying the basic rules of probability.

6.3.3.2 Normal distribution

Perhaps the most well-known probability density function for a continuous random variable is the **normal distribution**, also known as a **Gaussian distribution**. The normal distribution is bell-shaped and symmetric, such that it has no skew. It is widely used in science because many variables are well described by a normal distribution can be used to describe continuous variables with positive or negative values. Variables tend to have a normal distribution when the outcome is shaped by many factors each with small effect, and that characterizes a lot of variables!

As an example, consider body temperature among people with an active viral temperature. Body temperature reflects many small influences (e.g., hydration, time of day, ambient temperature), so the distribution of body temperature is often approximately normal.

For a normal random variable, the probability density of any value X is quantified as

$$f(X) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

where $f(X)$ is the probabilty density of value X, μ is the mean, and σ^2 is the variance (σ is the standard deviation) of the random variable. This is an ugly equation, but don't fret. The main takeaway here is that a normal distribution has two parameters. The mean (μ) controls the central tendency of the distribution, and the standard deviation (σ) controls the spread.

In R we can use the `dnorm` function to compute the probability density of X given values for μ and σ. For example, suppose body temperature of people with a viral infection follows a normal distribution with μ = 100.4°F and σ = 0.8°F.

Figure @ref(fig:c06_f7) shows the resulting probability density function, which we can use to highlight important characteristics of the normal distribution:

1. Symmetry around the mean. The normal distribution is bell-shaped and symmetric around the mean. In other words, the probability that body temperature is below 100.4°F is the same as the probability of body temperature being above 100.4°F (both 0.5). We can verify this with the `pnorm` function

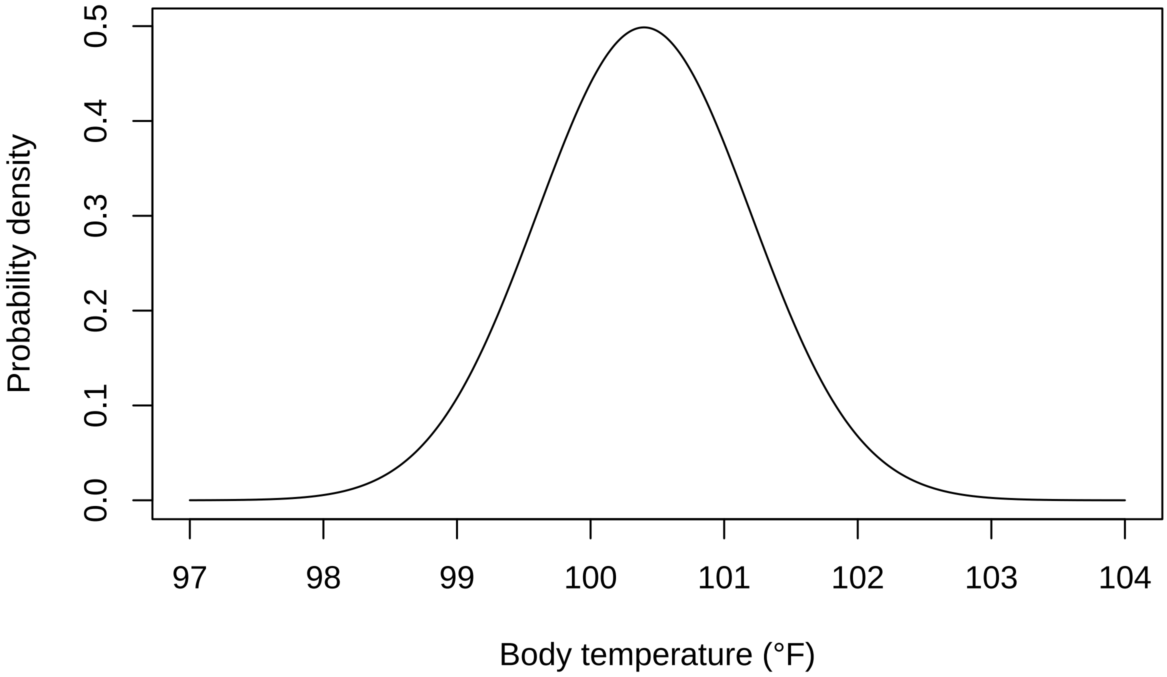


Figure 6.7: Probability density function assuming a normal distribution of incubation period mean = 100.4 and SD = 0.8 days.

```r
pnorm(100.4, mean = 100.4, sd = 0.8, lower.tail=TRUE) # P(X < 100.4)
```

```
## [1] 0.5
```

```r
pnorm(100.4, mean = 100.4, sd = 0.8, lower.tail=FALSE) # P(X > 100.4)
```

```
## [1] 0.5
```

The `pnorm` function computes probability mass for an interval as the area under the curve for that interval. We can quantify probability mass for any interval in this way. For example, what is the probability that body temperature is less than 99°F?

```r
pnorm(99, mean = 100.4, sd = 0.8, lower.tail=TRUE) # P(X < 99)
```

```
## [1] 0.04005916
```

2. The mean controls location. The mean is the expected value of the normal distribution and specifies the central tendency of the distribution. Figure 6.8 shows three normal distributions each with different means. Notice the changing μ shifts the center of the distribution but leaves its width unchanged.

3. The standard deviation controls spread. The standard deviation σ affects how variable the observations are around the mean. Figure 6.9 shows three normal distributions each with identical means but different standard deviations. Notice how the width of the normal distribution grows as the standard deviation increases.

6.3.3.2.1 Probability mass and the empirical rule Recall that we can quantify probability mass as area under the probability density function for any interval of interest. Consider again the normal distribution with $\mu = 100.4^{\circ}F$ and $\sigma = 0.8^{\circ}F$. What's the probability that body tempearture is between 99.6°F and 101.2°F? The interval of interest is shaded in the probability density function in Figure 6.10.

We can compute the probability mass with `pnorm`:

```r
pnorm(101.2, 100.4, 0.8, lower.tail=TRUE) - pnorm(99.6, 100.4, 0.8, lower.tail=TRUE)
```

```
## [1] 0.6826895
```

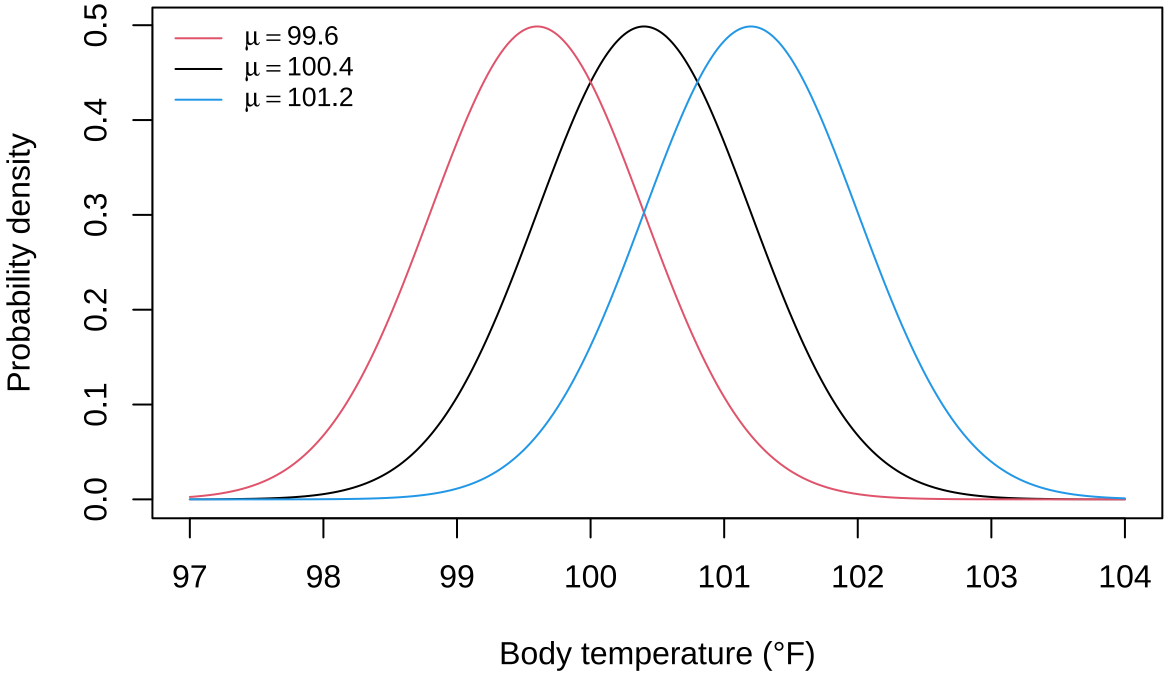


Figure 6.8: Probability density functions with varying means but identical standard deviations.

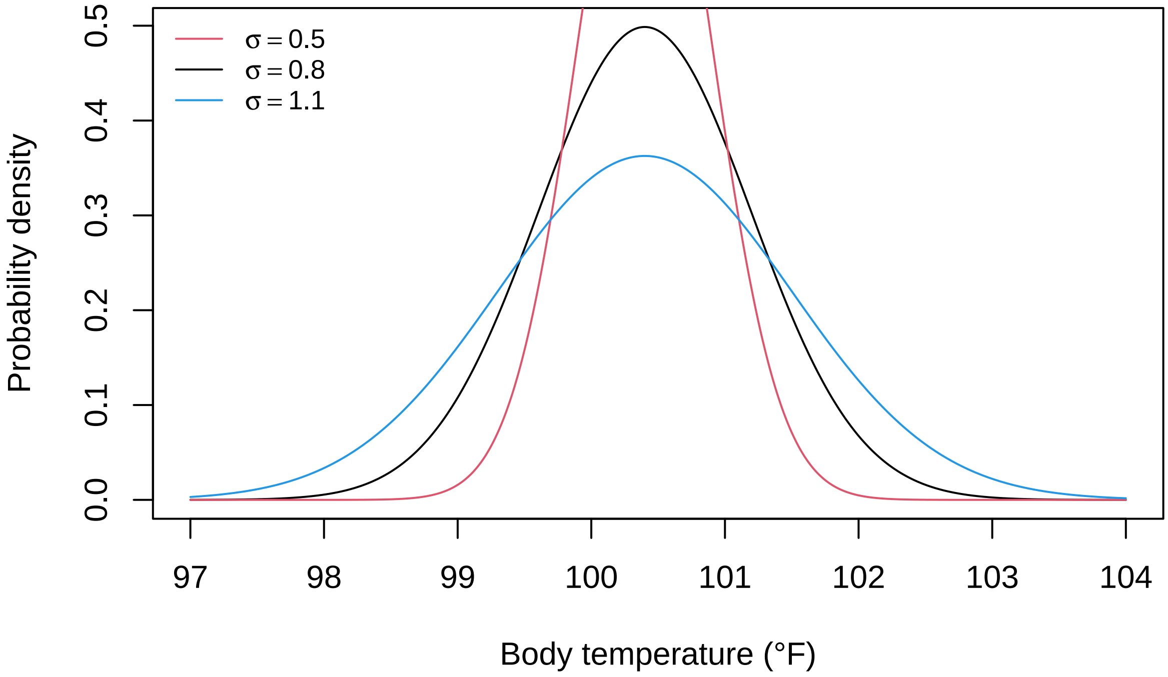


Figure 6.9: Probability density functions with identical means but varying standard deviations.

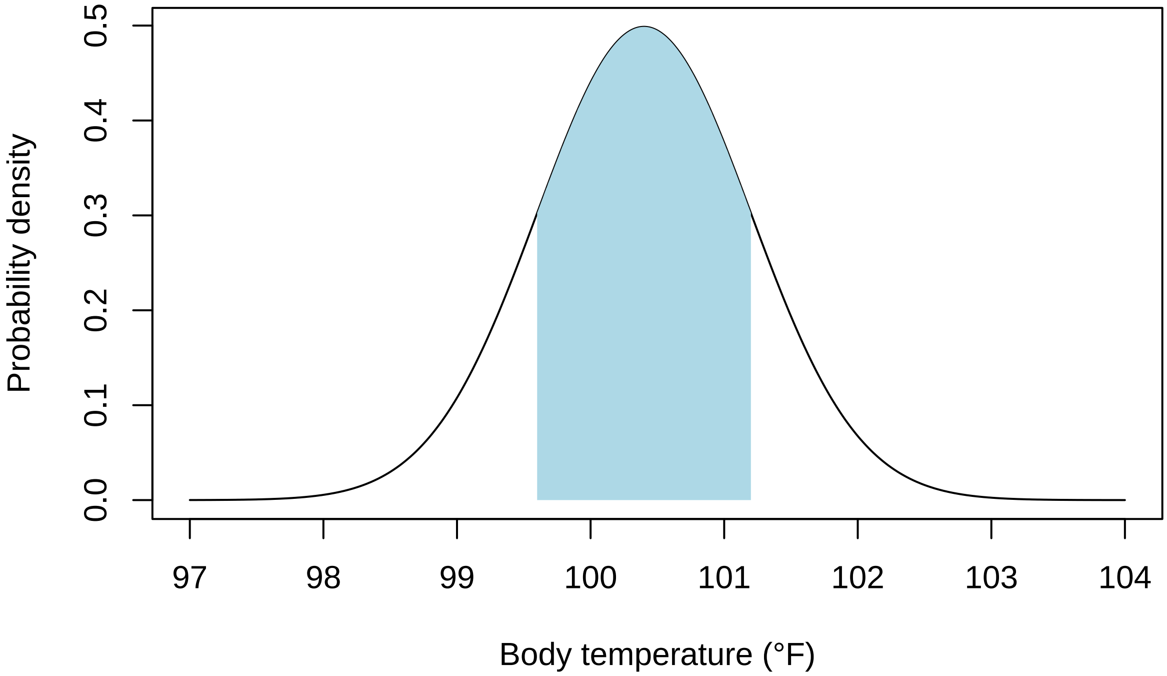


Figure 6.10: Probability density function highlighting the interval 99.6°F to 101.2°F

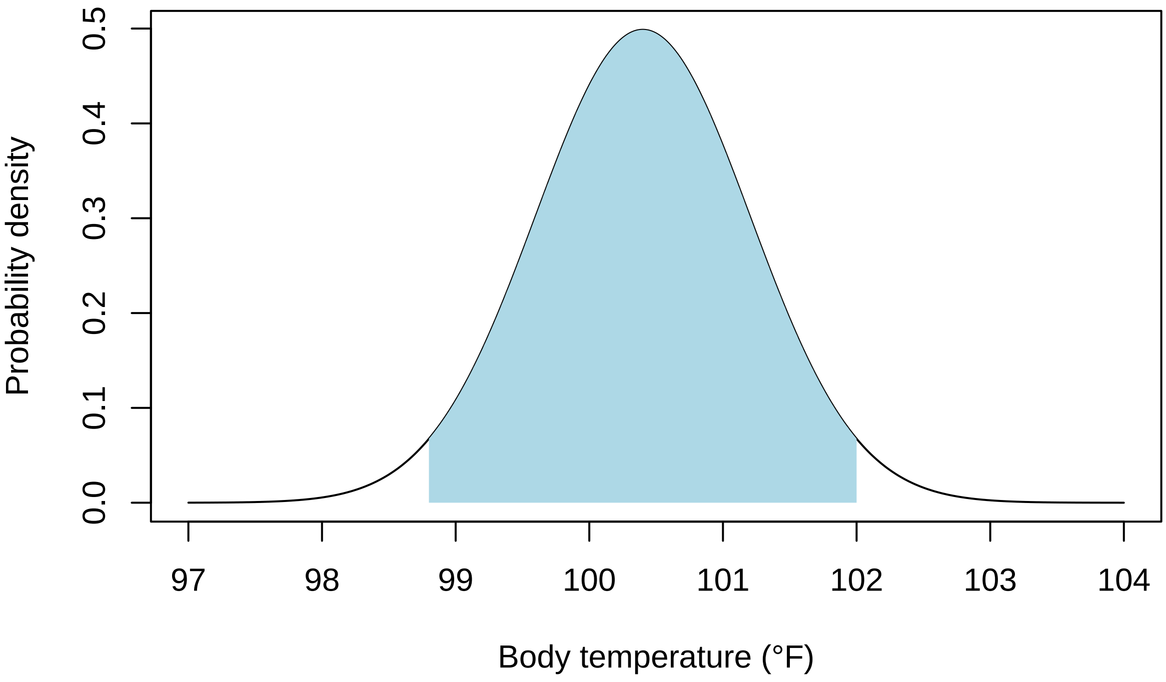


Figure 6.11: Probability density function highlighting the interval 98.8°F to 102°F

This interval encompasses exactly one standard deviation above and below the mean and includes 68.3% of the observations. Let's expand the interval to include values between 98.8°F and 102°F (Figure 6.11))?

```
pnorm(102.0, 100.4, 0.8, lower.tail=TRUE) - pnorm(98.8, 100.4, 0.8, lower.tail=TRUE)
```

```
## [1] 0.9544997
```

This interval encompasses exactly two standard deviations above and below the mean and includes 95.4% of the observations.

This illustrates what's known as the **empirical rule**. For a normal distribution, about 68% of the observations are within 1 standard deviation of the mean (i.e., $\mu \pm 1\sigma$), whereas about 95% of the observations are within 2 standard deviations of the mean (i.e., $\mu \pm 2\sigma$). Figure 6.12 shows the empirical rule graphically for our example normal distribution of body temperature.

6.3.3.2.2 Standard normal distribution Any combination of μ and σ produces a unique normal distribution, so there's an infinite variety of possible normal distributions. However, any normal distribution can be converted to a **standard normal distribution**, which is a normal distribution with $\mu = 0$ and $\sigma = 1$ (Figure 6.13). Because the standard deviation is one, the values of the standard normal distribution are in units of standard deviations. The values of a standard normal distribution are denoted Z, or $Z - scores$.

We can convert any observation X drawn from a normal distribution to a Z-score using the following formula:

$$Z = \frac{X - \mu}{\sigma}$$

Consider the normal distribution of body temperature with $\mu = 100.4°F$ and $\sigma = 0.8°F$. Let's say we have an observation X of 101.2°F. The corresponding Z-score is thus

$$Z = \frac{X - \mu}{\sigma} = \frac{101.2 - 100.4}{0.8} = 1$$

In other words, the value 101.2°F is one standard deviation above the mean. If another person has a temperature of 99.2°F, then

$$Z = \frac{X - \mu}{\sigma} = \frac{99.2 - 100.4}{0.8} = -1.5$$

The negative value of Z indicates the observation X is below the mean, so can say the observation of 99.2°F is 1.5 standard deviations below the mean.

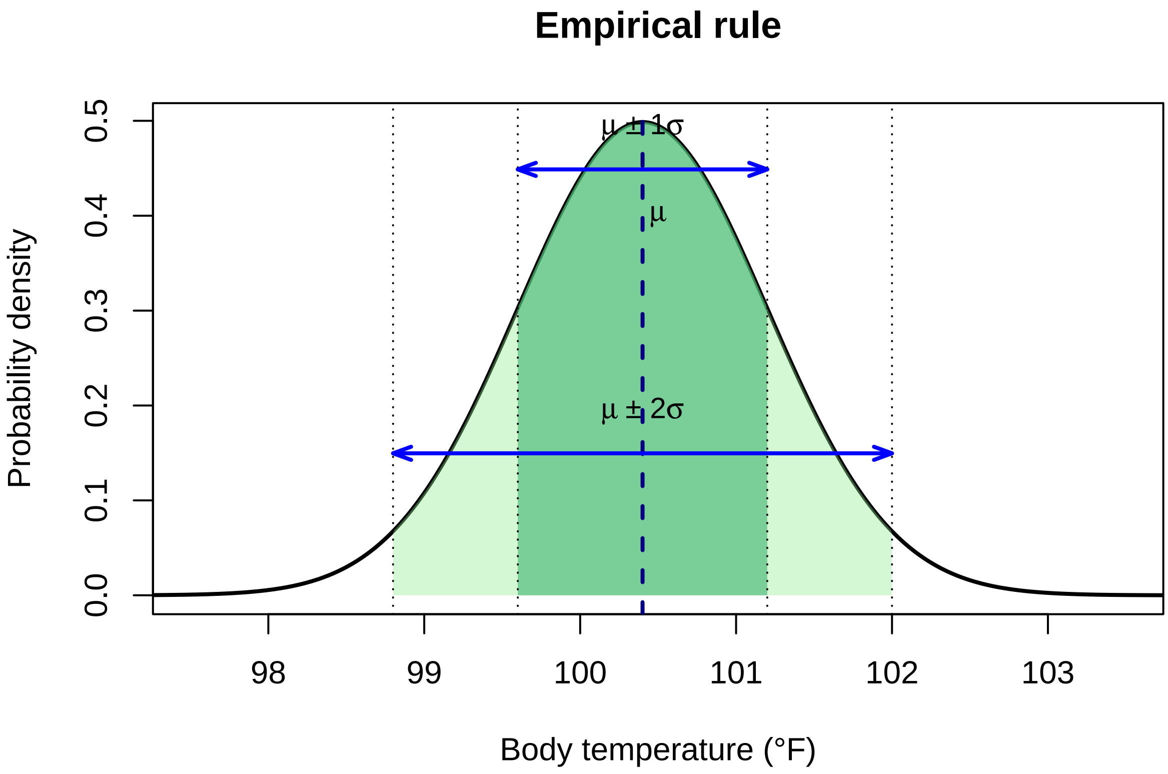


Figure 6.12: Normal probability density function showing the empirical rule, namely that about 68% of observations are within one standard deviation of the mean, and about 95% of observations are within two standard deviations of the mean. In this example, the mean body temperature is 100.4°F and the standard deviation is 0.8°F.

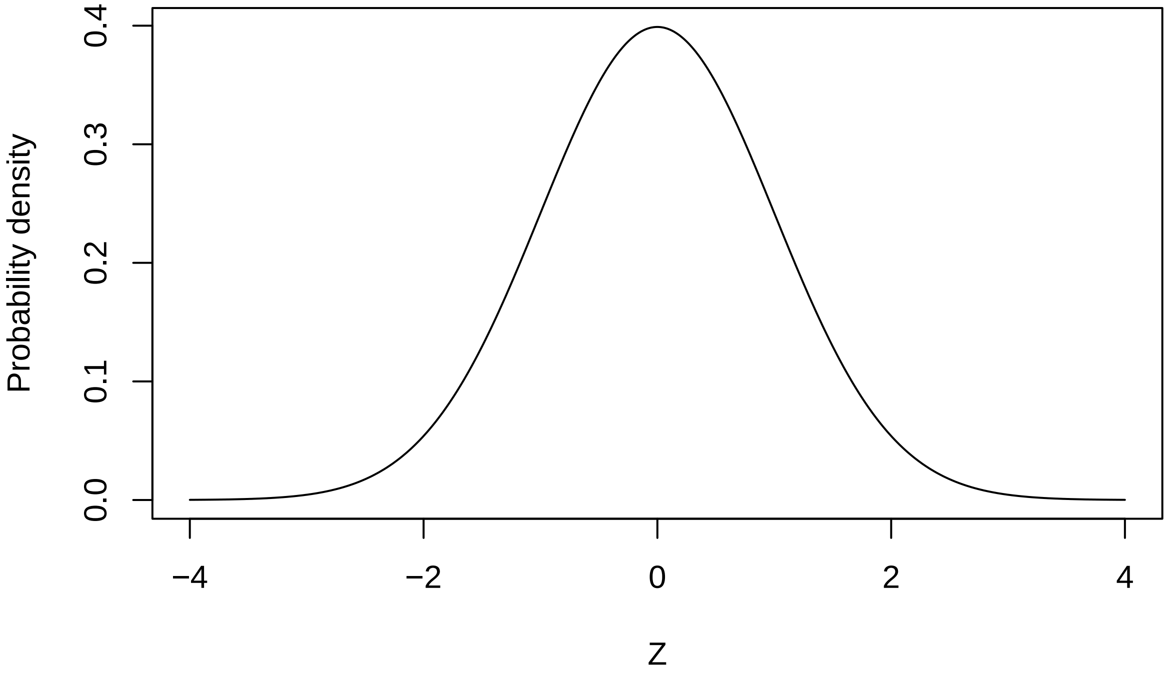


Figure 6.13: Standard normal distribution with mean = 0 and standard deviation = 1. The values Z of a standard normal distribution are in units of standard deviations away from the mean.

The standard normal distribution is perfectly symmetric around 0, so half the observations are positive and half are negative.

The standard normal distribution is useful as a common language to talk about any normal distribution, but there's also practical value that we'll encounter in statistics of converting variables to Z-scores. The process of transforming a variable to Z-scores is called **standardization**, and in R we can use the `scale` function to make the conversion. For example, here I generate a dataset of 10 values, which I then convert to z-scores:

```
x <- c(0, 1, 3, 5, 6, 6, 7, 7, 9, 10)
z <- scale(x)
z
```

```
##             [,1]
##  [1,] -1.6673586
##  [2,] -1.3585885
##  [3,] -0.7410483
##  [4,] -0.1235080
##  [5,]  0.1852621
##  [6,]  0.1852621
##  [7,]  0.4940322
##  [8,]  0.4940322
##  [9,]  1.1115724
## [10,]  1.4203425
## attr(,"scaled:center")
## [1] 5.4
## attr(,"scaled:scale")
## [1] 3.238655
```

The scale function returns each value X as a Z-score, and it also computes the mean (`scaled:center`) and standard deviation (`scaled:scale`). Note that if we standardize the value directly, we'd get the same Z-scores:

```
z <- (x-mean(x))/sd(x)
z
```

```
##  [1] -1.6673586 -1.3585885 -0.7410483 -0.1235080  0.1852621  0.1852621
##  [7]  0.4940322  0.4940322  1.1115724  1.4203425
```

6.3.4 Sampling from probability distributions

Why does any of this matter?! Well, keep in mind that when we collect data to test scientific hypotheses, we're almost always sampling from a broader population. Imagine if I'm conducting a study to estimate the mean incubation period for a viral illness. I'd love to track down every single person who has the viral illness and record the incubation, but I can't do that. Instead, we rely on sampling. We know that sampling is a stochastic, or random process. In other words, the variables that we observe should be considered *random* variables, and if we sample in an unbiased way, the values of the variable we observe through sampling are drawn from a probability distribution.

What probability distribution are observations of random variables drawn from? Often we don't know with certainty! We usually have to make some assumptions, ideally informed by knowledge of your particular discipline. I've introduced two broad types of probability distributions - discrete and random - and particular probability distribution of each type, the binomial (discrete) and normal (continuous) distributions. In reality, there are many more probability distributions one can choose from, and we'll encounter some others throughout the book, but the binomial and normal distributions provide a useful starting point. They may not be a perfect fit for a particular random variable (for example, a normal distribution allows for negative values, but incubation period can't be negative), but often these distributions can be useful approximations.

Chapter 7

Bayesian estimation and inference

With some basic probability rules under our belt, we turn our attention to the fundamental problem posed in the last chapter. Recall our goal was to estimate the prevalence of a disease in a community of 10,000. We're particularly interested in knowing if the prevalence is greater than 10%, because that's the threshold for triggering public health interventions. The population is too big to test everyone, and so a sample of 15 people were selected for testing, one of which tested positive, producing a frequentist estimate of $\hat{p}_{\text{infected}} = 0.067$. How good is this estimate, and how likely is it that the prevalence of the disease is truly less than 10%? This question implies that even though only 6.7% of the people I sampled were infected, it's still possible that more than 10% of the population is infected. And this of course is the fundamental problem of statistics: when I sample populations, the quantities I want to know are estimated and uncertain. In this chapter, we look directly at how we can use samples to estimate a quantity *and* represent our uncertainty about that quantity with the language of probability.

7.1 Scientific workflow for the problem

In Chapter 2 I introduced a general workflow outlining how we will tackle scientific research questions, specifically how we will use the data we collect to provide insight into the questions we propose. Let's remind ourselves of the steps of that workflow.

7.1.1 Theory and the research question

Our research question is "What proportion of people in the community are infected with a virus?" Although this question is descriptive in nature, there's certainly theory from the fields of infectious disease and epidemiology that informs our understanding of how viral infection occurs, how it's transmitted among people, and the approaches we can use to reliably test for infection. That theory is especially important for informing our causal assumptions about how the data are generated.

7.1.2 Generative model and estimand

A generative model describe how the data we observe are produced given our study design, and the estimand is the quantity or quantities we want to know. Having a simple descriptive research question, we can identify the estimand as the proportion of people infected with the virus. We've randomly sampled 15 people to test for the infection, which we would like to use to estimate the proportion infected. The generative model describes how the test results for samples of 15 people are produced. In other words, what are we assuming about the data generation process?

At this point we begin with a simple generative model, which I describe here as three assumptions:

- We assume that we sample individuals randomly from the population and that the probability of each individual being infected is identical to the proportion of people infected in the population. This is likely a good assumption if the sampling is truly random. It wouldn't be true if we sampled people who differed in some way from unsampled people in terms of disease risk. For example, if we relied on volunteers for testing, perhaps individuals who are vaccinated - being health consciousness - are more likely to participate in screening than individuals who are not vaccinated. But for now we assume truly random sampling.

- We assume the test results are perfectly accurate. In other words, when someone is infected with the virus, we assume the probability they will test positive is 1: $P(\text{positive result}|\text{infected} = 1)$. Similarly, when someone is not infected with the virus, we assume the probability that they will test negative is 1: $P(\text{negative result}|\text{not infected} = 1)$.

- We assume the test results for each individual are *independent*. That means the test result of one person sampled are causally unrelated to a test result of another person sampled.

7.1.2.1 Simulating synthetic data with a generative model

The neat thing about generative model is that we can use them to generate synthetic data. The process of generating synthetic data is called **simulation**. In this section I want to show you two ways you could simulate data from our generative model on test results for a viral infection.

First, the `sample` function provides a simple mechanism that can be used to generate sample data based on particular assumptions. With this function we can define the possible outcomes of the data (`pos` = positive test, `neg` = negative test), the number of people we sample for testing (`N`), and the probability of observing each outcome. In the code below, we define `p` as the proportion of people infected, and therefore according to our generative model, this is the probability of seeing a positive test. If the test is not positive, it must be negative (`neg`). We can generate data under any values of the parameters in the model, and here I've assumed a sample of 15 people from a population where 8% are infected:

```
set.seed(123)
N <- 15    #assumed sample size
p <- 0.08  #assumed proportion of population infected
sample(c("pos", "neg"), size = N, prob = c(p, 1-p), replace = TRUE)
```

```
##  [1] "neg" "neg" "neg" "neg" "pos" "neg" "neg" "neg" "neg" "neg" "pos" "neg"
## [13] "neg" "neg" "neg"
```

We see that R returns a single set of 15 test results according to the model, two being positive and thirteen being negative. So in this one synthetic dataset, the proportion infected is 2/13 = 13.3%, which is different from the true (assumed) proportion infected of 8%. This difference is due to sampling error. We could simulate datasets like this over and over to get a sense for the variation in the outcome due to sampling error. Here I use the `replicate` function to simulate 1000 datasets with `N` = 15 and `p` = 0.08:

```
set.seed(123)
N <- 15    #assumed sample size
p <- 0.08  #assumed proportion of population infected
sims <- replicate(n = 1000,
                  sample(c("pos", "neg"), size = N,
                         prob = c(p, 1-p), replace = TRUE))
sims[,1:10] #print the first 10 samples of 15
```

```
##       [,1]  [,2]  [,3]  [,4]  [,5]  [,6]  [,7]  [,8]  [,9]  [,10]
##  [1,] "neg" "neg" "pos" "neg" "neg" "neg" "neg" "neg" "neg" "neg"
##  [2,] "neg" "neg" "neg" "neg" "neg" "neg" "neg" "neg" "neg" "neg"
##  [3,] "neg" "neg" "neg" "neg" "neg" "neg" "neg" "neg" "neg" "neg"
##  [4,] "neg" "neg" "neg" "neg" "neg" "neg" "neg" "neg" "neg" "pos"
##  [5,] "pos" "pos" "neg" "neg" "neg" "neg" "neg" "neg" "neg" "neg"
##  [6,] "neg" "neg" "neg" "neg" "neg" "neg" "neg" "pos" "pos" "neg"
##  [7,] "neg" "neg" "neg" "neg" "neg" "neg" "neg" "neg" "neg" "neg"
##  [8,] "neg" "neg" "neg" "neg" "neg" "neg" "neg" "neg" "neg" "neg"
##  [9,] "neg" "pos" "neg" "neg" "neg" "neg" "neg" "pos" "neg" "neg"
## [10,] "neg" "neg" "neg" "neg" "neg" "neg" "neg" "neg" "neg" "neg"
## [11,] "pos" "neg" "neg" "neg" "neg" "neg" "neg" "neg" "neg" "neg"
## [12,] "neg" "neg" "neg" "neg" "neg" "pos" "neg" "neg" "neg" "neg"
## [13,] "neg" "neg" "neg" "neg" "neg" "neg" "neg" "pos" "neg" "neg"
## [14,] "neg" "neg" "neg" "neg" "neg" "neg" "pos" "neg" "neg" "neg"
## [15,] "neg" "neg" "neg" "neg" "neg" "neg" "neg" "neg" "neg" "neg"
```

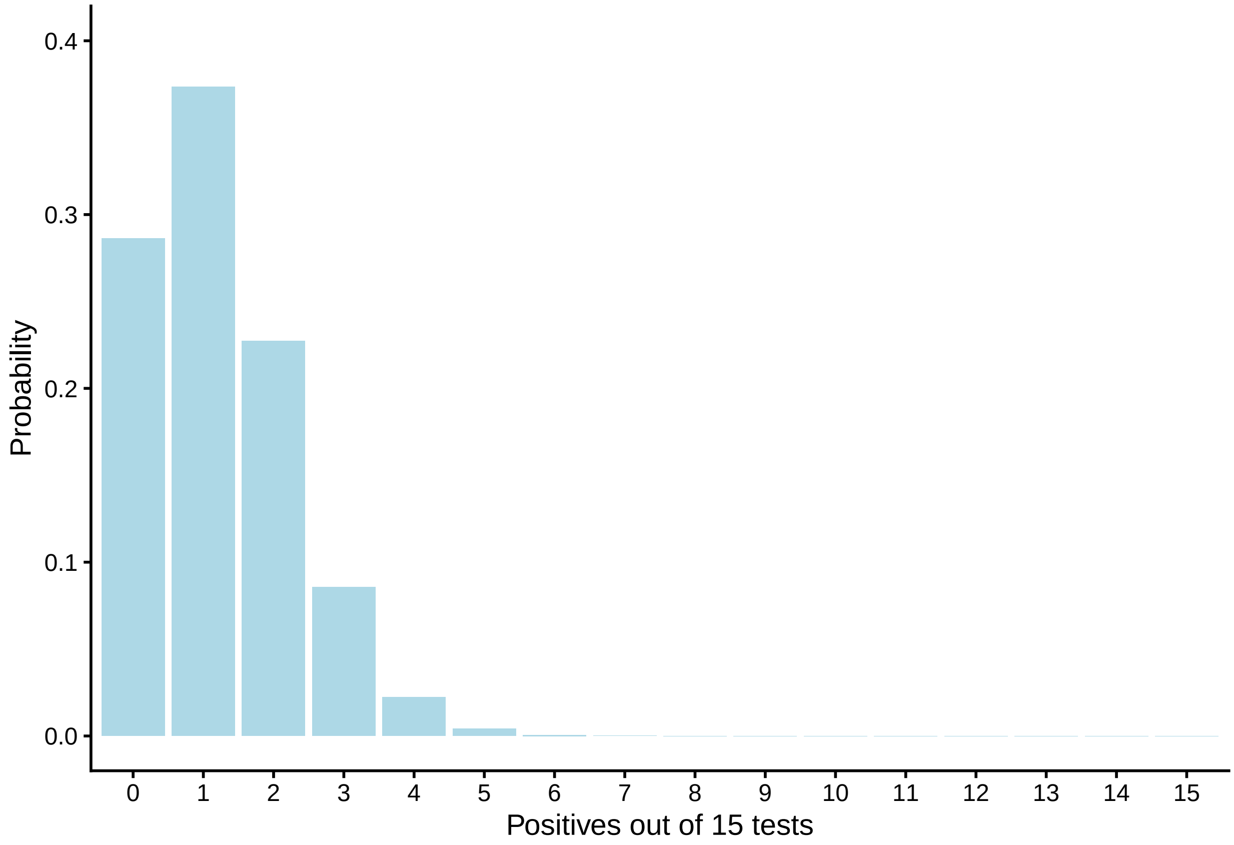
I've printed the first 10 simulated datasets, with columns representing the simulation and rows representing the 15 outcomes in each simulation. We can see the first simulation has two positives, the second has one positive, the third has 0 positives, and so on. Let's compute the frequency distribution of the number of positives across all 1000 simulated datasets:

```
positives <- colSums(sims == "pos")
table(positives) #frequency table
```

```
## positives
##   0   1   2   3   4   6 
## 314 368 214  89  13   2
```

We see the most common outcomes are 0, 1, or 2 positives out of 15 when the true proportion infected is 0.08, with the probability of an increasing number of positives declining sharply as shown in the graph above. This makes sense given that we assumed the proportion infected in the population was only 8%. The figure below shows the distribution of outcomes across 1000 synthetic datasets when we assume a much larger prevalence of 30%. When the prevalence is 30%, it's most likely to see three to six positives out of 15, more than what we expect when the prevalence is only 8%, but still quite variable due to sampling error.

A second way we can simulate data - and this will be the approach we use throughout the book - is to use probability distributions. This is one of the reasons we spent an entire chapter working with probability rules and distributions. Recall in the last chapter that we described the number of positives X out of out of n trials as a random variable following a binomial distribution. The binomial distribution makes the same assumptions that we articulated in our generative model, namely a constant probability of infection (implying random sampling and a perfectly accurate test) and independent observations. Indeed, we used the binomial distributiont o compute the probability of observing any number of positive tests out of 15, in that case when we assumed a prevalence of 5%. The graph below shows the binomial distribution of positive test results out of 15 when the prevalence is 8%:



We can also use the binomial distribution to *simulate* data. This can be done with the `rbinom` function. Here I use the `rbinom` function to simulate a single sample (`n = 1`) of 15 individuals `size = 15` where the probabability of success (infection) is 0.08:

```r
set.seed(123)
rbinom(n = 1, size = 15, prob = 0.08)
```

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

The `rbinom` function returns the total number of "successes" (X), which we define as positive test results, out of the 15. Just like the `sample` function, we can simulate many datasets. The code below produces 10 datasets of size 15:

```
set.seed(123)
rbinom(n = 10, size = 15, prob = 0.08)
```

```
##  [1] 1 2 1 2 3 0 1 3 1 1
```

We could of course do this 1000 times just like we did with the `sample` function and compute the frequency distribution of the number of positive results out of 15:

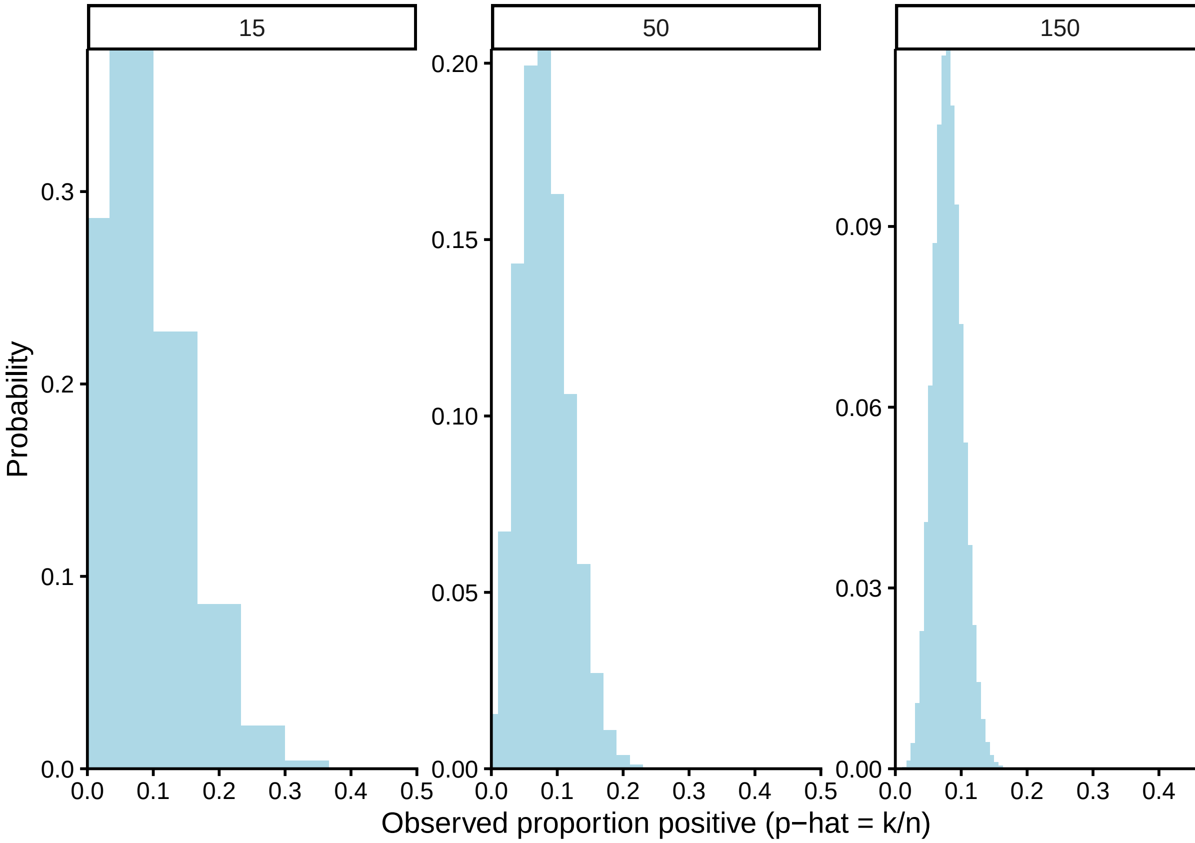
```
set.seed(123)
sims <- rbinom(n = 1000, size = 15, prob = 0.08)
table(sims)
```

```
## sims
##   0   1   2   3   4   5   6 
## 286 376 228  84  24   1   1
```

We see a very similar distribution of positive results in comparison to the `sample` function, which makes sense because the assumptions we made in the `sample` function were the assumptions of the binomial distribution. One of the nice things about using the probability distribution directly, however, is that it gives us the *exact* probability of each possible outcome given our assumptions.

The last thing I want to show you is that a generative model can give you insight into the quality of the data you can expect from your sampling design, specifically accuracy and precision. For example, assume the true prevalence is 8%, but we're considering how many tests we should invest in to estimate the proportion infected. We can use the binomial distribution - our generative model - to look at how the sampling distribution changes with sample size. For example, suppose we want to evaluate the implications of sampling 15, 50, or 150 people. The figure below shows the probability of observing a particular proportion of positive test results based on each sample size:

```
set.seed(4)

p <- 0.08
ns <- c(15, 50, 150)

d <- do.call(rbind, lapply(ns, function(n){
  k <- 0:n
  phat <- k / n
  data.frame(
    n    = factor(n, levels = ns),
    phat = phat,
    prob = dbinom(k, size = n, prob = p),
    xmin = pmax(0, phat - 0.5/n),
    xmax = pmin(1, phat + 0.5/n)
  )
}))

ggplot(d) +
  geom_rect(aes(xmin = xmin, xmax = xmax, ymin = 0, ymax = prob),
            fill = "lightblue", color = NA) +
  facet_wrap(~ n, nrow = 1, scales = "free_y") +
  scale_x_continuous(limits = c(0, 0.5), expand = c(0, 0)) +
  scale_y_continuous(expand = c(0, 0)) +
  labs(
    x = "Observed proportion positive (p-hat = k/n)",
    y = "Probability"
  ) +
  theme_classic()
```



You can see that as the number of individuals samples increases, the width of the sampling distribution decreases, meaning you can expect more precise estimates with larger sample sizes. Note that in each scenario, you can get an estimated prevalence of >10% even though the true prevalence is 5%, but the likelhood of that results decreases as sample size increases. Also note that sample size has no effect on accuracy, as each of the distributions is centered on the true prevalence that we assumed: 8%.

7.1.3 Statistical model and estimate

Whereas the generative model expresses our causal assumptions about how the data are estimated and can be used produce simulated datasets, the statistical model is what we use to *estimate* what we want to know in the target population. The statistical model needs to take the 15 test results we actually observe and use them to estimate the proportion infected in the population of 10,000. Remember we observed one person out of 15 tested positive, so

Table 7.1: Joint and marginal probabilities for mammogram screening tests

Disease Status	Positive Test	Negative Test	Total
Breast Cancer	0.0087	0.0013	0.01
No Breast Cancer	0.1089	0.8811	0.99
Total	0.1176	0.8824	1.00

you might be tempted to say that the estimate of the prevalence is simply $\widehat{p}_{infected} = \frac{1}{15} = 0.067$. And if you were strictly using a frequentist definition of probability, you would be correct. Frequentist estimates are **point estimates**, a single best estimate for the quantity we're interested in. One problem with frequentist point estimates is that we can't talk about them probabalitically. In other words, we can't answer the question "What is the probability that the true prevalence of the infection in the population is 6.7%". Nor can we talk about the probability of the prevalence taking on any other values. Remember that our main goal is to determine if the prevalence of the infection is above 10%, becuase that's when interventions are triggered. In other words, we want to know "What is the probability that the infection prevalence is >10%?". Frequentist point estimates can't answer that question, but fortunately there is a solution, and that solution is to use a Bayesian interpretation of probabiglity. So in the remainder of this chapter, I will introduce you to Bayesian probability, and then show you how we can use Bayes to estimate quantities with statistical models.

7.2 Bayes Theorem

Step away from disease prevalence for a moment and consider medical screening tests. Medical screening tests are used to identify health issues in people before symptoms are present. In the 1980s the American Cancer Society started recommending annual or semi-annual mammograms for women beginning in middle age to look for breast cancer. In contrast to our toy example that assumes a perfect test for a viral infection, most medical testing is imperfect. There is some chance an individual will test negative when they truly have a disease (**false negative**), and there is some chance that an individual will test positive when they don't have a disease (**false positive**).

Table 7.1 shows the joint and marginal probabilities for mammogram screening tests (Lehman et al. 2016). The joint probabilities are shown for test results (positive and negative) and disease status (does or does not have breast cancer). For example, we can see that 0.87% of individuals have breast cancer and test positive.

How well does the test perform? From the joint probabilities in Table 7.1, we can quantify the probability of test outcomes conditional on disease status. For example, what is the probability the test will be positive when a person has the disease? This probability is known as the **sensitivity** of a test in medical research. We know from the rule of conditional probability that the sensitivity can be quantified as

$$P(\text{Positive}|\text{Cancer}) = \frac{P(\text{Positive and Cancer})}{P(\text{Cancer})}$$

We can see in Table 7.1 that the joint probability of testing positive and having breast cancer is 0.0087, and the marginal probability of breast cancer is 0.01. Thus, the sensitivity of the test is

$$P(\text{Positive}|\text{Cancer}) = \frac{0.0087}{0.01} = 0.87$$

In other words, when someone has breast cancer, they have a 0.87 probability of testing positive in a screening mammogram. This means they have a 0.13 probability of testing negative, so 0.13 is the false negative rate for the test.

How well does the test return a negative result when an individual doesn't have breast cancer, an outcome referred to as **specificity** in medical research. Again, we can specify the conditional probability of interest as:

$$P(\text{Negative}|\text{No Cancer}) = \frac{P(\text{ Negative and No Cancer})}{P(\text{No Cancer})} = \frac{0.8811}{0.99} = 0.89$$

We see the specificity is 0.89, meaning the probability of a false positive is 0.11.

Now as you might surmise, medical doctors are faced with a conundrum when interpreting the results of these kind of screening tests. Clearly we can see from the estimation of sensitivity and specificity that the screening test is imperfect. Sometimes it returns a positive for people who don't have breast cancer, and sometimes it returns a negative for people who do have breast cancer. Doctors are well aware of this. Before approving medical tests for general use, regulatory agencies evaluate the performance of tests as measured by sensitivity and specificity (along with potential side effects), so these values are known by practitioners. The question we need to grapple with is how doctors should interpret the resutls of an imperfect screening test. What does a test result imply about the probability of actually having the disease?

When a routine mammography is performed on an individual, what information is known about that person with respect to the likelihood of having breast cancer? For screening tests on individuals without a family history of breast cancer, there is very little information to go on. In such cases, the probability of the individual having breast cancer might be estimated best by the prevalence of breast cancer among individuals who are screened. As we can see in Table

7.1, that's the marginal probability of breast cancer which is 1%. In a Bayesian analysis, this marginal probability of breast cancer is the **prior probability**, namely the probability of the event of interest (having breast cancer) before we collect new data.

How does the probability of having breast cancer change when we learn new information, specifically the result of the mammography? Suppose a person has a positive test. What is the probability this person has breast cancer, given that they have a positive mammogrpahy? Some people would be tempted to say the probabilty is 87%, but that isn't right. The probability 87% represented the probability of testing positive *given* a person has breast cancer. What we want is the reverse conditionality. What is the probability of having breast cancer, *given* the positive test result?

This is where Bayesian thinking becomes valuable. Thomas Bayes discovered that a conditional probability could be quantified (in part) from the *reverse* conditional probability. Consider the conditional probability of interest, which we can frame using our standard rule to quantify conditional probability:

$$P(\text{Cancer}|\text{Positive Test}) = \frac{P(\text{Cancer and Positive Test})}{P(\text{Positive Test})}$$

Notice that the numerator is a joint probability. We know from the general multiplication rule that the joint probability of A and B is $P(A|B) * P(B)$, which can be equivalently expressed as $P(B|A) * P(A)$. Bayes showed this rearrangement and expressed the conditional probability in the following way, which is known as **Bayes Theorem**:

$$P(\text{A}|\text{B}) = \frac{P(\text{B}|\text{A}) * P(\text{A})}{P(\text{B})}$$

Applying Bayes Theorem to the breast cancer example, the probability of breast cancer given a positive test becomes:

$$P(\text{Cancer}|\text{Positive Test}) = \frac{P(\text{Positive Test}|\text{Cancer}) * P(\text{Cancer})}{P(\text{Positive Test})}$$

All we have done here is re-expressed the joint probability of breast cancer and positive test result as the sensitivity of the test (probability of true positive) and the prevalence of the disease. This is an important insight because those probabilities are generally more intuitive than joint probabilities (the probability of randomly selecting an individual who jointly has breast cancer and tests positive).

Given this conceptual background on Bayes Theorem, let's now compute the probability that the medical wants to know, namely the probability of having breast cancer given a positive test result:

$$P(\text{Cancer}|\text{Positive Test}) = \frac{P(\text{Positive Test}|\text{Cancer}) * P(\text{Cancer})}{P(\text{Positive Test})} = \frac{0.87 * 0.01}{0.1176} = 0.074$$

To most people this is a surprising result! In the case of a standard screening test for breast cancer, the probability of actually having breast cancer given a positive test is only 7.4%. Let's be clear about the logic logic of our approach. First, before we had any new information from the mammogram, we stated the *prior probability* of an individual having breast cancer is 1%. That's simply the prevalence of the disease among those who are being screened for breast cancer. Bayesian inference takes that prior knowledge and updates it with new information, namely the result of the test. Given the new information of a positive test, the updated probability of breast cancer increases to 7.4%. This updated probability, conditional on the new data, is called the **posterior probability**. This is the essence of Bayesian inference: start with your prior knowledge expressed as a prior probability, and update it based on new data.

Why is the probability of breast cancer conditional on a positive test so low? Bayes Theorem makes this clear. Although the sensitivity and specificity of mammograms are quite high, the mammogram is not a perfect test. Some of the positive tests are true positives, and some are false positives. What Bayes Theorem shows us is that we have to take into consideration the facts that 1) the prevalence of breast cancer is relatively low at 1%, and 2) only a portion of the positive test results are true positives. The numerator of Bayes Theorem gives us the proportion of individuals with breast cancer and positive tests, and then we divide by the total probability of positive tests to arrive at the proportion of positive tests that are true cases of breast cancer. More intuitively, Bayes Theorem tells us that because the prevalence of breast cancer is low and the mammogram produces false positives, the vast majority of the positive test results are actually coming from people who don't have breast cancer.

This particular application of Bayes Theorem demonstrates how a low base rate - or prevalence - of a disease significantly influences the conditional probability of having the disease based on a test result. Even when sensitivity is high, low prevalence of the disease means that positive tests from screening programs are mostly false positives. This is known in the medical literature as the **base rate fallacy**, in which practitioners mistakenly believe high sensitivity implies high likelihood of a disease for a positive test.

A critical assumption in the example to this point is that we don't know anything about the person being tested with respect to their likelihood of having the disease. That's why the prevalence of the disease was used as the prior knowledge, which implies that individuals were being tested at random. Had the test been motivated by particular symptoms, we would want to revise the prior probability to reflect that knowledge.

For example, imagine a patient has symptoms that raises the suspicion of breast cancer to 50%. We can easily update our computation based on this information.

If the probability of breast cancer for this individual (prior to testing) is 50%, then the probability of not having breast cancer is also 50%, and the total probability of a positive test is

$$P(\text{Positive}) = P(\text{Positive} \mid \text{Cancer})\, P(\text{Cancer}) + P(\text{Positive} \mid \text{No Cancer})\, P(\text{No Cancer})$$
$$P(\text{Positive}) = 0.87(0.5) + 0.11(0.5) = 0.435 + 0.055 = 0.49$$

Now we update our computation of the conditional probability of having the disease based on a positive test:

$$P(\text{Cancer}|\text{Positive Test}) = \frac{P(\text{Positive Test}|\text{Cancer}) * P(\text{Cancer})}{P(\text{Positive Test})} = \frac{0.87 * 0.50}{0.49} = 0.89$$

We see, unsurprisingly, that when symptoms raised our suspicion of breast cancer to a coin flip, the probability of breast cancer given a positive test is now 89%. This dramatic change illustrates how prior knowledge - expressed quantitatively as the prior probability - can have a dramatic impact on how we interpret the results of new evidence.

Broadly speaking, the example above highlights how Bayesian inference incorporates prior knowledge *and* new data to estimate quantities of interest. Whether you are a doctor interpreting test results or a scientist testing a hypothesis, Bayes Theorem allows us to specify our initial assumptions based on prior knowledge and update our beliefs with new empirical evidence.

7.3 Applying Bayes Theorem to statistical analysis

In the context of statistical analysis, we can think about Bayesian inference as allowing us to estimate quantities that represent a scientific hypothesis by combining our prior knowledge about the hypothesis with new data from a study:

$$P(\text{Hypothesis}|\text{Data}) = \frac{P(\text{Data}|\text{Hypothesis}) * P(\text{Hypothesis})}{P(\text{Data})}$$

There are four key parts of Bayes Theorem being represented here:

- **Prior probability**, $P(Hypothesis)$: Quantitative statement of your degree of belief about the hypothesis prior to collecting new data.

- **Likelihood**, $P(Data|Hypothesis)$: Probability of new data you observe given that a hypothesis is true.
- **Total probability of the data**, $P(Data)$: The overall probability of the data integrated across all possible hypotheses. This is quantified using the total law of probability, where P(Data) represents a weighted average of the data, weighing the probability the data in accordance to the prior probability of each possible hypothesis. This is sometimes called the **marginal likelihood** (as it is a marginal probability integrated across hypotheses) or the **average probability of the data** (as it is a weighted average). Sometimes it is simply called the **evidence**. We won't dwell on this component too much, as it more or less provides a mathematical purpose for us of standardizing the posterior probability such that it integrates to one.
- **Posterior probability**, $P(Hypothesis|Data)$: This is our updated probability of the hypothesis given the new data.

This is fundamentally different than frequentist inference, where the P-value for a single null hypothesis represents the probability of data (or more extreme values) given the null hypothesis is true. We use that information as part of hte likelihood, but critically, we combine the likelihood with our prior ($P(Hypothesis)$) to compute the conditional probability that we are actually interested in, namely the posterior probability of the hypothesis given the data.

Remember the whole point of science is to evaluate ideas with empirical data. We've defined the estimand as the (unknown) quantity that we want to estimate in the target population. It's the quantity that will provide insight into the research question. So rather than referring to a verbal "Hypothesis" in Bayes theorem, instead we make it entirely quantitative by representing different hypotheses as different values of the *estimand*, which we denote θ. Making this substitution, Bayes theorem can be stated as

$$P(\theta|\text{Data}) = \frac{P(\text{Data}|\theta) * P(\theta)}{P(\text{Data})}$$

For example, if we want to compare the difference in the mean number of fish between polluted and unpolluted areas in a lake, then θ could represent the difference between the means in polluted and unpolluted areas, which could take on any real value. Bayes theorem provides a method for leverating our prior knowledge and new data to compute the probaiblity that the difference in the means takes on *any* of those values. In other words, we can compute the posterior probability that the difference in means takes on any value, such that teh posterior probability is an entire distribution of parameter values. In Bayesian inference we can use the posterior probability distribtion to reason probabilistically about hypotheses comparing the probability of parameter values that represent different hypotheses. For example, if the estimand is the the difference in mean fish between polluted and unpolluted areas computed as $\mu_{unpolluted} - \mu_{polluted}$, we can compare the probability that the difference is positive - values that represent fish abundance being greater in unpolluted areas - to the probability that the difference is negative - values that represent fish abundance being greater in polluted areas.

7.4 Steps of estimation with Bayesian inference

In the remainder of this chapter we will apply Bayesian inference to estimate the prevalence of a viral infection when we randomly sample N = 15 people for testing. Before we take a look at each step of Bayesian estimation, it is worth emphasizing that the estimates produced by Bayesian inference are entire distributions. This is in stark contrast to frequentist estimation, which produces a point estimate of the parameter of interest. For example, recall that we found 1 positive tests out of N = 15 randomly sampled individuals. The frequentist estimate of the proportion infected is simply

$$\hat{p}_{infected} = \frac{1}{15} = 0.067$$

We know that a proportion can take on any value continuously distributed between (and inclusive of) 0 and 1, but frequentist estimation effectively arrives at a single value. The output of a Bayesian analysis is the posterior distribution, which is a probability distribution for estimand. The posterior probability distribution *is* the estimate in Bayesian inference, and because of this fact, we can evaluate the probability of any value the parameter might take on. And that's exactly what we need! Remember how public health interventions will be implemented if the prevalence of the disease is greater than 10%. Bayesian inference will let us compute the probability that the prevalence is >10%. Let's take a look at the steps of Bayesian estimation:

7.4.1 Specify the prior distribution

The first thing to do in a Bayesian analysis is to specify the prior distribution for the parameters being estimated. In our example, we have only a single parameter: the proportion of people infected. The goal here is to represent our prior knowledge about the parameter values by probability distributions. This can be a complicated task, and in this book I will present only a surface level treatment of specifying prior distributions. For now, we will start with the simplest possible prior distribution, the **uninformative prior**.

Uninformative priors represent a situation where you no prior knowledge about the parameters of interest. For parameters like proportions that have a bounded range (between 0 and 1), the most common way of specifying an uninformative prior is with the **uniform distribution**. The uniform distribution simply says that the probability density of all values X between values a and b are identical ($X \sim \text{Uniform}(a, b)$. Because we are working with a proportion, we would say the prior probability of the proportion infected follows a uniform distribution between 0 and 1:

$$p_{infected} \sim \text{Uniform}(0, 1)$$

In R we can compute the uniform distribution with the `dunif` function. Before we do so, however, we need to select a set of parameter values at which we will compute the prior distribution, and ultimately the posterior distribution. The problem of course is that there's an infinite number of possible parameter values, even in cases like ours where the parameter is bounded between a minimum and maximum. Because of this, we proceed with an approximation of the prior and posterior distributions by selecting a set of values that covers the plausible range of the posterior distribution. This method is called **grid approximation**. In this example, I will create a grid of possible values for the proportion infected from 0 to 1 by every 0.01 units and call it `p.grid`.

The resolution of the grid trades off completeness of the probability distribution with computational efficiency. In other words, if the grid is sparse (e.g., every 0.1 units), the computation is equite efficient, but the resulting probability distributions will not be approximated precisely. If the grid is too dense (e.g., every 0.0000001 units), the probability distributions will be very precise but might take a long time to compute. By specifying a resolution of 0.01 units, I'm trying to balance precision with computational efficiency.

Once the grid of possible values of the parameter is created, the `dunif` function is used to compute the prior probability distribution, `p.prior`.

We see in Figure 7.1 that the prior probability distribution for the prevalence is completely flat. Indeed, the uniform prior is often called a **flat prior**. Because the the probability density is identical at all values of the proportion infected, this prior does a good job of representing our complete lack of prior knowledge.

7.4.2 Quantify the likelihood of the data

Once the prior distribution is specified, we can go ahead and compute the likelihood, which is the probability of the data for each possible value of the parameter. Recall that we observed 1 positive out of 15 tests. The likelihood is simply the probability of this specific observation under different values of prevalence. Earlier we talked about how we can compute these probabilities with the binomial distribution, which makes the same assumptions as our generative model. For example, what's the probability of 1 out of 15 positive tests when the prevalence of the infection is 0?

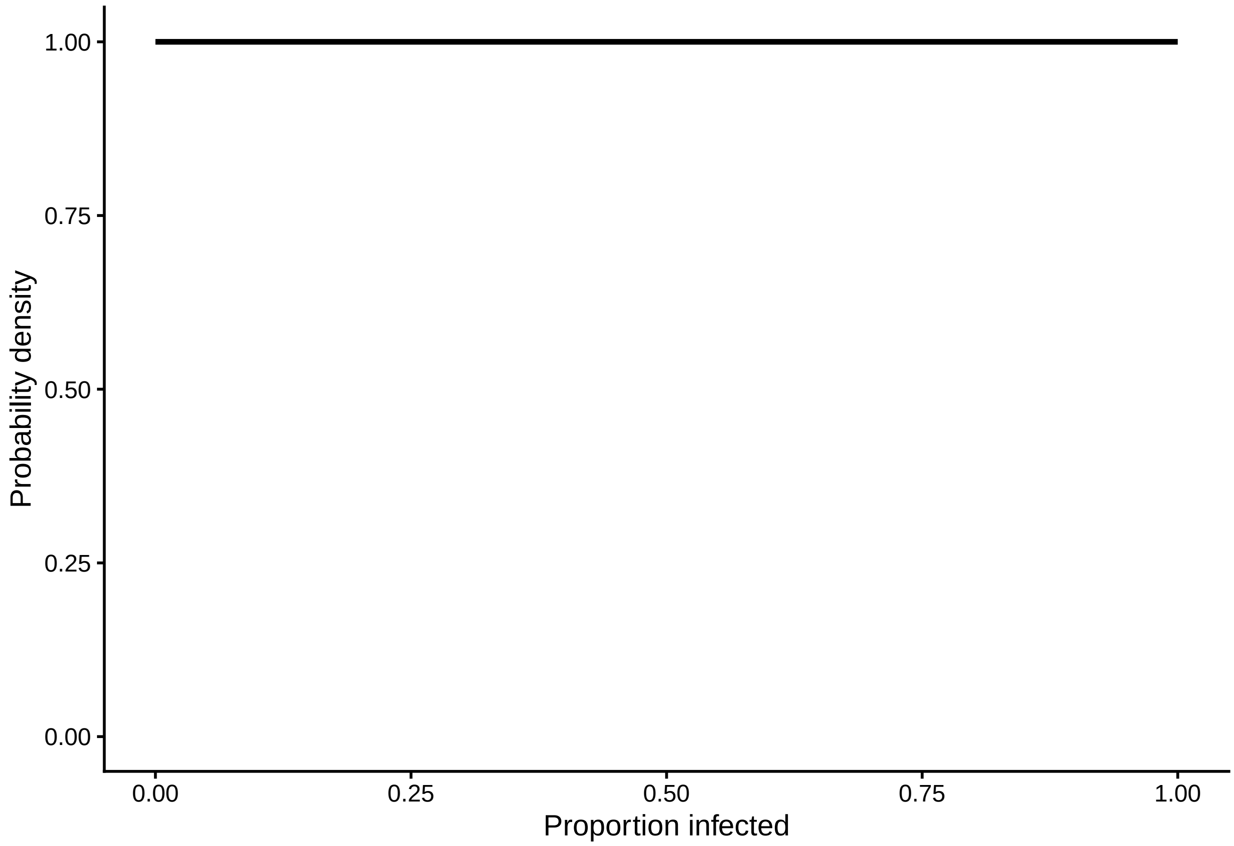


Figure 7.1: Uniform prior distribution of the proportion infected.

```
dbinom(x = 1, size = 15, prob = 0)
```

```
## [1] 0
```

We see the probability of 1 out of 15 positives is 0 when the true prevalence is 0. That makes sense! If the disease does not occur at all, then no one should be infected. But that's just one possible value of the prevalence. Let's take a look at the next possible value in the grid, where the prevalence is 0.01:

```
dbinom(x = 1, size = 15, prob = 0.01)
```

```
## [1] 0.1303119
```

So when the infection prevalence is 1%, there's a 13% chance of observing 1 out of 15 positives. Ultimately we need to go ahead and compute these likelihoods for every value of prevalence in the grid. We can do this by specifying our grid of values, `p.grid` as the `prob` argument in the `dbinom` function, saving the output as `lik` in our dataframe `d`.

Figure 7.2 shows the likelihood distribution. Each point in the graph shows the likelihood of the observed data for each value of proportion infected in the grid. We can clearly see that the observed data are most likely when the proportion infected is around 0-25%. But keep in mind that the likelihood is not the distribution we ultimately want. These are the probabilities of the observed data conditional on each value of the prevalence. What we really want is the posterior probability of the prevalence conditional on the data. But before we can compute that, we just need one more step.

7.4.3 Quantify the total probability of the data (marginal likelihood)

The last step before computing the posterior distribution is to quantify the marginal likelihood, which the total probability of the data. Using the total law of probability, we need to take the product of the likelihood (the probability of the observed data conditional on each value of the prevalence) and the prior probability of each value of the parameter, then sum those values up.

```
d$marg.lik <- d$lik*d$prior
```

Again, the marginal likelihood isn't very interesting on its own. We compute it so that we can standardize the posterior distribution, making it integrate to 1.

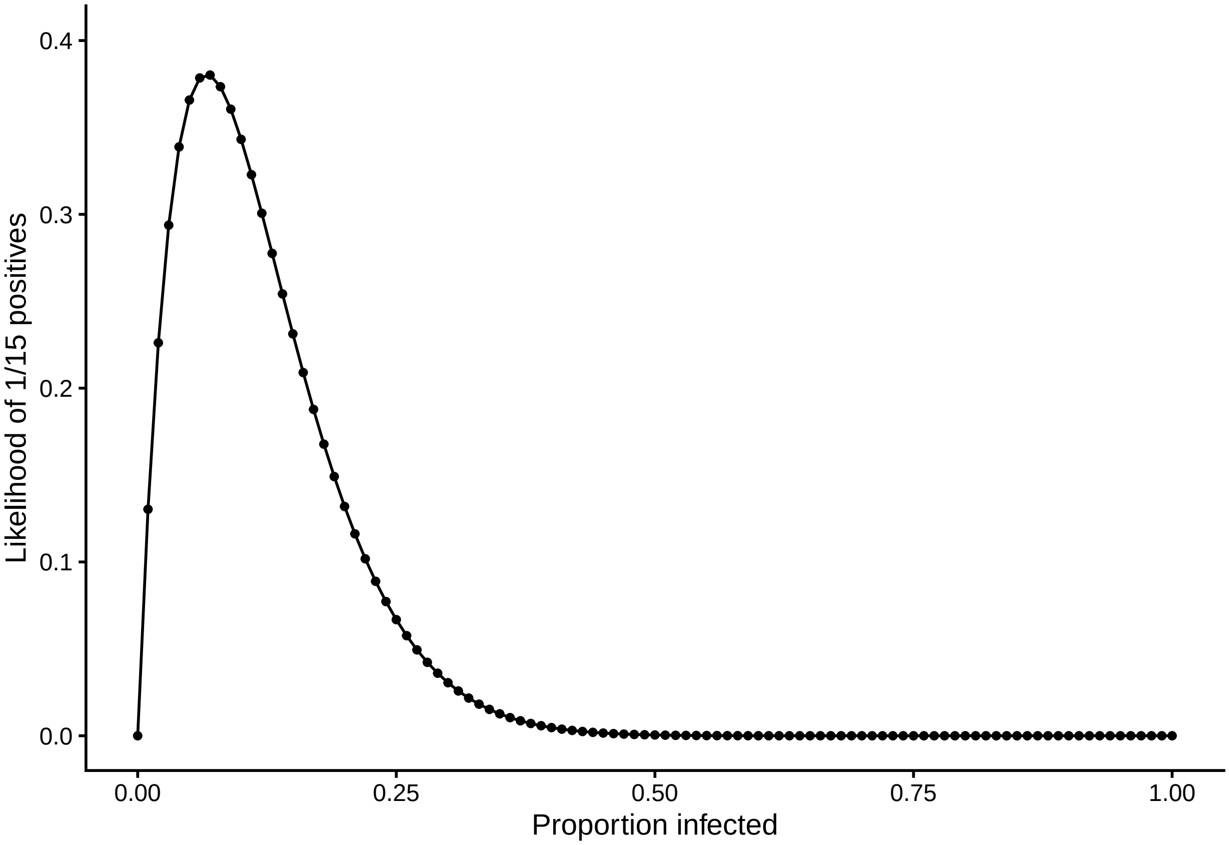


Figure 7.2: Likelihood distribution showing the probability of the observed data (1 out of 15 positive tests) conditional on each value of the prevalence of infection.

7.4.4 Quantify the posterior distribution

Finally we are ready to compute the posterior probability distribution, namely the probability of each possible value of the prevalence given the data we observed. All we need to do here is apply Bayes Theorem:

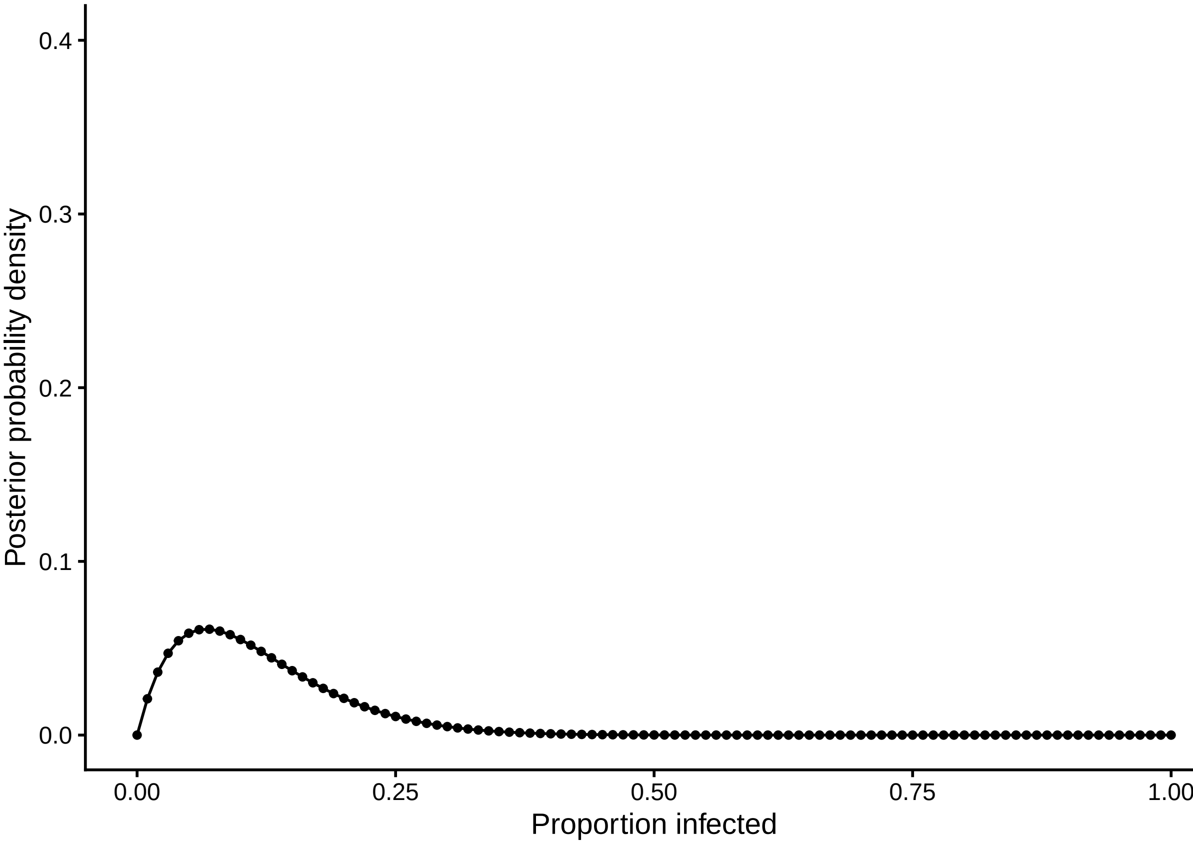


Figure 7.3: Posterior probability distribution for the infection prevalence conditional on the observed data (1 out of 15 positives).

There we have it. Figure 7.3 shows the posterior distribution of the infection prevalence. From the posterior distribution we clearly see that some values of the proportion infected are more likely than other. We will formally examine how we can describe the posterior distribution in more detail shortly, but for now, consider just to summaries. For example, what is the most plausible value for the infection prevalence? We can extract the maximum value from the posterior:

```
#extract the max value of the posterior
d[which.max(d$posterior), ]
```

```
##   p.grid prior       lik  marg.lik  posterior
## 8   0.07     1 0.3801461 0.3801461 0.06094516
```

The value of the parameter with the maximum posterior probability is called the **maximum a posteriori (MAP) value**, and it is one of the simplest values used to describe the posterior distribution with a single point value. In this case, we see the MAP for the prevalence is 7% (`p.grid`). That happens to be the point estimate for the frequentist estimate (rounded from 0.067 because of the finite nature of our grid), but the symmetry here is only because we used a completely uninformative prior. Indeed, in simple problems of estimation, the frequentist point estimate and Bayesian MAP are often the same when using an uninformative prior. But as we will soon see, there is rarely a reason to use a completely uninformative prior.

Moreover, even when we use an uninformative prior, we can do so much more with the Bayesian analysis compared to frequentist estimation. Recall that our primary interest in estimating the prevalence of the disease was to determine if the prevalence is greater than 10%, because that's the threshold where public health interventions would be triggered. We saw with frequentist inference that we can't easily specify a null hypothesis that encompasses multiple values of the parameter. But with Bayesian inference, we can quantify exactly what we want. Let's go ahead and compute the probability that the prevalence of the infection is greater than 10%:

```
sum(d$posterior[d$p.grid > 0.1])
```

```
## [1] 0.4885063
```

Here we see that there's a 48.9% chance that the prevalence is greater than 10%. That of course means there's a 51.1% chance that the prevalence is 10% or less. In other words, based in our prior knowledge and the new data we collected (the 15 test results), the it's basically a 50/50 coin flip on whether the prevalence is above the threshold for triggering interventions. That is exactly the kind of information public health officials would want to consider as part of their decision making.

7.4.4.1 Using a more informative prior

Now let's approach the same estimation problem but with a more informative prior. Suppose you had very good reason to believe that the prevalence of the infection was no greater than 15%. For example, perhaps you know the infection was very recently introduced to the area from someone who recently returned from travel in another area where the disease is common. Or perhaps you know from others areas that the prevalence has never been observed to be above 15%. And suppose you also know of at least one case in the population, such that the prevalence can't be 0.

We can go ahead and specify the prior as $p_{infected} \sim \text{Uniform}(0.001, 0.15)$, which says that all values of the prevalence from 0.001 to 0.15 are equally likely, and that all values outside of that range are impossible. Let's create the grid and visualize the prior:

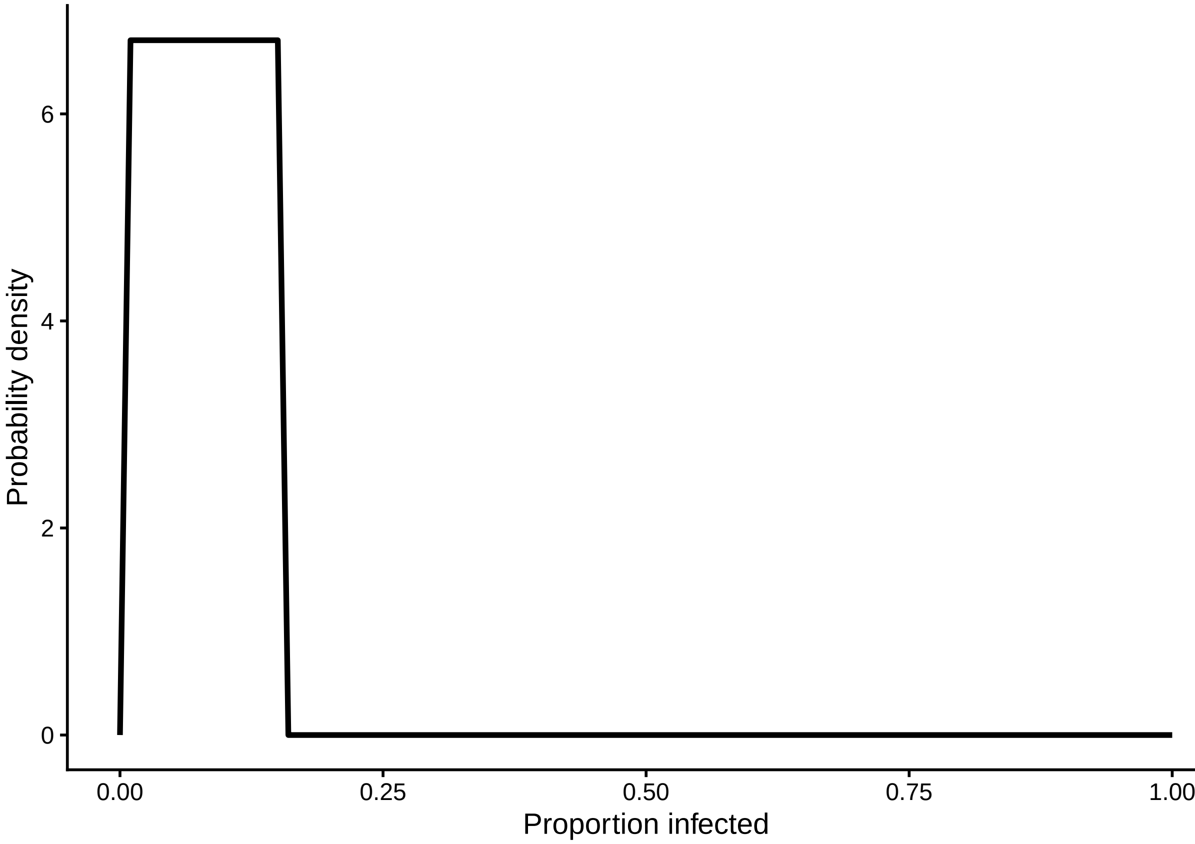


Figure 7.4: Uniform prior distribution of the proportion infected ruling out values above 15% prevalence.

Figure 7.4 shows the expected prior distribution, with uniform probability from 0 to 0.15 and all other values being plausible. Now we can combine this prior with the data and compute the posterior distribution:

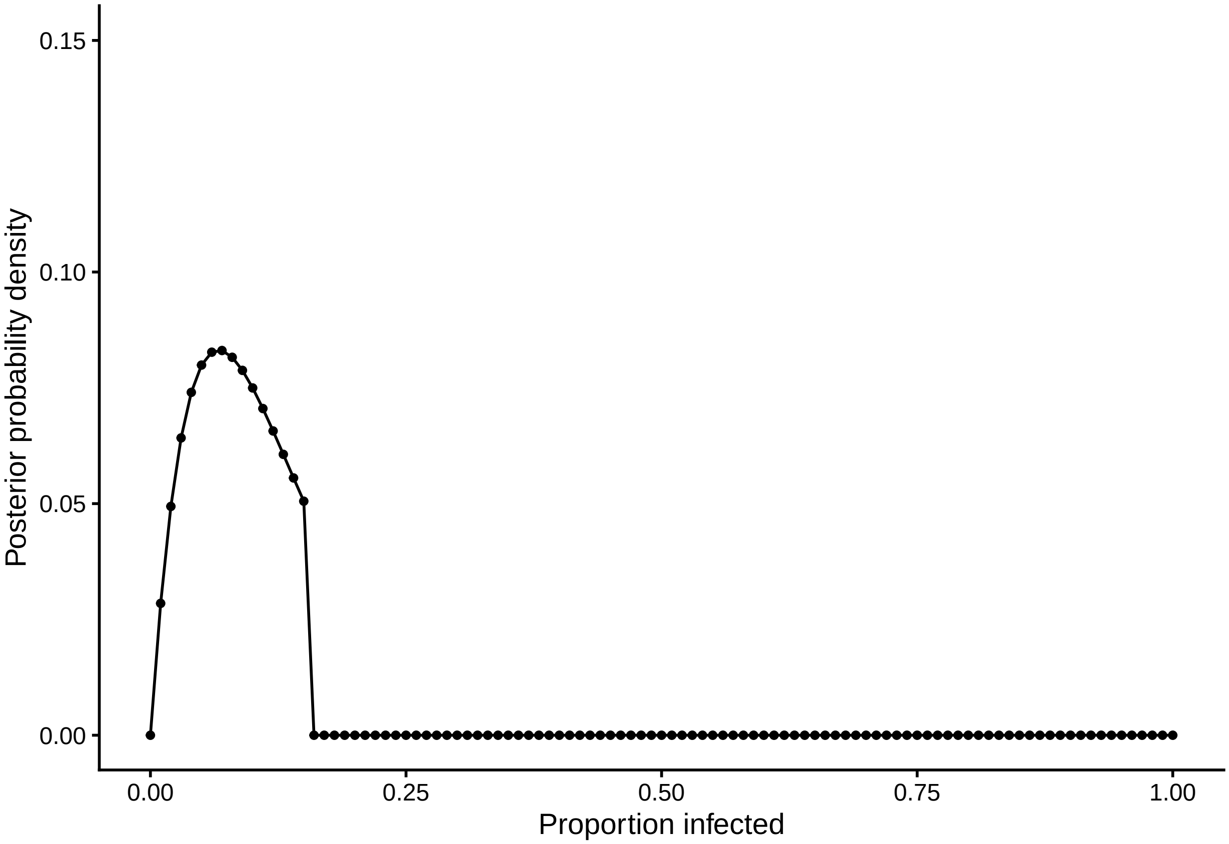


Figure 7.5: Revised posterior probability distribution for the infection prevalence conditional on the observed data when using a uniform prior from 0 to 15%.

We see here a slight difference in the posterior distribution. Graphically the most likely value still appears to be 0.07, but let's re-quantify the probabilty that the prevalence is greater than 10%:

```
sum(d$posterior[d$p.grid > 0.1])
```

```
## [1] 0.302917
```

Not surprisingly the posterior probability of the prevalence being greater than 10% is now a bit lower at 30%, much lower than when we used an uninformative prior. Thus, the shape of the prior probability distribution - representing our prior knowledge of the system - can play a very important role in shaping the estimate as expressed by the posterior distribution.

7.5 Summarizing the posterior distribution

How should we summarize the posterior distributions? We have already seen two approaches, specifically the MAP and an interval of probability mass ($p > 0.1$)). Because posterior distributions are simply probability distributions, we can describe them with any of the tools we've used to describe probability distributions. We can describe the central tendency of the posterior by quantifying the mean, median, or mode (which is the MAP), and we can describe variation by metrics that quantify the width of the posterior distribution. The central tendency will give us a look at the most likely values of the parameter given the data, whereas variation will give us a sense of uncertainty. Wider posterior distributions imply more uncertainty about the parameter value.

7.5.1 Sampling from the posterior distribution

How do we quantify descriptive metrics of the posterior distribution, such as the mean? Given the simplicity of the research problem in this chapter, we could quantify the mean directly from the posterior by weighing each value i of the prevalence from the grid by its posterior probability (i.e, $\sum p_i P(p_i)$. But posterior distributions will rarely be this simple, and so we need a more general approach.

The more general approach we'll use to summarize posterior distributions in the rest of the book is to take many samples from the posterior distribution. For example, let's take 1000 samples of the prevalence values from the posterior distribution. For each draw from the posterior, we will specify that the probability of a particular value of the prevalence being selected is equal to its posterior probability. This is much like drawing marbles from an urn. Imagine the marbles have values of the prevalence printed on them. Some are 0.01, some

0.02, 0.03, etc. Based on the posterior distribution we've estimated, there will be more marbles with 0.07 (which is the MAP) than 0.05, or 0.12, so it's more likely to select a marble with 0.07 than other values on each particular draw.

Let's use the `sample` function to draw 1000 values, and then plot a histogram of the results:

```r
set.seed(123)

#extract 1000 samples from the posterior distribution
p.samples <- sample(d$p.grid, prob=d$posterior, size=1000, replace=TRUE)
p.samples <- as.data.frame(p.samples)

ggplot(p.samples, aes(x = p.samples)) +
  geom_histogram(aes(y = after_stat(density)), binwidth = 0.01,
                 color = "black", fill = "lightblue", alpha = 0.5) +
  geom_density(color = "red", size = 1) +
  scale_y_continuous(name = "Probability Density",
                     sec.axis = sec_axis(~ . * nrow(p.samples) * 0.01, name = "Frequency")) +
  labs(x = "Infection prevalence") +
  theme_classic()
```

We can see the most likely values of the prevalence among the sampled values are 0.05-0.10, and no values below 0.01 or above 0.15 were selected. This makes sense based on the posterior distribution, which peaked at 0.07 and had very low probability at 0 and above 0.15. Now that we have these samples, we can use them to easily quantify numeric values summarizing the posterior distribution.

7.5.2 Central tendency and variance

Let's start by computing the posterior mean and median:

```r
p.samples <- p.samples$p.samples

mean(p.samples)
```

```
## [1] 0.08157
```

```r
median(p.samples)
```

```
## [1] 0.08
```

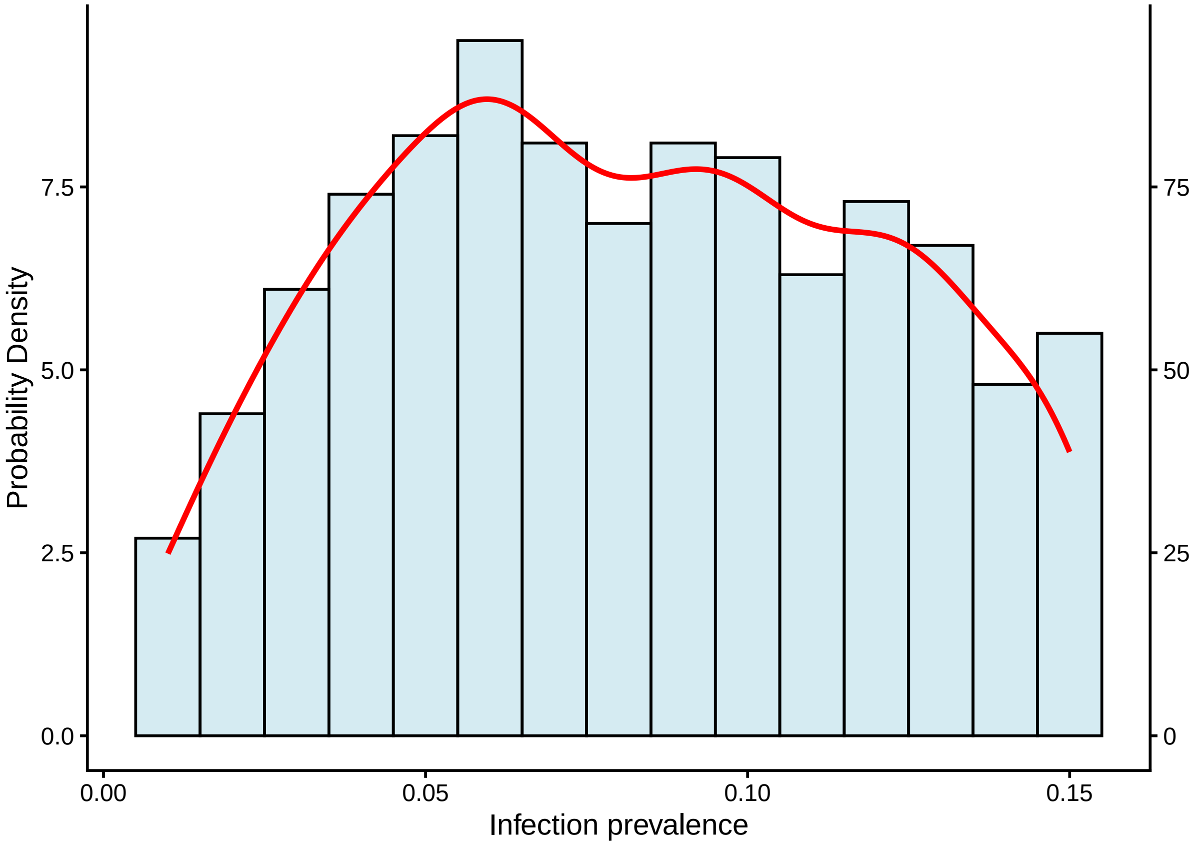


Figure 7.6: Histogram and overlaid probability density for 1000 sampled values of prevlaence from the posterior distribution.

The posterior mean is 8.2%, whereas the posterior median is 8%. These aren't too different, which isn't surprising given that the bell shape of the posterior distribution in this case.

In addition to metrics of central tendency, we can also quantify metrics of spread, which sheds light on uncertainty. The most typical metric of spread for the posterior distribution is the standard deviation:

```
sd(p.samples)
```

```
## [1] 0.03877549
```

The standard deviation of the posterior distribution reflects uncertainty about the parameter in light of the prior and the data. Bigger values of the standard deviation mean more spread in the probability distribution across possible values of the estimand, which means more uncertainty. Conversely, smaller values of the standard deviation imply less uncertainty.

7.5.3 Intervals

Although single values like the posterior mean or median can be helpful (and are often reported in the literature), it can be helpful and more informative identify an interval of plausible values for the the parameter. Indeed, we've already seen one interval from the posterior, specifically the probability that the prevalence is greater than 10%. We can easily compute this from the posterior samples too:

```
mean(p.samples > 0.1)
```

```
## [1] 0.306
```

Here we are simply asking which values of `p.samples` are greater than 0.1, which R specifies logically as TRUE or FALSE, and then we return the proportion by applying the `mean` function (because R treats `TRUE` as 1 and `FALSE` as 0), which R are greater th summing up the number of samples that are greater than 0.1 .Here we get a a value very close to what we saw before.

We can also quantify intervals that describe the probability of the parameter value falling between two points. In Bayesian estimation these intervals are often called **credible intervals.** Let's quantify a 90% credible interval. This is straightforward by using the `quantile` function. For a 90% credible interval, the lower boundary would be the value of prevalence for which 5% of the probability mass falls below the value, and the upper boundary is the value for which 5% of the probability mass lies above the value.

```
quantile(p.samples, c(0.05 , 0.95))
```

```
##   5%  95%
## 0.02 0.15
```

The interpretation here is that there's a 90% probability that the true prevalence is between 2% and 15%. How about a 98% credible interval?

```
quantile(p.samples, c(0.01 , 0.99))
```

```
##   1%  99%
## 0.01 0.15
```

We conclude there is a 98% probability that the true prevalence is between 1 and 15%. Note that as you increase the degree of credibility in an interval, the interval becomes wider. Converseely, when you decrease the degree of credibility, that interval narrows. That should make sense intuitively. At its extreme, a 100% credibility interval must include all the possible values of the estimand.

7.6 Specifying priors

Arguably the most difficult part of Bayesian analysis is selecting the priors. We have seen that the choice of prior distribution influences the posterior estimate. In our example the difference was only slight, but this will not always be the case. For example, suppose we had specified a uniform prior for the prevalence between 50% and 100%. That prior says any values below 50% are impossible, and so even though the data indicate values around 8% are most likely, those values would not be plausible in the posterior distribution.

In this book we will almost never use completely uninformative priors, precisely because we rarely conduct a study with no prior knowledge. At the same time, we will almost never use strongly informative priors that could ultimately stack the deck against the data we observe. More often than not, we'll use priors that are *weakly informative*, or what we'll call *regularizing priors*. These priors will generally not stack the deck in one particular direction when testing a hypothesis about a particular effect. For example, if we were testing whether a vaccine reduces the prevalence of an infection, we generally wouldn't use a prior that makes it more likely that the vaccine reduces the prevalence than having no effect at all. Instead, we will generally use priors that rule out values that we are confident are quite implausible.

For example, suppose the disease we're working with to generate a vaccine is known to evolve rapidly. The particular strains change from year to year, and so we wouldn't expect a vaccine to be 100% effective. In this case, we could specify a prior distribution that makes it very unlikely (or potentially impossible) that the effectiveness is 100%.

Consider another example. Suppose you are estimating the average height of people in your community. We have a lot of prior knowledge about human height. You know there are many 10-foot adults walking around, for example. We can specify a prior for the mean that is centered around typical values of mean height in other communities (perhaps in the 4-6 ft range) and that makes values outside of that range very unlikely.

How do we specify priors? We use probability distributions. We've already seen this for the infection prevalence example. We specified uniform prior distributions for prevalence. But we aren't limited to a uniform prior. We could have specified a prior distribution that makes some value of the proportion more likely than others. One possibility is the normal distribution, which is particularly useful for parameters that can be positive or negative. In the height example, we might specify a normal prior distribution for the mean height with a mean of 68 inches and a standard deviation of 2 inches. This would effectively say that our prior knowledge leads us to believe that the mean height is most likely (indeed, about 95% likely) between 64 and 72 inches. We'll use normal priors in following chapters.

For proportions, which is our focus in this chapter, a particularly valuable probability distribution is the **beta distribution**, which is a continuous probability distribution constrained to values between 0 and 1. In fact the values 0 and 1 are excluded in the beta distribution, so this is a really handy distribution when we want to exclude the two extremes of a proportion that we know are unreasonable.

The beta distribution has two parameters, α and β, that control the shape of the distribution. In statistical notation we indicate a random variable X follows a beta distribution as

$$X \sim Beta(\alpha, \beta)$$

The beta probability density for different values of the proportion can be quantified with the `dbeta` function. Figure @ref(fig:c07_f7) shows three examples of how the shape of the beta distribution changes with α and β. The shape parameters don't control the distribution in a simple way, and so it is advisable to play around with the values before settling on a particularity prior distribution. There are useful applications on the web that allow you to visualize the beta distribution while changing the shape parameters with a slider.

Let's apply a beta prior to the infection prevalence example. Let's assume we are very confident the infection prevalence is most likely 1-10% but and very unlikely to be above 20%. We'll specify the following beta distribution for the

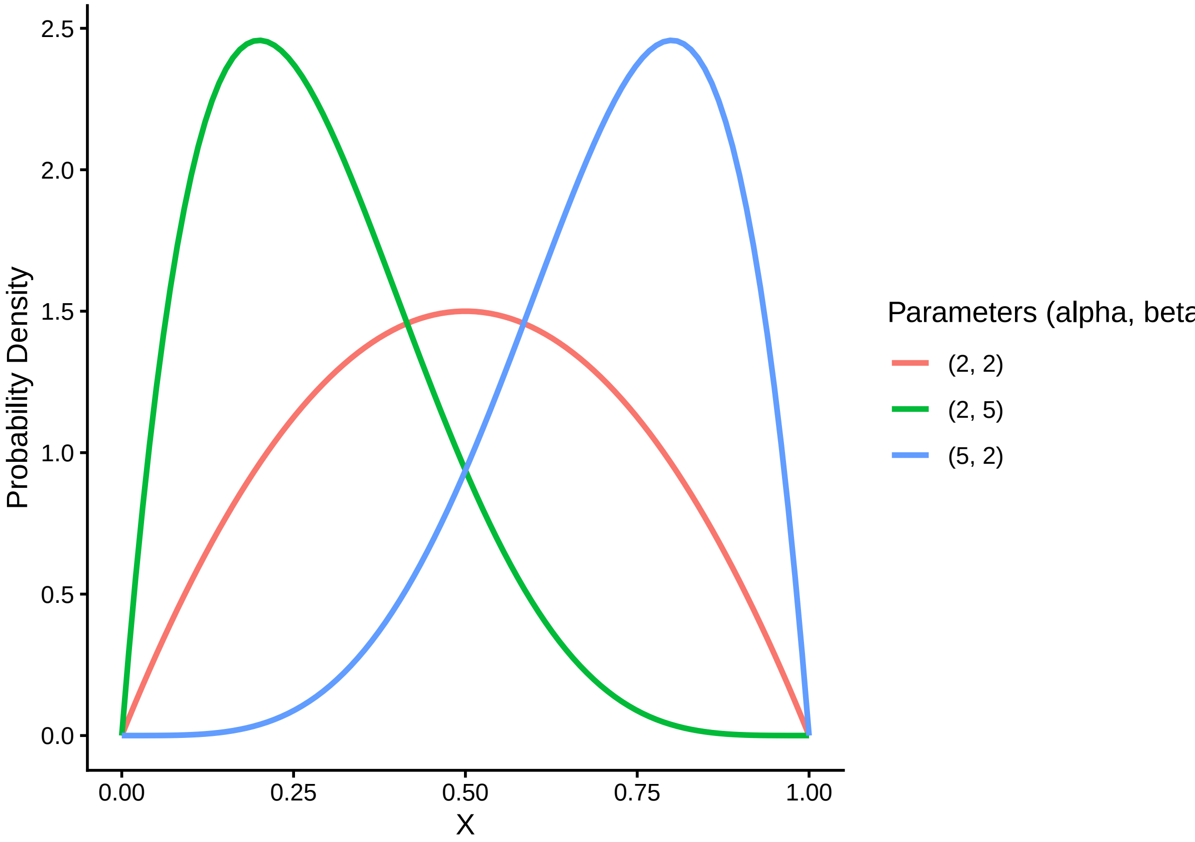


Figure 7.7: Example beta distributions with varying parameters alpha and beta.

prevalence: $p_{infected} \sim Beta(2.5, 30)$. Let's first visualize this particular beta distribution to confirm it's doing what we want:

```
#create a grid of values at which to evaluate the parameter
p.grid <- seq(from = 0, to = 1, by = 0.01)

#compute the beta priorfrom 0 to 0.2
prior <- dbeta(x = p.grid, shape1=2.5, shape2 =30)

#combine to df
d <- cbind.data.frame(p.grid, prior)

# Plot the beta prior distribution
ggplot(d, aes(x = p.grid, y = prior)) +
  geom_point(size = 1) +
  geom_line() +
  ylim(0, max(d$prior)) +
  labs(x = "Proportion infected", y = "Probability density") +
  theme_classic()
```

Looks good! We can see this prior distribution says that the most likely value of prevalence is 5%, and that 0% and values above 25% are basically implausible. Now we continue by computing the likelihood, marginal likelihood, and posterior distribution:

For comparison, let's compute the MAP and the probability that the prevalence is greater than 10% based on this posterior

```
#extract the max value of the posterior
d[which.max(d$posterior), ]
```

```
##   p.grid    prior       lik marg.lik posterior
## 7   0.06 9.630747 0.3784709 31.55932 0.1154954
```

```
#p(greater than 10%)
sum(d$posterior[d$p.grid > 0.1])
```

```
## [1] 0.1860338
```

Here we see the most plausible value of prevalence is 6%, and the probability that the prevalence is greater than 10% is now only 19%. This reflects our prior information that the infection is in the community (not 0%) but is most likely around 5% and not greater than 25%, combined with the new data which has a maximum likelihood value at 7%. Thus, the posterior distribution is

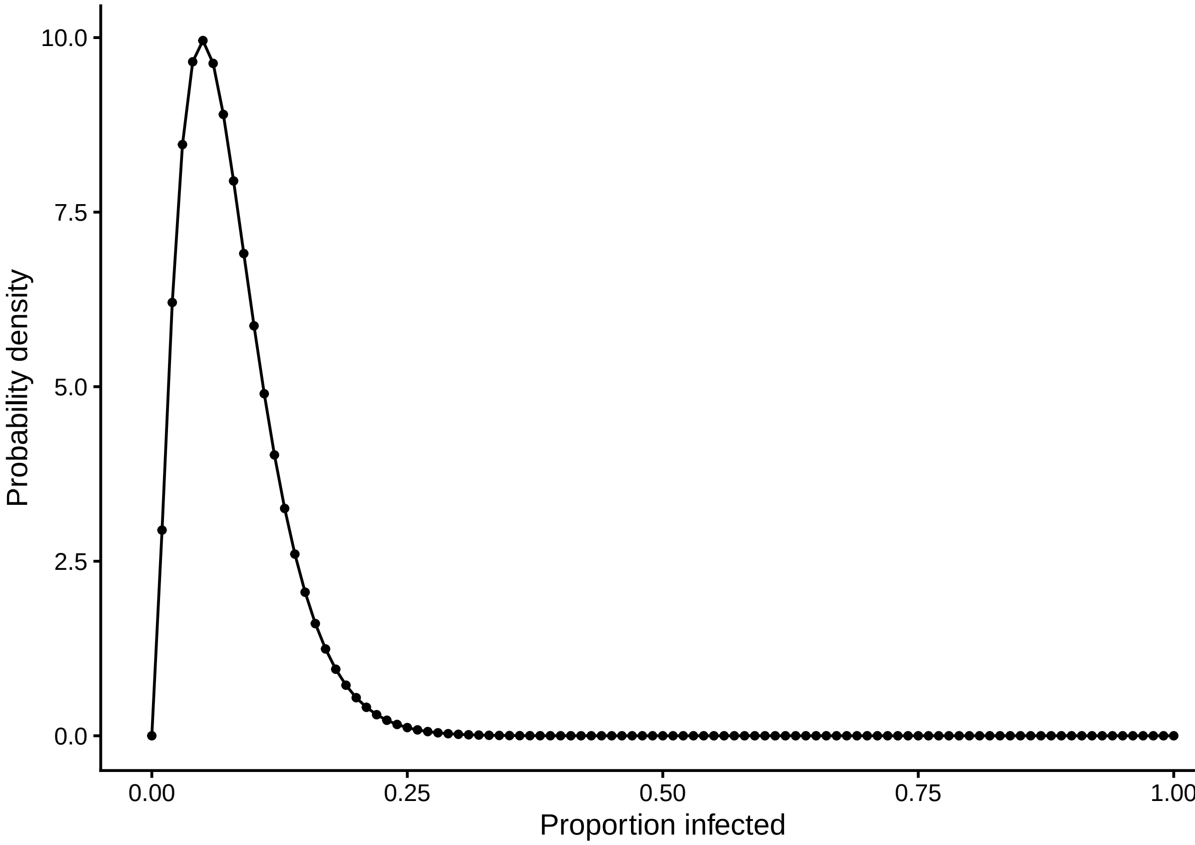


Figure 7.8: Prior distribution of prevalence specified as a beta distribution with alpha = 2.5 and beta = 30

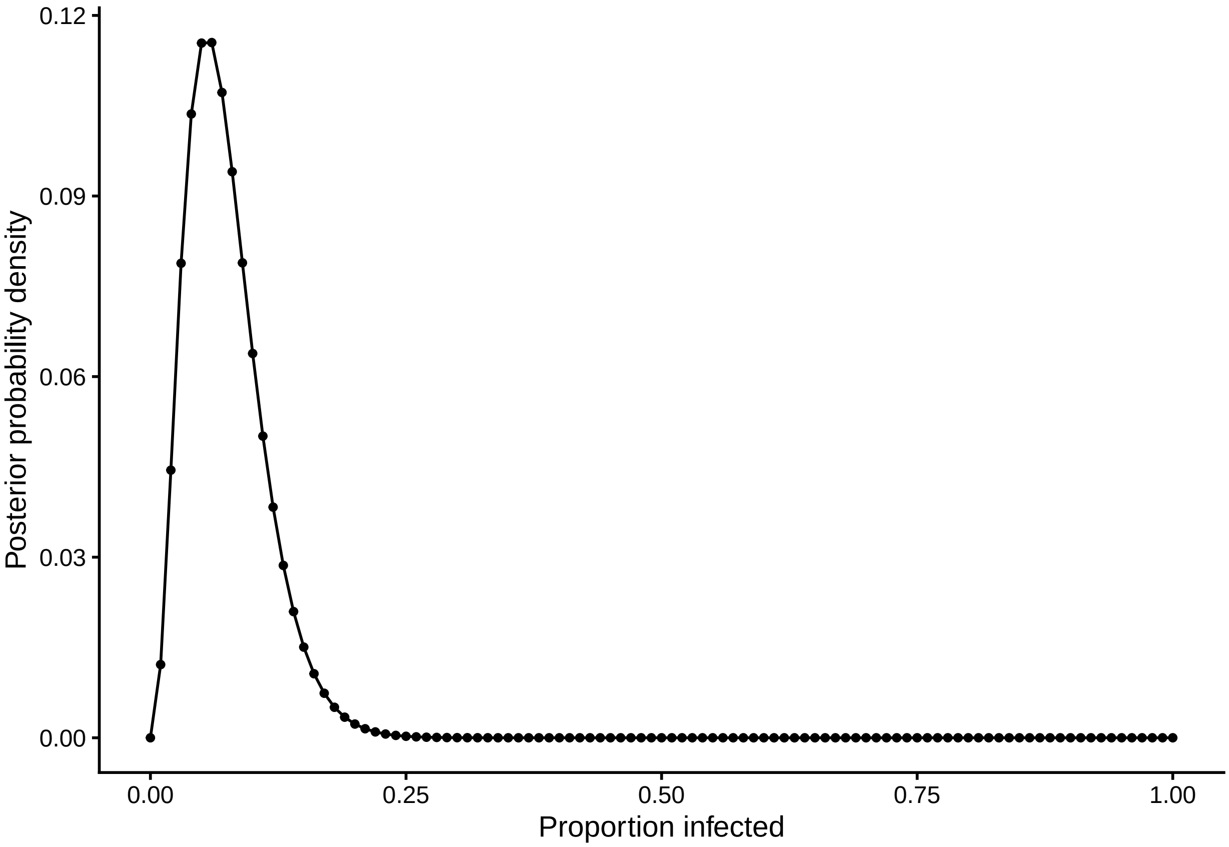


Figure 7.9: Posterior distribution of the infection prevalence when using a beta prior distribution.

slightly shifted to higher values than our prior. This is the essence of Bayesian analysis. You come into the analysis with prior knowledge, and you update that prior knowledge with new data, and express your updated view as a posterior distribution. The key is to have a good rationale for how you express the prior distribution. If it were plausible that the prevalence is above 25%, the prior we just used would not be a good choice!

7.7 Decision making

How do we make decisions about hypotheses? In Bayesian inference, the estimate is the entire posterior distribution representing the probability that the estimand takes on any particular value. As we just saw, this allows you to quantify the direct probabilities that are of interest based on the research question. For example, if you are testing the hpyothesis that pollution reduces fish abundance, you might estimate the effect by quantifying the posterior distribution for the difference in abundance between polluted and unpolluted areas. Suppose the estimate is the difference in mean abundance, specified as the mean unpolluted abundance minus the mean polluted abundance. You can use the posterior distribution to directly quantify the probability that the difference in means is a positive value. You might conclude something like, “there was a 91% probability that the mean abundance was greater in unpolluted than polluted areas”.

The bottom line is that decision making with Bayesian inference is less about “yes” or “no” and more about evaluating hypotheses - as represented by values of the estimand - with probability, which directly represents our uncertainty. Some might find the absence of firm, binary decisions to be inconvenient, but one should remember that in science there is no certainty. There is only uncertainty, and we should quantify it. Bayesian inference allows us to quantify uncertainty about hypotheses, represented as parameter values, directly.

7.8 Specifying and fitting statistical models with Bayes

For the remainder of this book, we will approach Bayesian data analysis by first specifying the statistical notation for our model, and then by fitting the model with R. Let’s take a look at those two components.

7.8.1 Statistical model specification

Once we have a generative model, a study design, and estimands clearly defined, we can define a statistical model for the data and use it to estimate the estimands. As we just saw, Bayesian estimation of parameters in statistical models requires three components: 1) the likelihood, 2) the prior(s), and 3) the marginal likelihood. The marginal likelihood is a numerical necessity to standardize the posterior distribution, and it is computed based on the likelihood and prior, so when defining statistical models for Bayesian estimation, we focus our efforts on defining the likelihood and prior. Fitting a statistical model requires specifying the model mathematically, and we will use a particular notation to do so.

Consider our infection prevalence example. Using the more informative prior, we can define the statistical model in this way:

$$I \sim Binomial(N, p) p \sim Beta(2.5, 30)$$

What do these lines represent? The first line is the likelihood and communicates our assumptions about how the observed data are produced based on the generative model. It says the observed number of infections (I) follow a binomial distribution with N tests and a constant probability of being infected, p. The probability of being infected, p, is the estimand - the parameter of interest - and what we need to estimate with the observed data, I and N. The tilde (~) specifies that the observed number of infections is a **stochastic** outcome, which just means that the observed number of infections is probabilistic rather than determined with certainty. Indeed, as we've defined the model, I is a random variable described by a binomial distribution.

The second line defines the prior distribution, which is just our mathematical way of describing our prior knowledge about the parameter. It says we assume the probability of infection follows a beta distribution with shape parameters $\alpha = 2.5$ and $\beta = 30$. When we looked at this beta distribution, we saw that it reflects our prior belief that the prevalence is most likely around 5% and very unlikely to be 0% or >20%.

Every statistical model we define for Bayesian estimation requires these two components, the likelihood and prior distribution. The likelihood includes observed variables (the data) and unobserved parameters (the quantities we need to estimate) and described how the observed data are generated. In contrast the prior distributions describe our prior knowledge for each unobserved parameter.

Some statistical models are quite simple. Indeed, or model of infection prevalence has only a single parameter, and it happens to be the estimand. Other models will have multiple parameters, not all of which are necessarily of direct interest. Every unknown parameter requires a prior distribution, whether you are interested in the parameter or now. So when we fit more complicated models with multiple parameters, you'll want to make sure that there's a prior distribution specified for each parameter. We'll see an example like that shortly.

7.8.2 Fitting models with *brms*

Moving forward we will use the *brms* package in R to fit models with Bayesian inference. Although we could continue to fit some models with grid approximation as we have this chapter, grid approximation becomes very cumbersome and computationally intensive when fitting complex models with multiple parameters.

There are different types of methods to fit approximations of posterior distributions, and a method called **Markov chain Monte Carlo (MCMC)** is by far the most common, and the method that *brms* uses. We will not get into technical details on how MCMC works, but at a high level, MCMC is similar to the sampling approach you've seen with grid approximation. However, instead of evaluating pre-specified values for parameters on a grid, MCMC uses an mathematical algorithm to explore possible values for the parameters (i.e., the *parameter space*) and hone in on the most likely values for the parameters.

MCMC works by generating a single sample of parameter values at a time, where each new sample depends on the previous samples (a Markov process). A sequence of samples is called a **chain**. You can think of a chain like a random walk. The chain explores parameter space, and each sample depends on the parameter space explored in the previous sample, but over a long enough run of samples, the parameter space at the end of the chain can be very different from the start. Ultimately the goal is to run chains long enough such that the parameter space converges to a consistent area where the true posterior distribution is found. So when we fit models in *brms* with MCMC, we will need to use specific metrics to evaluate whether the chains converged. When the model converges, we will use the samples generated to describe the posterior distribution, much like we did when we sampled from our grid approximation of the posterior.

So how do we fit our model of infection prevalence with MCMC in *brms*? Let's take a look:

```r
library(brms)

#organize data
x <- 1
n <- 15
d <- list(x = x, n = n)

#specify model formula
m.formula <- bf(x | trials(n) ~ 1,
                family = binomial(link="identity"))

#specify priors
m.prior <- prior(beta(2.5, 30), class = "Intercept", lb=0, ub=1)
```

```r
#simulate the priors
m <- brm(data = d,
         formula = m.formula,
         prior = m.prior,
         refresh = 0)
```

```
## Running /Library/Frameworks/R.framework/Resources/bin/R CMD SHLIB foo.c
## using C compiler: ‘Apple clang version 17.0.0 (clang-1700.0.13.3)’
## using SDK: ‘MacOSX15.4.sdk’
## clang -arch arm64 -std=gnu2x -I"/Library/Frameworks/R.framework/Resources/include" -DNDEBUG
## In file included from <built-in>:1:
## In file included from /Library/Frameworks/R.framework/Versions/4.5-arm64/Resources/library/Sta
## In file included from /Library/Frameworks/R.framework/Versions/4.5-arm64/Resources/library/Rcp
## In file included from /Library/Frameworks/R.framework/Versions/4.5-arm64/Resources/library/Rcp
## /Library/Frameworks/R.framework/Versions/4.5-arm64/Resources/library/RcppEigen/include/Eigen/s
##   679 | #include <cmath>
##       |          ^~~~~~~
## 1 error generated.
## make: *** [foo.o] Error 1
```

```r
print(m)
```

```
##  Family: binomial
##   Links: mu = identity
## Formula: x | trials(n) ~ 1
##    Data: d (Number of observations: 1)
##   Draws: 4 chains, each with iter = 2000; warmup = 1000; thin = 1;
##          total post-warmup draws = 4000
##
## Regression Coefficients:
##           Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## Intercept     0.07      0.04     0.02     0.17 1.00     1098     1281
##
## Draws were sampled using sampling(NUTS). For each parameter, Bulk_ESS
## and Tail_ESS are effective sample size measures, and Rhat is the potential
## scale reduction factor on split chains (at convergence, Rhat = 1).
```

The main steps here are to organize the data (usually in a list or data frame), specify the model formula, specify the priors, and then fit the models. The model formula is defined with the `bf` formula, and it specifies the systematic part of the model, namely how the data and parameters are related to each other. In this case we define `x` successes out of `n` trials as a function of (`~`) the parameter. Specifying `1` for the parameter simply means the data are a function of a single parameter that will be referred to as the `Intercept`. In this particular example, the `Intercept` will represent the proportion infected. The second part of the formula specifies the `family`, which is probability distribution for the response variable. In this case the family is the binomial distribution. Each probability distribution as a `link` function. Specifying `identity` link for a binomial distribution makes the 'Intercept' represent a probability.

The `prior` function defines the a prior for a particular parameter. Because there is only a single parameter, we have only a single parameter to define. The `prior` function requires a particular probability distribution to describe the prior, in this case a beta distribution with its two parameters. We then specify the `class`, which is the type of parameter for which we are defining a prior. As mentioned, the parameter in this case is denoted the `Intercept`. W

Finally the `brm` function wraps everything together (the `data`, `formula` and `prior` previously defined) and fits the model. When a model is being fit, it defaults to providing messages of its progress, but here I used the `refresh` argument to suppress those messages. Once the model is fit, we print the output `m`. The output prints a key descriptors about the model that was fit. The only new part here is the "Draws", which reports that four chains were fit, each with a total of 2000 iterations, 1000 of which were a "warmup". The iterations during the warmup phase are discarded and not considered to describe the posterior distribution. We use a warmup phase when starting a chain because the paramater values are expected to vary widely. The 1000 samples following the warmup phase are retained to describe the posterior distribution. Because we have four chains each with 1000 samples retained, the total number of samples to describe the posterior distribution is 4000. Sometimes it's useful to retain a subset of these final draws, in which case we can apply a `thin` value greater than one. Generally we should only need to do that when computer space is limited, and for the models in this book, that won't be an issue.

Below the basic model description is a section called "Regression Coefficients", which is the main output of the mdoel. The `Intercept` line here represents the probability of infection. The `Estimate` is the mean of the posteiror distribution (in this case 7%), and the `Est.Error` is the standard deviation of the posterior (in this case 4%). We are also provided a 95% credible interval, which is 2-16%. The other three values are statistics used to evaluate whether the model converged to an appropriate parameter space. More on those in the next chapter.

In addition to the standard model output, we can extract the posterior samples with the `as_draws_df` function, which we will save as `m.post`.

```
#posterior samples
m.post <- as.data.frame(as_draws_df(m))
head(m.post)
```

```
##   b_Intercept  Intercept    lprior      lp__ .chain .iteration .draw
## 1  0.03800522 0.03800522 2.250775 -2.162432      1          1     1
## 2  0.03278834 0.03278834 2.186140 -2.441243      1          2     2
## 3  0.02463775 0.02463775 2.000807 -3.072289      1          3     3
## 4  0.04761258 0.04761258 2.297758 -1.815258      1          4     4
## 5  0.04761258 0.04761258 2.297758 -1.815258      1          5     5
## 6  0.04655451 0.04655451 2.296249 -1.845059      1          6     6
```

The only column we need to focus on now is `b_Intercept`, which represents each of the 4000 samples of the proportion infected from the posterior distribution. We could use this to illustrate the posterior distribution and quantify any of the descriptors of the posterior we discussed earlier in the chapter:

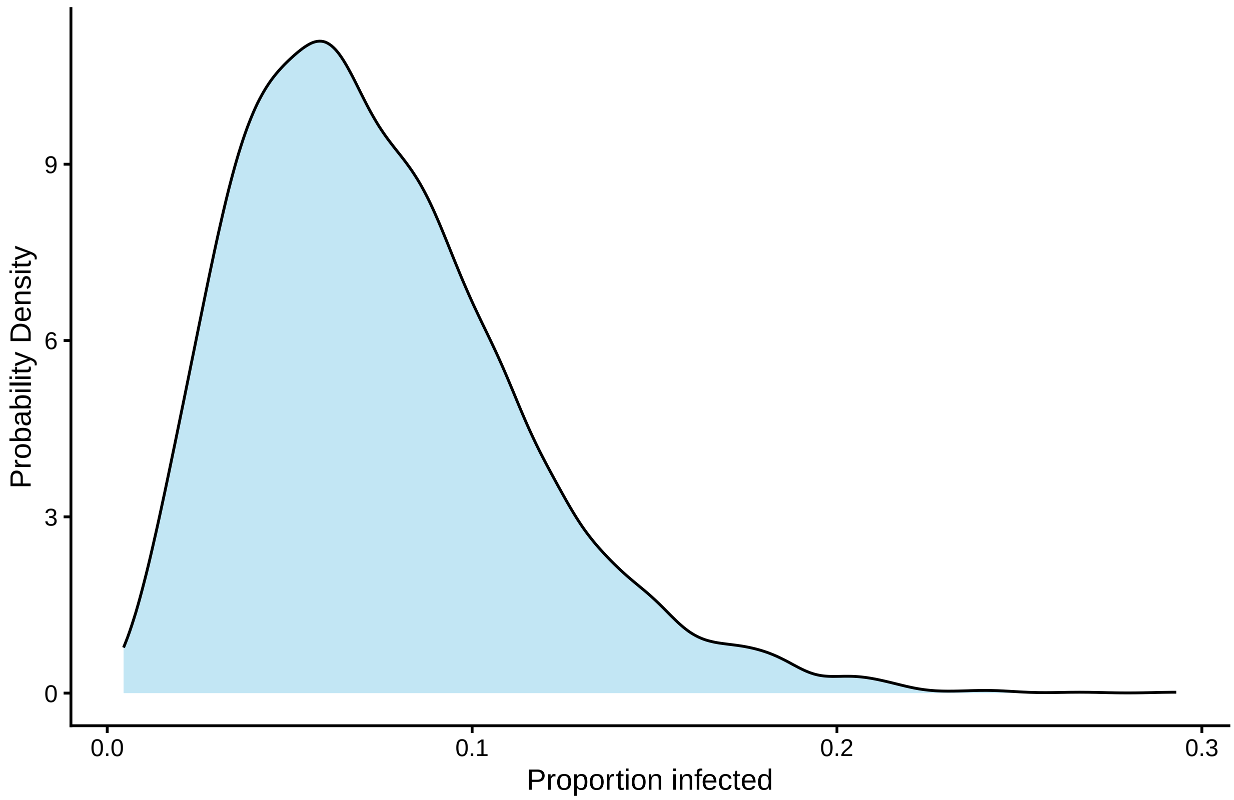


Figure 7.10: Posterior distribution of the infection prevalence estimated with brms.

```
## [1] 0.07432989
```

```
## [1] 0.2245
```

Don’t worry if this all seems confusing. It’s a lot of information, but we will get lots of practice defining statistical models, fitting them with data in brms, and interpreting the outputs. I will orient you to the key parts as we move forward.

Chapter 8

Bayesian Workflow

In this chapter we extend the process Bayesian estimation in two ways. First, we will look at estimation of a model with parameters for a continuous random variable. Second, we will begin practicing a consistent approach to estimating parameters that includes explicit steps of picking an appropriate statistical model, checking the performance of the model with chosen prior distributions, fitting the model, and checking the performance of the fitted model. This kind of systematic approach to model selection and checking is called **Bayesian workflow**. To explore the components of Bayesian workflow in this chapter, we will consider the following question: What is the average resting heart rate for adults in the United States?

The data we'll this chapter use are included in an R packages called *NHANES*, which stands for the U.S. National Health and Nutrition Examination study. NHANES is a survey of a variety of health parameters for people in the U.S. Let's start by loading the data:

```r
#install.packages("NHANES") #install the package if necessary
library(NHANES) #load the package
data(NHANES) #load the dataframe NHANES
d <- NHANES #simplify the name of the data frame
```

Running the code above produces a data frame called `d`. I encourage you to look at the structure of the full dataframe (`str(d)`), which includes dozens of variables, but for now let's turn our attention to a summary of two variabiles of interest: `Age` (years) and `Pulse` (resting heart rate in beats per minute, bpm).

```r
summary(d[, c("Age","Pulse")])
```

```
##       Age             Pulse
##  Min.   : 0.00   Min.   : 40.00  
##  1st Qu.:17.00   1st Qu.: 64.00  
##  Median :36.00   Median : 72.00  
##  Mean   :36.74   Mean   : 73.56  
##  3rd Qu.:54.00   3rd Qu.: 82.00  
##  Max.   :80.00   Max.   :136.00  
##                  NA's   :1437
```

You can see based on the summary distribution of each variable that the sample includes individuals of any age. For our research question, we will focus on estimating the average resting heart rate of *adults* in the U.S. We can subset the data to remove individuals under 18 years old, leaving observations for 7216 adults (excluding missing observations):

```
d <- d[d$Age >=18, ]
sum(!is.na(d$Pulse)) #sample size
```

```
## [1] 7216
```

8.1 Step 1: Choose an appropriate model for the likelihood

How are the observed data of resting heart rates *generated*? That's the question we have to address first. We need a model that characterizes how our sample data are created. In the broader context of scientific workflow introduced in Chapter 2, this process represents the arrow from the generative model to the statistical model. We need a statistical model that adequately represents the assumptions made in the generative model.

In the last chapter, our main example focused on estimating the prevalence of a respiratory infection in a population. The observed data consisted of X people who tested positive for the infection out of a sample of N people tested. When we do statistical analysis, we have to make a decision about how best to characterize how the observed data were generated. This part of our statistical model is called the **likelihood model**. For infection prevalence, we recognized that the observed data was a discrete random variable that could be described by the binomial distribution. The binomial distribution simply functioned as a model we thought was appropriate to describe how the observed number of infections was generated given the true probability of infection (the parameter we wanted to learn) and the number of people sampled.

We're going to need a different kind of likelihood model for estimating the mean resting heart rate. Resting heart rate is a continuous random variable, so the binomial distribution isn't an appropriate description of how heart rates are generated and observed. As we learned in Chapter 6, one probability distribution that can be useful for describing a continuous random variable is the normal (Gaussian) distribution. The normal distribution isn't the only probability distribution that describes continuous random variables, and for some continuous variables we need something else, but it's a good starting point.

So the first step in Bayesian workflow is selecting an appropriate likelihood model, and for resting heart rate we'll use the normal distribution for this purpose. By using the normal distribution, we assume resting heart rates are observed at random with variation around a population mean (μ) and with among-individual variation described by a standard deviation (σ). We also assume that heart rate is measured without error.

Unlike the binomial distribution - which has just a single parameter - the normal distribution has two parameters - mean and standard deviation. Because we are assuming heart rates come from a normal distribution, we have to include both the mean and the standard deviation in our likelihood model. Indeed, when we assume a variable is drawn from a particular probability distribution, we have to estimate (or assume values for) all the parameters for that probability distribution whether we want them or not. That's why it's useful to differentiate between types of parameters, the estimands being the parameters that we are most interested in.

Here's how we can communicate our choice of the normal distribution for the likelihood with statistical notation:

$$r_i \sim \text{Normal}(\mu, \sigma)$$

What does the likelihood model say? Here we are defining the observed values of heart rate r for each individual i as a random variable following a normal distribution with parameters mean = μ and standard deviation = σ. Remember the tilde symbol (~) defines a relationship as stochastic, meaning the observed values r_i are probabilistic. The normal distribution with its parameters defines the probability of drawing particular values of heart rate. We don't know what those parameter values are, so we will have to estimate them.

8.2 Step 2: Decide on prior distributions for each parameter in the likelihood

Once your likelihood model is defined, it is necessary to define prior distributions for *every* parameter in the likelihood model. How should we go about doing this? At this point in the book I want to introduce two considerations for choosing priors. Prior distributions should be selected in a way that a) respects the mathematical constraints of the parameter of interest and b) allows us to adequately specify our beliefs about values of the parameters that are more or less likely based on our prior knowledge.

First, consider the mathematical constraints of a parameter. Remember our example in the last chapter about estimating the prevalence of an infection? Our model for infection prevalence had just a single parameter, the probability of infection. Probability is bounded between the values of 0 and 1, so it is wise to choose a prior distribution that respects those boundaries. That's why we eventually landed on using the beta distribution as an appropriate probability distribution for the prior, as the beta distribution is bounded between 0 and 1. It makes values outside of the bounds of 0 and 1 impossible, and that makes good mathematical sense.

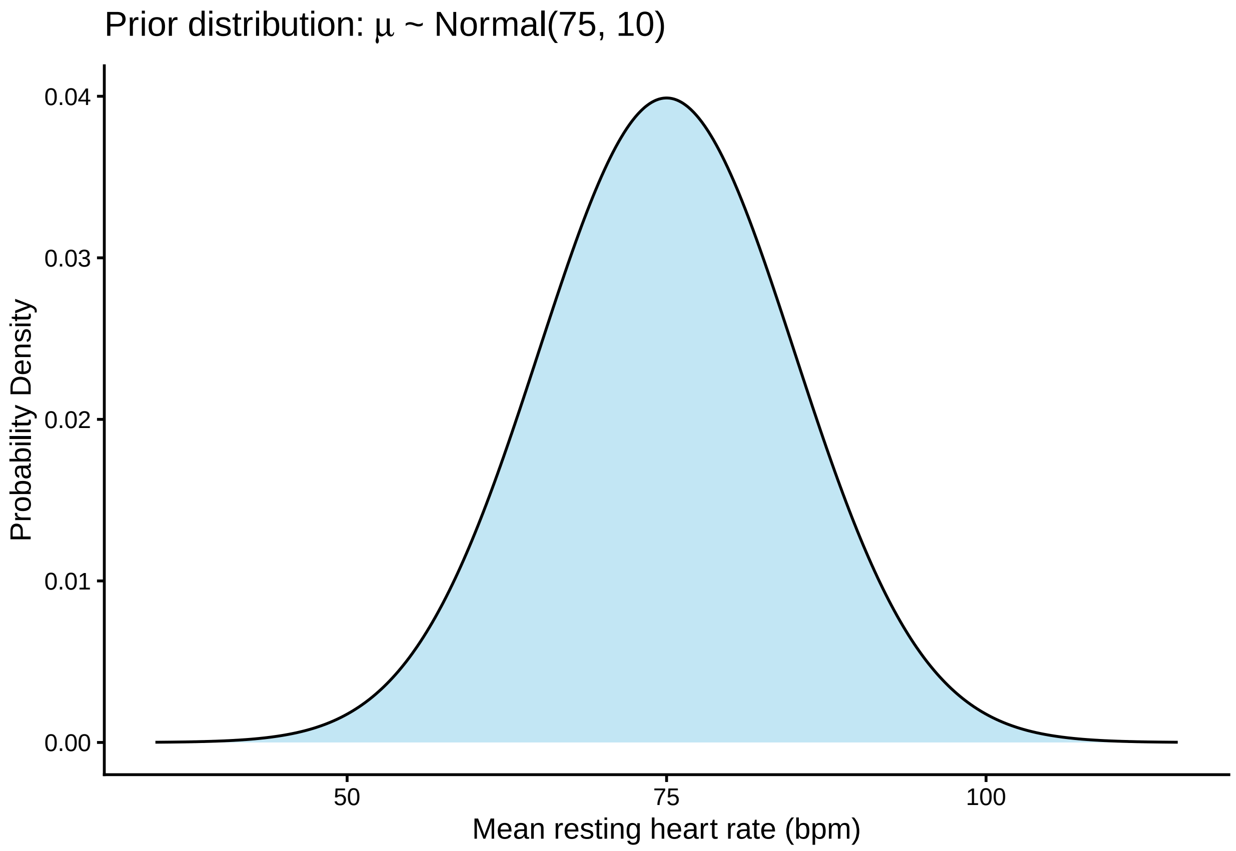
Let's apply this principle to the question before us this chapter of estimating the mean resting heart rate. Our likelihood model has two parameters, the mean and standard deviation, so we need a prior for both. Let's consider the mean first. Are there any mathematical constraints on the mean resting heart rate? The only constraint that comes to mind is that the mean resting heart rate can't be negative. It could be any real positive value, but it can't be less than 0.

Which probability distribution might be appropriate for the mean heart rate? At this point in the book we've encountered four probability distributions: binomial, uniform, beta, and normal. The binomial and beta distributions are clearly not appropriate in this case. What about the uniform distribution? Suppose we're 100% confident that the mean resting heart rate of our target population is not less than 50 or greater than 100. We could specify a uniform prior for the mean that rules out values below 50 and above 100: $\mu \sim \text{Uniform}(50, 100)$, which meets our goal of respecting the mathematical constraints of the parameter, namely that the mean heart rate can't be negative.

But now let's consider our other goal of selecting a prior that adequately reflects our prior beliefts about the parameter. The downside of the uniform distribution is that it doesn't give us any wiggle room to make some values within the range of 50 to 100 more likely than others. We might think the most plausible values for the mean resting heart rate are around 75 bpm, but the uniform distribution we defined says all the values between 50 and 100 are equally plausible. So let's consider another possibility.

What about the normal distribution? Could we use the normal distribution to define a prior for the mean heart rate? It certainly addresses the problem we just faced with the uniform distribution in that we could specify a normal prior that makes values of the mean around 75 bpm more likely than other values. For example, suppose based on prior knowledge that you are 95% confident that the true mean resting heart rate is between 55 and 95 bpm, but it's more likely to be around 75 than values at the margins. We could define the prior distribution for the mean with a normal distribution having a mean of 75 bpm and a standard deviation of 10 bpm: $\mu \sim \text{Normal}(75, 10)$. That's a nice application of the empirical rule, which says that 95% of the values of a normal distribution are within two standard deviations of the mean. This could work!

```r
mu <- 75
sigma <- 10

x <- seq(mu - 4*sigma, mu + 4*sigma, length.out = 1000)
df <- data.frame(x = x, dens = dnorm(x, mean = mu, sd = sigma))

ggplot(df, aes(x = x, y = dens)) +
  geom_area(fill = "skyblue", alpha = 0.5) +
  geom_line() +
  labs(x = "Mean resting heart rate (bpm)", y = "Probability Density",
       title = expression(paste("Prior distribution: ", mu, " ~ Normal(75, 10)"))) +
  theme_classic()
```



How did I know to use those particular values for the normal prior (mean = 75, standard deviation = 10). This is where domain knowledge is helpful. Based on prior knowledge from the medical literature and my own experience on track and cross country teams where we often measured our heart rates, I have a general sense about the most plausible values for average heart rate. I don't know it for sure, but I think it's around 75 bpm. It could be more or less, but I think it's *very* unlikely that the *typical* heart rate is less than 50 bpm or over 100 bpm. I also don't think it's equally plausible that the mean heart rate is anywhere between 50 and 100 bpm. I think it's most likely around 75, with decreasing probability moving away from 75. The normal distribution captures that prior knowledge nicely (albeit imperfectly, as every model is imperfect).

One potential problem with the normal distribution is that it it allows for values that are positive or negative. So here we face a trade-off. We've landed on a probability distribution that allows us to specify our prior belief about values near 70 bpm being most likely for the mean, but it allows negative values (as well as extremely high positive values, like one 10,000 bpm). What do we do? We have a few options. We could ditch the normal distribution and look around for another probability distribution that has a similar shape as the normal distribution (high probability density at the mean and lower elsewhere), but only allows for real values above zero. There are options, such as the gamma distribution. On the other hand, we could proceed with the normal distribution and be comfortable with the fact that the probability density of negative values and extreme positive values are minuscule. Remember that a Bayesian analysis uses both the prior *and* the observed data to quantify the posterior distribution that we make inferences with. Extremely unlikely values based on the prior probability alone will not have high posterior probability unless the observed data and likelihood model are very consistent with those values. So for now, let's proceed with the normal distribution as our prior for the mean resting heart rate.

Wow, that was a lot of effort to select a prior distribution for just a single parameter in our likelihood model. And we still have another parameter to deal with! What should we select as a prior for the standard deviation? First, let's just remember what the standard deviation represents. It represents the typical variation in resting heart rate among individuals in the target population. Admittedly, I feel like I know less about the variation among people in their heart rates than I do about the average heart rate. I wish I could just ignore the standard deviation, but because it's a parameter in my likelihood model, it needs a prior.

Now although I don't know much about variation in heart rates, I do know there's a mathematical constraint we can respect. The standard deviation cannot be less than 0. Variation either exists, in which case the standard deviation is positive, or it doesn't, in which case the standard deviation is 0. So I want to use a prior distribution that says the standard deviation is zero or positive values only, is possible.

How large could the standard deviation be? That's where things are murky for me, although if I push myself, I do think there are some obvious limits that come to mind. For example, it seems absurd the the standard deviation could be something like 30 bpm. How do I know that? Well let's suppose I'm right that the mean resting heart rate is 75 bpm. A normal distribution with a mean of 75 bpm and standard deviation of 30 bpm would allow a not insignificant portion of the population to have absurd resting heart rates, like less than 30 bpm (just 1.5 standard deviations below the mean) or greater than 110 bpm (just 1.5 standard deviations above the mean). A standard deviation of 30 bpm seems too high. What about 20 bpm? If I'm right that the mean resting heart rate is 75 bpm, a standard deviation of 20 bpm would imply that 95% of the population has a resting heart rate between 35 bpm and 115 bpm. Honestly that still seems too high, but I'm not sure where exactly to draw the line, so I'm going to draw it a tad higher than I think is believable.

Now you probably think that some values within the range of 0 to 20 bpm are more plausible for the standard deviation than others. I do too, and we could consider a normal distribution as our prior here. But I'm going to caution against that. Unlike the mean resting heart rate, the range of values for the standard deviation are much closer to 0, and I'm concerned about giving too much credence to negative values for the standard deviation. So for now, let's choose a uniform distribution between 0 and 20 as our prior for the standard deviation: $\mu \sim \text{Uniform}(50, 100)$. This simply says that, before seeing the data, we consider any value of the standard deviation between 0 and 20 bpm to be equally plausible. The lower bound of 0 makes good sense because standard deviations cannot be negative (there can't be negative variation). The upper bound of 20 bpm is also conservative. Uniform distributions can work well as prior distribution for paramter that are only positive and when we only know enough to define hard limits on possible values. Yet the uniform distribution isn't completely uninformative. Indeed, it makes a rather strong assumption that values anywhere in the range of 0 and 20 are equally plausible, and other values are impossible.

So there we have it. We've selected a likelihood model, and we've selected prior distributions for each parameter in our likelihood model. This is our full statistical model that we've defined:

$$\begin{aligned} &r_i \sim \text{Normal}(\mu, \sigma) \\ &\mu \sim \text{Normal}(75, 10) \\ &\sigma \sim \text{Uniform}(0, 20) \end{aligned}$$

8.3 Step 3: Prior predictive check

Are the prior distributions we selected reasonable? We can - and indeed we should - interrogate this question before we fit the model. This is the goal of a **prior predictive check**. We want to formally look at what the prior distributions we selected imply about the data, independently of the observed data. For simple models like ours, this is a straightforward task that we can accomplish by simulating data. Our prior distributions allow for an infinite number of combinations of mean and stadnard deviation for the resting heart rate, but they make some combinations more likely than others. The goal here is to evaluate the *joint distribution* of the mean and standard deviation bsaed on the priors and ensure that we're not implying values that make no sense.

Let's simulate some data so you can see what I mean:

```
#from https://bookdown.org/content/4857/geocentric-models.html#a-language-for-describi
n <- 10000

set.seed(123)

#randomly draw means from the prior
mu.sim <- rnorm(n, mean = 75, sd = 10)

#randomly draw SDs from the prior
sigma.sim <- runif(n, min = 0, max = 20)

#randomly draw values of resting heart rate from the combined means and SDs
r.sim <- rnorm(n, mean = mu.sim, sd = sigma.sim)

#plot the simulated distribution of heart rates
df <- data.frame(r.sim = r.sim)

ggplot(df, aes(x = r.sim)) +
  geom_density(fill = "lightblue", color = "darkblue") +
  labs(title = "Implied distribution of resting heart rate from priors",
       x = "Resting heart rate (bpm)",
       y = "Density") +
  theme_minimal()
```

What is this code doing? First, we randomly draw 10,000 values of the mean resting heart rate from the prior distribution that we defined for the mean (a normal distribution with mean = 75 and SD = 10). Next, we randomly draw 10,000 values for the standard deviation from the prior distribution we defined for that parameter (a uniform distribution between 0 and 20). Finally, we randomly draw 10,000 values of individual resting heart rates from the normal distributions implied by the simualted means and standard deviations from our priors. Then we plot the implied distribution of resting heart rates.

What do we learn here? We can see the distribution of heart rate implied by the priors has a mean right around 75 bpm as expected, so that's good. But what's not good is that the variation around that mean looks a bit unrealistic. The priors are implying that people (albeit only a few) can have a resting heart

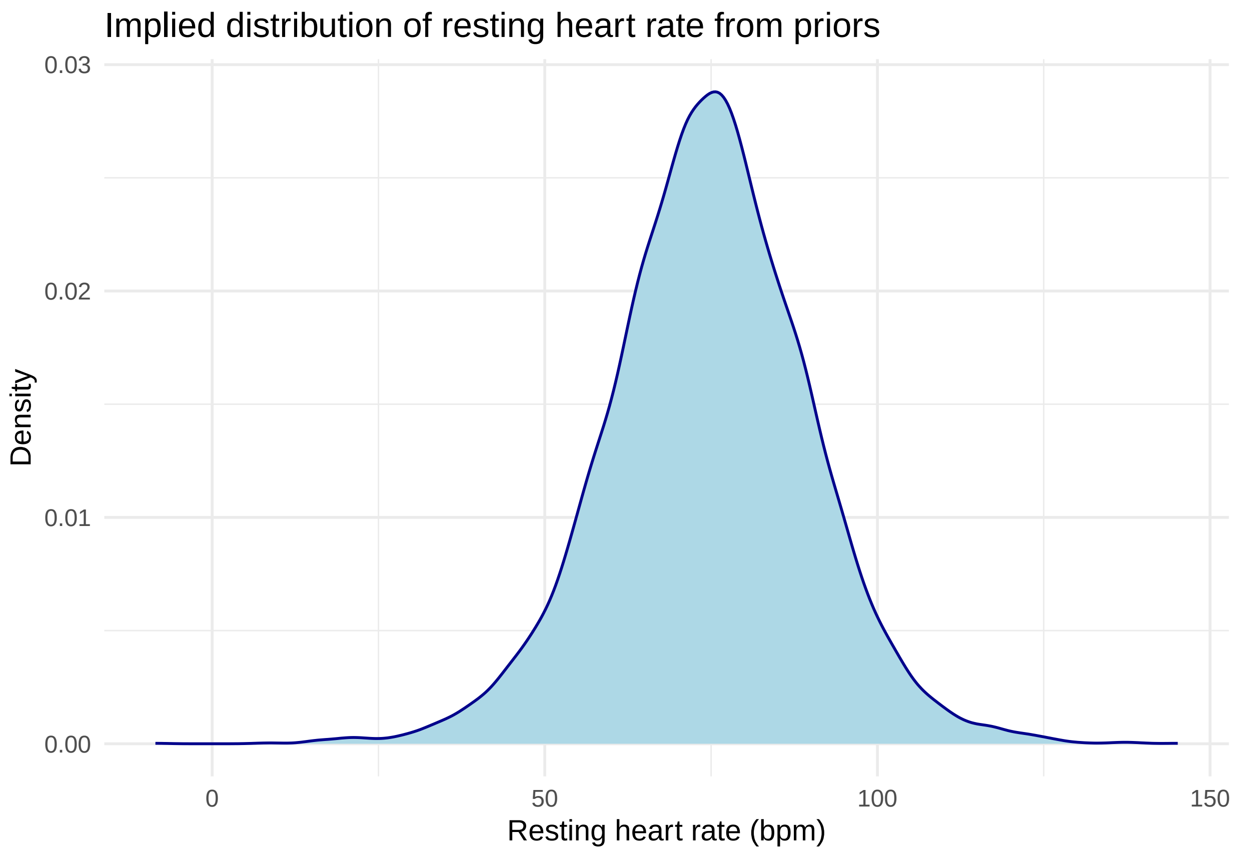


Figure 8.1: TODO: caption.

rate of 0 or even negative values, or resting heart rates approaching 150 bpm. Good thing we did this prior predictive simulation. This is exactly why you'd do such a thing; to see if the priors you assumed are realistic. Clearly the priors we're using can be improved.

How should we improve the priors? We have two choices here, we could tighten the normal distribution for the mean, and we could reduce the maximum value for the standard deviation. Either of those alone should tighten the distribution of resting heart rates implied by the priors. I'm going to suggest we do both. Let's tighten the normal prior for the mean to have a mean of 75 and standard deviation of 7.5 (implying a 95% probability that the mean resting heart rate is between 60 and 90; still probably a bit wide). Let's also trim the upper bound of the standard deviation to 15 bpm. Here's our revised statistical model:

$$
\begin{aligned}
r_i &\sim \text{Normal}(\mu, \sigma) \\
\mu &\sim \text{Normal}(75, 7.5) \\
\sigma &\sim \text{Uniform}(0, 15)
\end{aligned}
$$

Now let's see if that implies a more realistic distribution of resting heart rates:

```
#from https://bookdown.org/content/4857/geocentric-models.html#a-language-for-describi
n <- 10000

set.seed(123)

#randomly draw means from the prior
mu.sim <- rnorm(n, mean = 75, sd = 7.5)

#randomly draw SDs from the prior
sigma.sim <- runif(n, min = 0, max = 15)

#randomly draw values of resting heart rate from the combined means and SDs
r.sim <- rnorm(n, mean = mu.sim, sd = sigma.sim)

#plot the simulated distribution of heart rates
df <- data.frame(r.sim = r.sim)

ggplot(df, aes(x = r.sim)) +
  geom_density(fill = "lightblue", color = "darkblue") +
  labs(title = "Implied distribution of resting heart rate from priors",
       x = "Resting heart rate (bpm)",
       y = "Density") +
  theme_minimal()
```

Now we see no negative values implied by the priors, and the upper extreme values have been reduced to a maximum of around 125 bpm. Honestly these

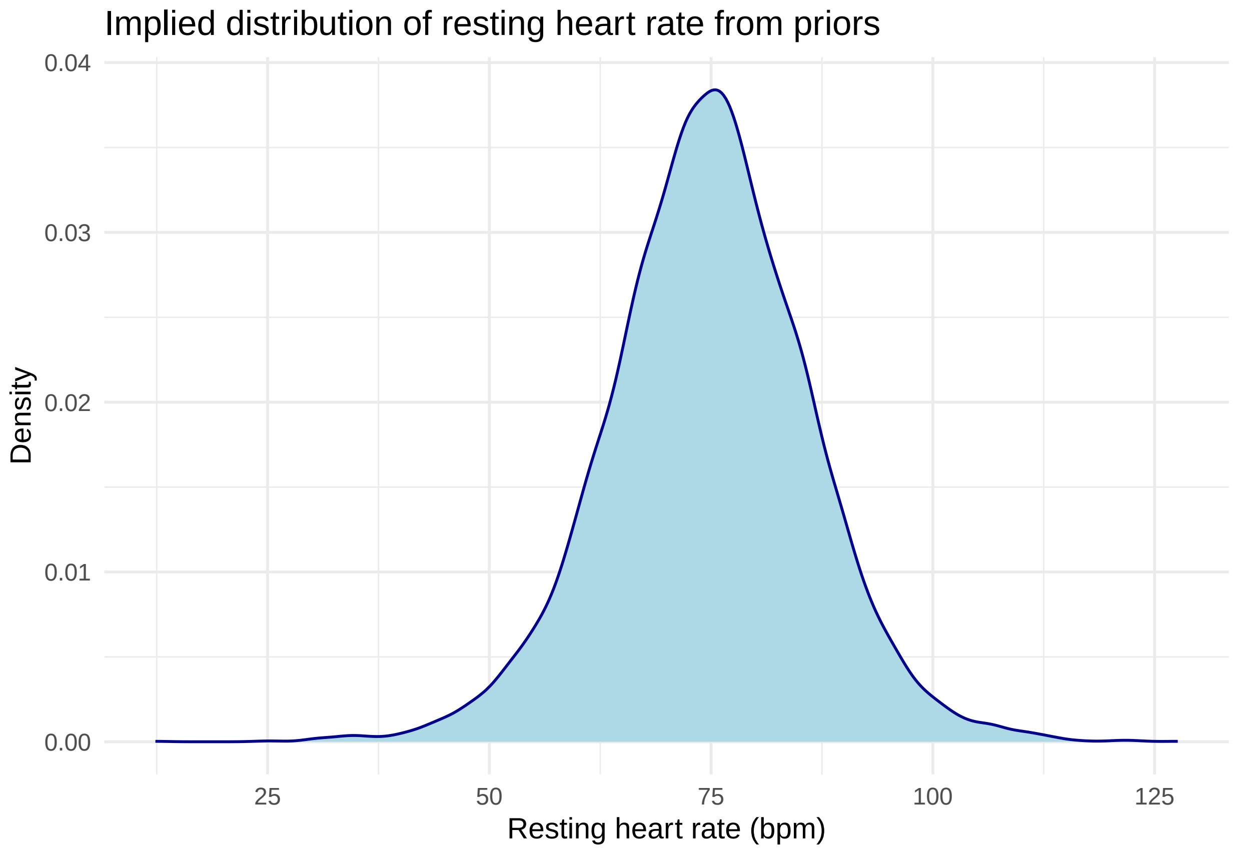


Figure 8.2: TODO: caption.

prior may still be a bit too wide, but those extreme values are very rare, and the data will have a lot to say about where the posterior distributions end up. Let's proceed!

8.4 Step 4: Fit the statistical model with the observed data

With our statistical model in defined, let's proceed to estimate the parameters in `brms`:

```r
library(brms)

#Likelihood model: heart rates normally distributed
m1.formula <- bf(Pulse ~ 1, family = gaussian())

# Priors:
#   Intercept (mu) ~ Normal(75, 10)
#   sigma ~ Uniform(0, 20)
m1.prior <- c(
  prior(normal(75, 7.5), class = "Intercept"),
  prior(uniform(0, 15), class = "sigma", lb = 0, ub = 15)
)

# Fit model
m1 <- brm(
  data = d,
  formula = m1.formula,
  prior = m1.prior,
  seed = 123,
  chains = 4, warmup = 1000, iter = 2000, cores = 4,
  refresh = 0
)
```

```
## Running /Library/Frameworks/R.framework/Resources/bin/R CMD SHLIB foo.c
## using C compiler: ‘Apple clang version 17.0.0 (clang-1700.0.13.3)’
## using SDK: ‘MacOSX15.4.sdk’
## clang -arch arm64 -std=gnu2x -I"/Library/Frameworks/R.framework/Resources/include" -
## In file included from <built-in>:1:
## In file included from /Library/Frameworks/R.framework/Versions/4.5-arm64/Resources/l
## In file included from /Library/Frameworks/R.framework/Versions/4.5-arm64/Resources/l
## In file included from /Library/Frameworks/R.framework/Versions/4.5-arm64/Resources/l
## /Library/Frameworks/R.framework/Versions/4.5-arm64/Resources/library/RcppEigen/inclu
##   679 | #include <cmath>
##       |          ^~~~~~~
## 1 error generated.
## make: *** [foo.o] Error 1
```

Let's walk through this. We've defined the likelihood with the `bf` function, where the formula `Pulse ~ 1` tells `brms` that we want to fit a model with only an intercept, which is just a fancy way of saying we want the mean of the normal distribution that we specified with `family = gaussian` (recall "Gaussian" is another name for the normal distribution). We then define our priors. In `brms`, each parameter getting a prior needs a distribution for the prior. You can see we specify `normal(75, 7.5)` for the mean and `uniform(0, 15)` for the standard deviation in this case. The `class` argument is required to indicate the type of variable, here being an `Intercept` for the mean and `sigma` for the standard deviation. Note that when defining a uniform prio, you not only need to define the limits within the `uniform` call, but also with the `lb` and `ub` arguments. Any inconsistencies here will lead to an error.

I'll just note right here that the process of fitting model can be much more complex than I'm suggesting in this section. For example, there are decisions to make about the particular MCMC engine to use, the number of chains to run, the length to run each chain (iterations), whether or not to thin the iterations, and how to fit the model efficiently using CPU's in parallel. For the purpose of an introductory statistics book, I've chosen to leave many of these details out and using relatively consistent model fitting decisions, especially for the relatively simple models addressed in a beginner's class. Thus you will almost always see that we fit models with four chains, 2000 iterations for each chain, 1000 iterations of warmup (more on that below), and with the chains fitted in parallel across four CPU's (`cores`) for efficiency. I've set the `seed` value so that the particular output in this book is reproducable, and I've set the `refresh` argument to 0 to minimize the amount of messages you get from `brms` during model fitting. All of these decisions can and should be open for discussion when fitting a particular model, and often these settings will need to be adjusted. But I consider these more advanced topics better approached when you have a basic handle on the general steps of Bayesian analysis, and other books cover these topics in detail and much more effectively than I could here.

8.5 Step 5: Evaluating and applying a fitted model

8.5.1 A quick glance at the estimates

After fitting the model, we print the model output (`print(m1)`) to see same basic numerical summaries of the posterior distributions for the parameters we estimated. Our estimate for the mean heart rate is listed under "Regression

Coefficients" as the "Intercept". Remember that our estimate is an entire probability distribution, and `brms` defaults to showing you the mean of the posterior as the "Estimate". Thus we see that the mean of the posterior distribution for resting heart rate is 72.57 bpm. The "Est.Error" represents the standard deviation of the posterior distribution, and we also see limits of a 95% credible interval (CI). Based on the analysis, we can say there's a 95% probability that the mean resting heart rate is between 72.30 bpm and 72.85 bpm.

In addition to the mean heart rate, we also get a posterior distribution for the standard deviation, here denoted "sigma" under "Further Distributional Parameters". There we see the posterior mean is 11.96 bpm, and there's a 95% probability that the standard deviation is between 11.77 and 12.15 bpm. Note that the credible intervals for the mean and standard deviation of resting heart rates are both quite narrow, indicating that we've estimated these parameters with a high degree of precision.

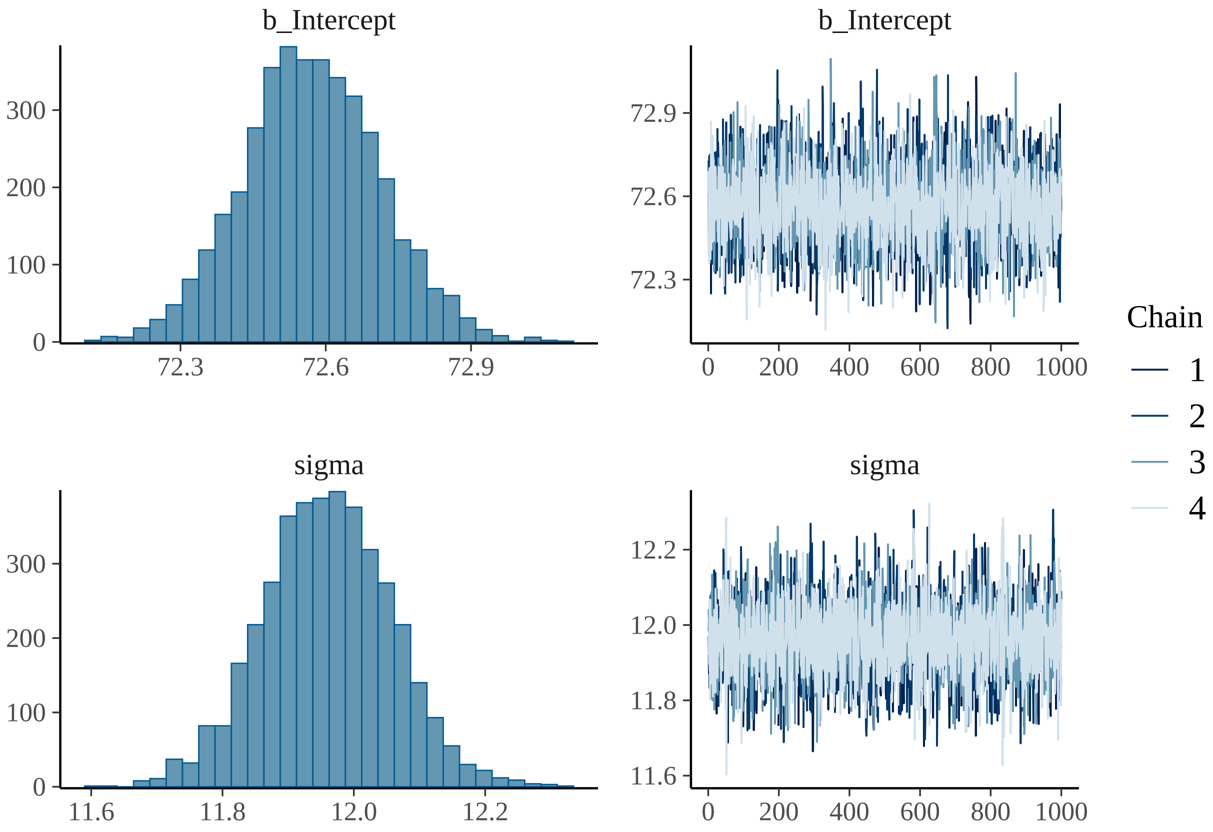
8.5.2 Checking for convergence

Before you dive into applying the posterior distributions for inferences, one thing you need to do is see if the sampling algorithm (the MCMC process) reached a stable set of posterior distributions. Remember the MCMC process is basically using a sampling process to estimate the posterior distributions, starting with some initial values that are supplied by the user or internally by the algorithm. Early in the sampling process the sampled values are biased by the initial values and can vary widely from iteration to iteration. This is called a **burn-in** or **warmup** period, and we generally do not want to use samples from this period for estimation. Indeed, the samples from the defined warmup period are entirely discarded. If the estimation process goes well, the values in each iteration stabilize and remain consistent, and it is these consistent values that we want to use for inference. We use the term **convergence** to describe a sampler that has reached the point of stability for a parameter. tIndeed, the samples from the defined warmup period are entirely discarded.

In our models, we run the sampling process for 2000 iterations with a warmup period of 1000 iterations. This means the first 1000 iterations are dropped entirely, and only the final 1000 iterations are retained. This is done in parallel across four chains, so we end up with a total of 4000 sampled values for each parameter.

Although we are discarding the first 1000 iterations entirely, we still need to check that the sampling process reached convergence in the final 1000 iterations we're retaining for inference. There are two ways we'll evaluate convergence in this book. First, you can execute the `plot` function on the object for your fitted model, which will produce a plot fo the posterior distributions for each parameter, as well as **trace plots** that can be used to evaluate convergence qualitatively:

```
plot(m1)
```



The left panel shows the posterior distributions for the mean (`b_Intercept`) and standard deviation (`sigma`) parameters, and the right side of the panel shows trace plots for each panel. Trace plots allow you to view the value of each parameter for each iteration of the model in each chain. What we're looking for here is relative consistency in the parameter values *within and among* the chains, that is convergence of the parameter values. Notice that for both of these parameters, the sampled values in sequence are relatively consistent, and sampled values across the chains converged around the same values (i.e., the chains are **well-mixed**). These trace plots indicate solid convergence because the values for each chain overlap extensively and hover around a common value.

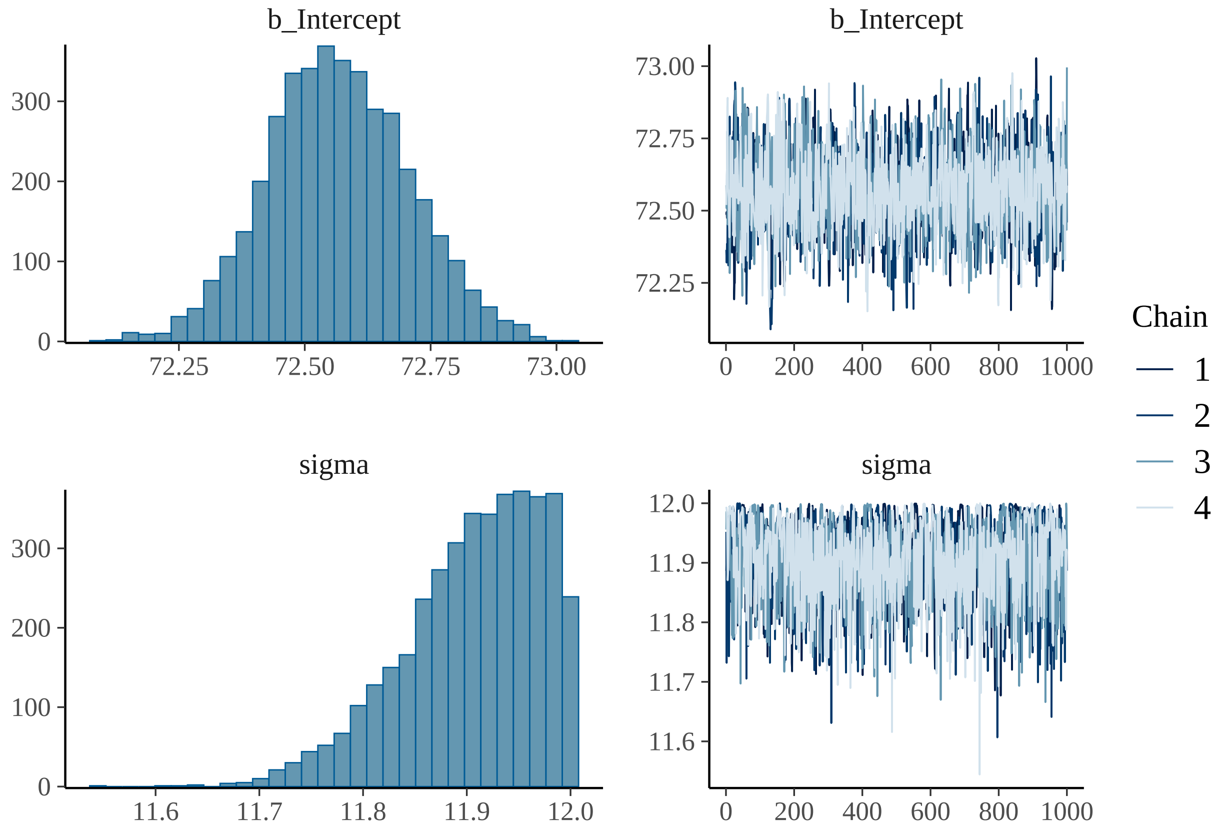
In addition to evaluating convergence qualitatively, we can also evaluate it numerically in the printed model output above. Note that for each parameter an `Rhat` value is reported. This is known as the **Gelman-Rubin statistic ($\hat{R}$), and it reflects the degree of convergence of each parameter. Values close to 1 imply convergence, and we genearlly want to see Rhat values less than 1.01. The Rhat values for our two parameters are reported as 1.00, implying convergence of our model.

8.6 Checking the shape of the posterior distribution

One other visual check you should make is of the posterior distributions, which can point to problems with the prior distributions (among other issues). For example, imagine that instead of using a uniform prior for the standard deviation between 0 and 15, I used a uniform prior between 0 and 12, which says the standard deviation can *only* take on values between 0 and 12. Values below 0 and above 12 are impossible.

Below is the output from the plot function for the fitted model using this new prior. Look closely at the posterior distribution for the standard deviation. Notice how the posterior distribution is strongly skewed to the left, with most of the values around 12 and a hard cut-off at 12? This is good evidence that the prior I used was too constraining, and that it's very possible that the standard deviation takes on values greater than 12. Basically the observed data via the likelihood are saying the standard deviation may well be above 12, but my prior distribution doesn't allow it. You can also see in the trace plot for the standard deviation that there's a hard limit at 12. These are good signs that my priors need to be relaxed and the model refit.

```
## Running /Library/Frameworks/R.framework/Resources/bin/R CMD SHLIB foo.c
## using C compiler: ‘Apple clang version 17.0.0 (clang-1700.0.13.3)’
## using SDK: ‘MacOSX15.4.sdk’
## clang -arch arm64 -std=gnu2x -I"/Library/Frameworks/R.framework/Resources/include"
## In file included from <built-in>:1:
## In file included from /Library/Frameworks/R.framework/Versions/4.5-arm64/Resources/
## In file included from /Library/Frameworks/R.framework/Versions/4.5-arm64/Resources/
## In file included from /Library/Frameworks/R.framework/Versions/4.5-arm64/Resources/
## /Library/Frameworks/R.framework/Versions/4.5-arm64/Resources/library/RcppEigen/inclu
##   679 | #include <cmath>
##       |          ^~~~~~~
## 1 error generated.
## make: *** [foo.o] Error 1
```



8.6.1 Inference and posterior predictive check

Finally, we should make a detailed inspection and summary of the posterior distributions and draw conclusions if all looks good. We start this process by using the `as_draws_df` function to retrieve the samples.

```
#posterior samples
m1.post <- as.data.frame(as_draws_df(m1))
head(m1.post)
```

```
##   b_Intercept    sigma Intercept    lprior      lp__ .chain .iteration .draw
## 1    72.60770 11.96995  72.60770 -5.692764 -28148.35      1          1     1
## 2    72.52242 11.87165  72.52242 -5.696455 -28148.70      1          2     2
## 3    72.48102 11.93042  72.48102 -5.698294 -28148.54      1          3     3
## 4    72.56183 11.83219  72.56183 -5.694733 -28149.06      1          4     4
## 5    72.62127 11.90600  72.62127 -5.692188 -28148.48      1          5     5
## 6    72.75178 11.89091  72.75178 -5.686821 -28149.33      1          6     6
```

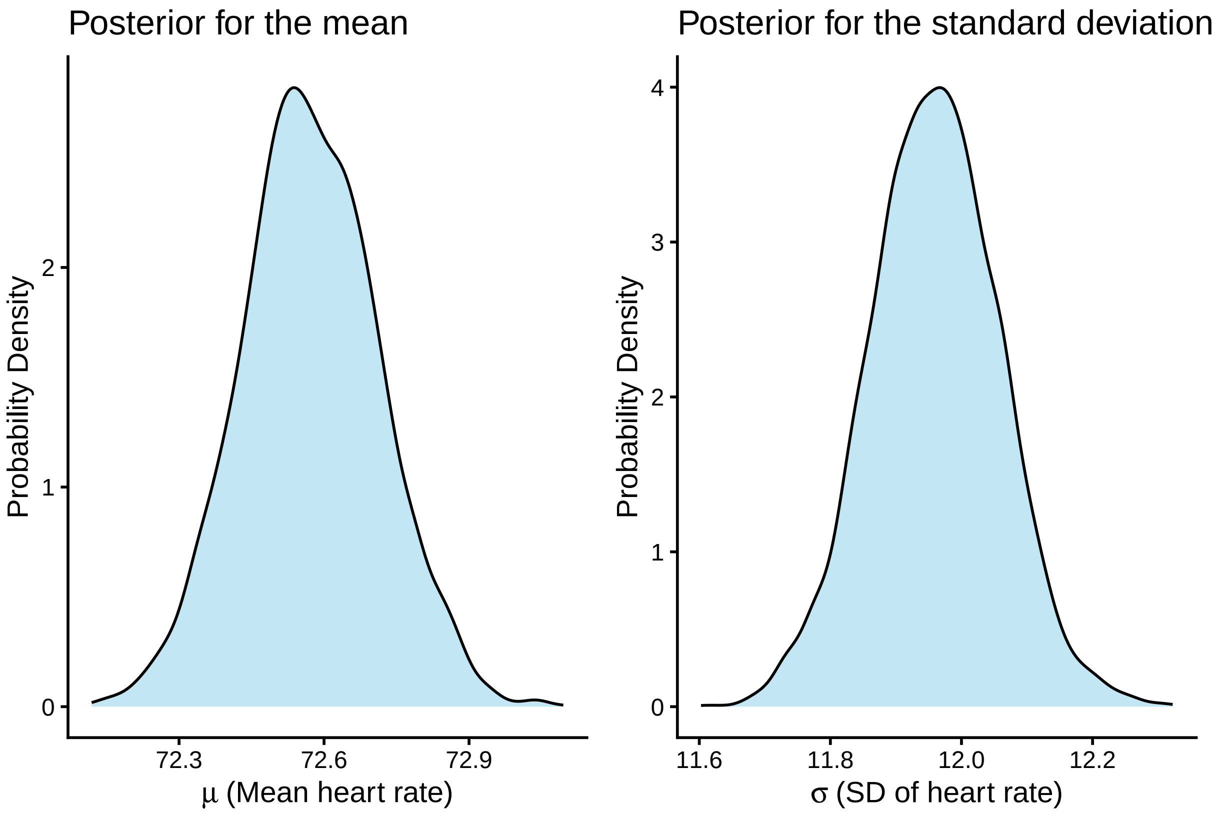
```
nrow(m1.post)
```

```
## [1] 4000
```

We've printed the first six rows, where we see values for the mean (`b_intercept`) and standard deviation (`sigma`). We also see that we have 4000 samples in total, exactly what we expect when we fit the model with 2000 iterations with 1000 warmup iterations discarded across four chains.

Now we can use the samples to plot the full posterior distribution for both parameters.

```
library(patchwork) #allows us to combine plots in one output

p_mu <- ggplot(m1.post, aes(x = b_Intercept)) +
  geom_density(fill = "skyblue", alpha = 0.5) +
  labs(x = expression(mu~"(Mean heart rate)"),
       y = "Probability Density",
       title = "Posterior for the mean") +
  theme_classic()

p_sigma <- ggplot(m1.post, aes(x = sigma)) +
  geom_density(fill = "skyblue", alpha = 0.5) +
  labs(x = expression(sigma~"(SD of heart rate)"),
       y = "Probability Density",
       title = "Posterior for the standard deviation") +
  theme_classic()

p_mu + p_sigma
```



To answer our initial question, our posterior dsitribution for the mean implies that the mean resting heart rate for adults in the U.S. is very likely around 72.5 bpm. We could of course compute some summary statistics:

```
#posterior mean
mean(m1.post$b_Intercept)
```

```
## [1] 72.57146
```

```
#posterior median
median(m1.post$b_Intercept)
```

```
## [1] 72.56715
```

```r
#interquartile range
IQR(m1.post$b_Intercept)
```

```
## [1] 0.1888912
```

```r
#90% credible interval
quantile(m1.post$b_Intercept, probs=(c(0.05, 0.95)))
```

```
##       5%      95%
## 72.34078 72.80674
```

Here we can be more specific and say that the mean of the posterior distribution for resting heart rate is 72.57 bpm, and the 90% credible interval says there's a 90% chance that the mean heart rate is between 72.34 and 72.81 bpm. That's a very high degree of precision, due in part to our large sample size of 7481 individuals.

We can also compute other probabilities of interest. For example, maybe you want to know the probability that the mean resting heart rate is greater than 73. That's easy to compute with the samples:

```r
#probability that the mean resting rate is >73 bpm
mean(m1.post$b_Intercept>73)
```

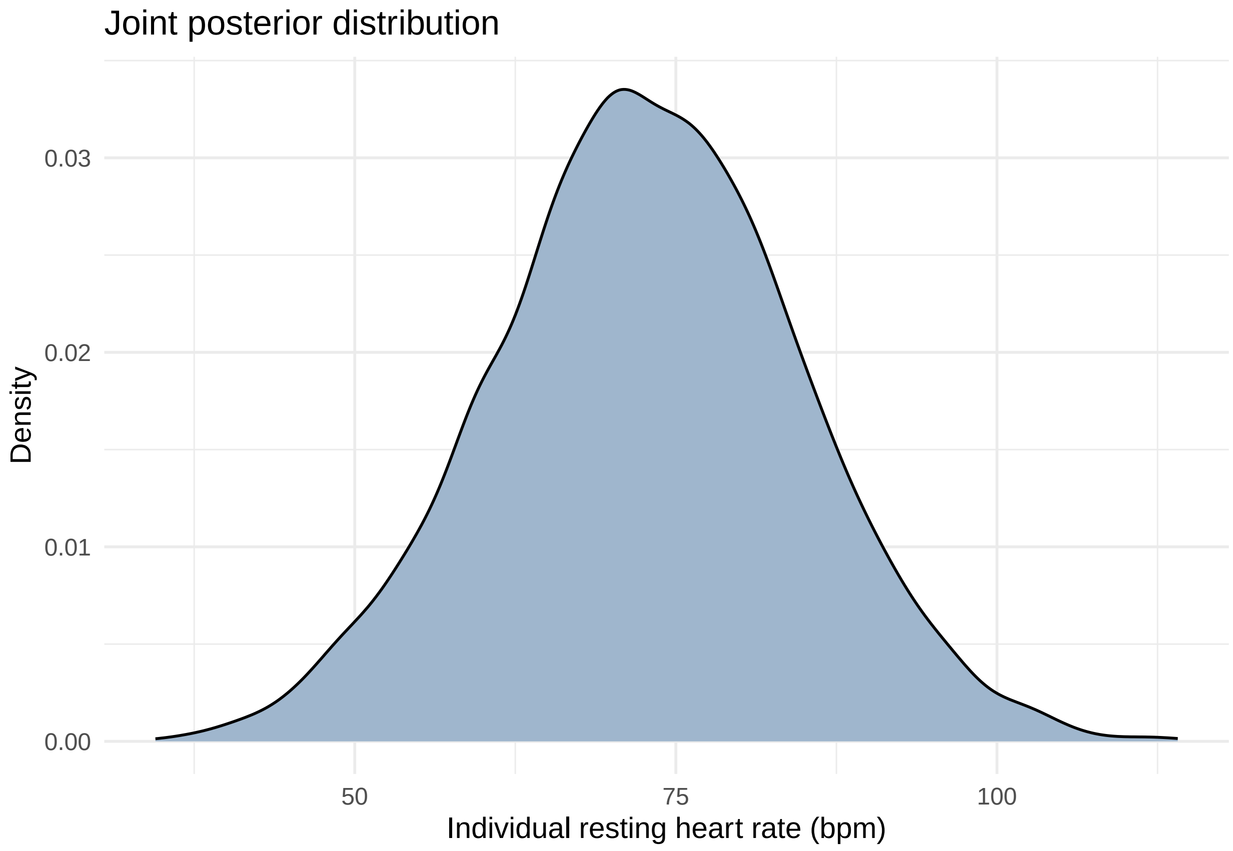
```
## [1] 0.00225
```

There's only a 0.2% chance that the mean resting heart rate is greater than 73 bpm.

Before reporting our conclusions, one final thing we should do is to check what the posterior distribution implies about the data (i.e., the individual heart rates). This is analogous to the prior predictive check, except now instead of using only the prior distribution to simulate the data, we use the posterior distribution. Let's go ahead and do this, drawing samples from the posterior distributions for the mean and standard deviation, and then randomly simulating observations of resting heart rate from the normal distributions of heart rate implied by the posterior.

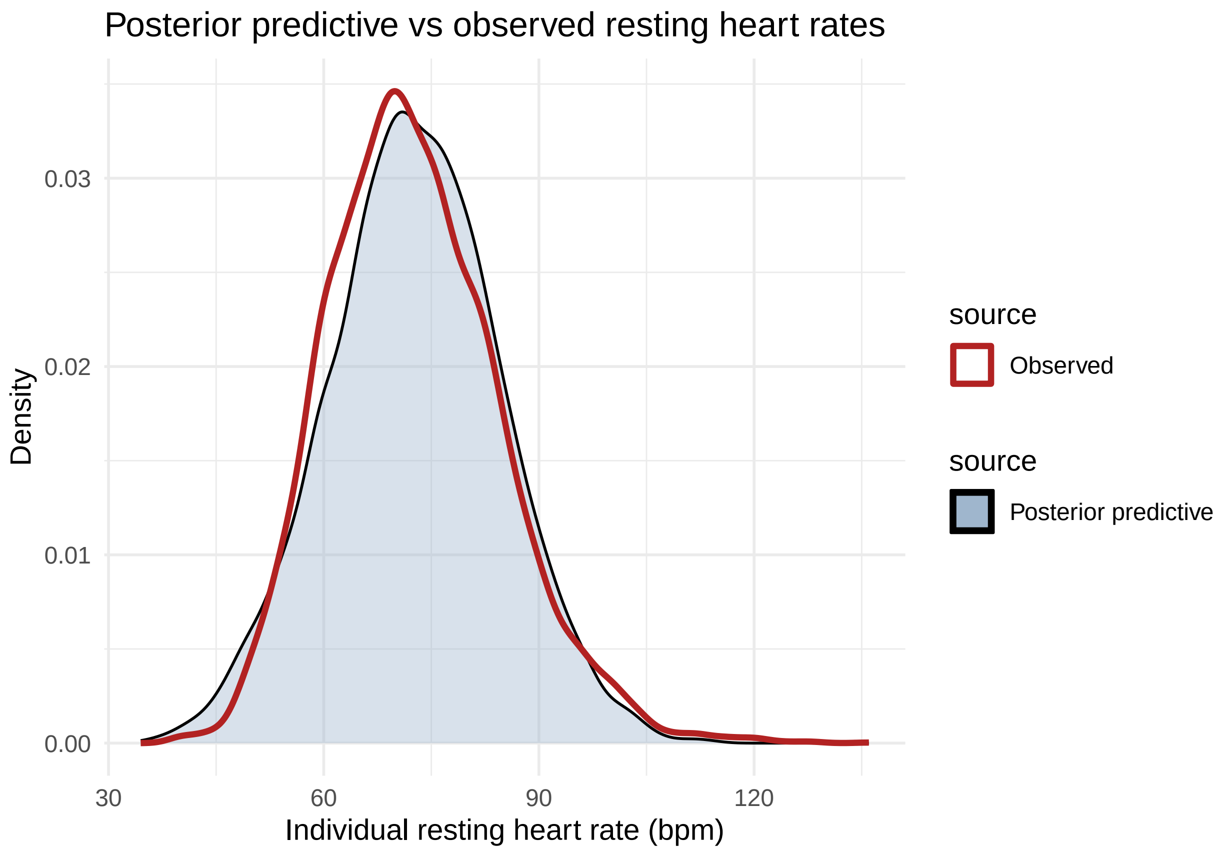
```r
set.seed(123)

# Simulate one individual heart rate per posterior sample
post_r <- rnorm(nrow(m1.post), mean = m1.post$b_Intercept, sd = m1.post$sigma)
```

```r
# Create data frame for plotting
df_post <- data.frame(pulse = post_r)
df_obs  <- data.frame(pulse = d$Pulse)

ggplot(df_post, aes(x = pulse)) +
  geom_density(fill = "slategray3", color = "black") +
  labs(title = "Joint posterior distribution",
       x = "Individual resting heart rate (bpm)",
       y = "Density") +
  theme_minimal()
```



The resulting distribution is called the **joint posterior distribution**, and it describes the distribution of the *individual* resting heart rates. One thing that would make us feel confident with our model is if the distribution of individual heart rates implied by the model reflects the data we actually observed. We can visually check that by overlaying a density plot of the observed data on top of the joint posterior distribution:

```r
# Organize the two data sources
df_pred <- data.frame(pulse = df_post$pulse, source = "Posterior predictive")
df_obs  <- data.frame(pulse = d$Pulse, source = "Observed")

# Overlay densities
library(ggplot2)
ggplot() +
  geom_density(data = df_pred, aes(x = pulse, fill = source),
               alpha = 0.4, color = "black") +
  geom_density(data = df_obs, aes(x = pulse, color = source),
               linewidth = 1.1, show.legend = TRUE) +
  scale_fill_manual(values = c("Posterior predictive" = "slategray3")) +
  scale_color_manual(values = c("Observed" = "firebrick", "Posterior predictive" = "bla
  labs(title = "Posterior predictive vs observed resting heart rates",
       x = "Individual resting heart rate (bpm)",
       y = "Density") +
  theme_minimal()
```



Here we can see that the posterior predictive and observed distributions have a close fit! That means the resting heart rates implied by the posterior distribution do a good job at describing the observed data. We can be reasonably confident in the model output for inference. Visually that would be the area under the posterior predictive distribution that's greater than 100, but we can easily compute that probability with the observations of heart rate randomly drawn from our model:

```r
mean(df_post$pulse>100)
```

```
## [1] 0.01175
```

We see that based on our statistical model, we expect 1.15% of the population has tachycardia. We could do the same thing for bradycardia, which is defined as a slow heart rate usually less than 60 bpm:

```
mean(df_post$pulse<60)
```

```
## [1] 0.14575
```

There's a much greater portion of the population that likely has bradycardia than tachycardia. This isn't too surprising, however, because bradycardia is common in people who exercise a lot, indicating high conditioning.

8.7 Summary

That's it! There are five basic components to Bayesian workflow, and we will try to make a habit of working through each of these components as we wrestle with new questions and models in the remainder of the book. Because the primary goal of this book is to teach the basic elements of statistic elements, we may not always cover each of these steps in the exact order or with the same degree of detail with every new example and model. But I do believe the workflow outlined here should be a starting point for anyone just starting out with statistical analysis, taking basic steps to ensure the model specification is adequate before drawing conclusions.

Chapter 9

The linear model

- change this chapter to use the BAC example in HW9

Over the last few chapters we have examined the process of estimating unknown quantities from samples with frequentist and Bayesian inference. Sample estimates vary in their quality, and we've emphasized the importance of quantifying metrics of uncertainty for our estimates, such as credible or confidence intervals. To this point the quantities we have estimated are very simple, such as a single proportion or mean characterizing a population. But what happens when life isn't so simple? What if we're not simply interested in the prevalence of an infection in a single city, but rather variation in the prevalence of the infection among cities? What if the variation in the prevalence among cities is affected by population density?

The same reasoning applies to population means for continuous variables. Suppose you are a real estate broker, and you're interested in the average price at which a home sells. It would be inadequate to estimate a single mean price for all homes, because you know home price is affected by a variety of factors, such as square footage. Thus, what you really need to do is estimate the mean home price at different levels of square footage.

In this chapter we will start developing a tool for situations like this where we want to link a mean (or a proportion) to other measurements. The tool we will use is the **generalized linear model**. The GLM is the powerhouse of statistical analysis. It's flexible enough to allow us to estimate the simplest of quantities, such as a single mean or proportion, but also to link a response variable to other measurements. We can use GLMs to examine the relationship between two variables in a simple experiment, and we can use to examine the relaitonship between two variables while adjusting for other variables, often a necessity in observational designs. We can use the GLM for situations where we expect a relationship to be constant, or in situations where the relationship between variables depends on a third variable. The bottom line is that GLMs are extremely flexible, and as such, they will be the focus of the remainder of this book.

In light of the criticisms of frequentist inference with null hypothesis significance testing, all the examples in the remainder of the book will be presented initially with Bayesian estimation procedures. However, because students of statistics should know how to interpret studies using frequentist inference, each example will include a short "How a Frequentist Would Analyze It" section.

9.1 Statistical models

The rest of this book is about designing statistical models to estimate quantities of interest. A **model** is just a simplified representation of some phenomenon of interest. We've already seen models in the form of DAGs, which represent our scientific model of how variables are causally related to each other. As scientific models, DAGs are largely conceptual in nature. Science is about confronting our ideas with data, and so we need a model to help us make that link. That's where **statistical models** enter the picture. Statistical models are quantitative representations of our scientific models. They can consist of an equation, or a set of equations, that describe single variables, and more often in causal inference, the relationship between variables.

Let's develop the idea of a statistical model with an example. In this chapter we will look at the the growth of perennial ryegrass, which is native to Europe and Asia but has been cultivated and introduced around the world. Ryegrass can be considered an invasive species in that it can outcompete native plants. We're going to look at the growth rate of ryegrass as measured in the lab, using data from (Inderjit et al. 2002).

Let's go ahead and load the data. This is actually only a subset of N = 9 observations. You'll see there are two variables in the data frame, `conc` and `rootl`, and for now we'll focus our attention on `rootl`, which is the root length of ryegrass measured in cm.

```
d <- read.csv("data/ryegrass_sub.csv")
print(d)
```

```
##      rootl conc
## 1 8.355556 0.94
## 2 6.914286 0.94
## 3 7.750000 0.94
## 4 6.871429 1.88
## 5 6.450000 1.88
## 6 5.922222 1.88
## 7 1.925000 3.75
## 8 2.885714 3.75
## 9 4.233333 3.75
```

```
ggplot(data = d, aes(x = rootl)) +
  geom_histogram(binwidth = 0.5, fill = "skyblue", color = "black") +
  labs(x = "Root length (cm)", y = "Frequency") +
  theme_minimal()
```

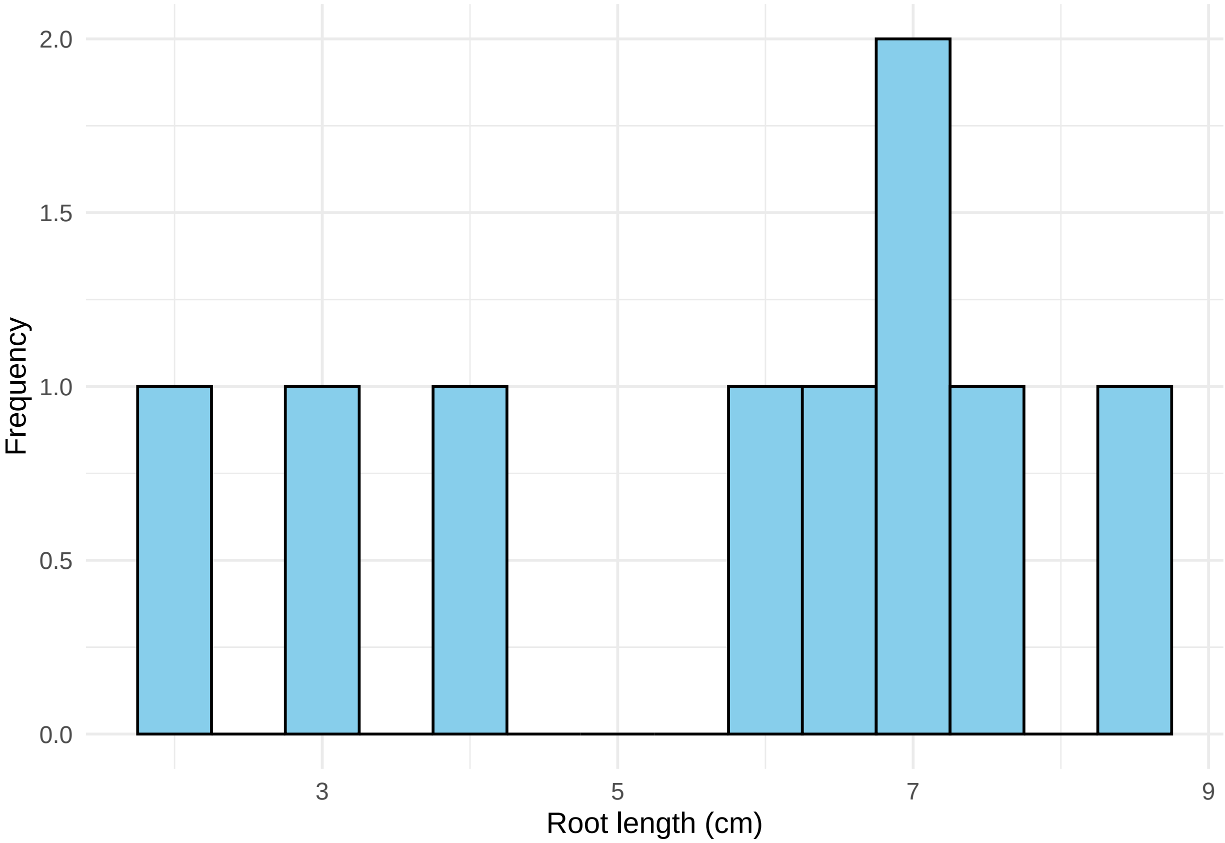


Figure 9.1: TODO: caption.

The histogram doesn't tell us much with only nine observations, but we do know root length is a continuous random variable. Root length in this dataset is being used to measure the growth of ryegrass in the lab. Growth rates and plant size typically exhibit bell-shaped, approximately normal distributions. Indeed, any attribute that is influenced by many processes with non-trivial effects (e.g., many genes and environmental factors) tends to exhibit an approximately normal distribution. So we will generate a simple statistical model that describes root length with a normal distribution.

We will follow the approach outlined in McElreath (2020) to define statistical models, which involves:

1. Identify the observed variables. These are the data we collect from samples.
2. Identify the unobserved parameters. These are the unknown quantities we wish to estimate with the data.
3. Define how the observed data are generated. This can be a simple as defining a single variable as a random variable from a particular probaiblity distribution, or we can define variables in terms of other variables.
4. For Bayesian analysis, define our prior distributions for each unknown parameter.

Let's apply this to our ryegrass example. To start, we are simply describing the observed root length data as a random variable drawn from a normal distribution. The normal distribution has two parameters that we need to estimate: the mean and the standard deviation. Thus, in our model, we need a line to describe root length as a random variable, and two lines to define the prior distributions for each of the unknown parameters. Here's the model we will use:

$$\begin{aligned} r_i &\sim \text{Normal}(\mu, \sigma) \\ \mu &\sim \text{Normal}(5, 2) \\ \sigma &\sim \text{Uniform}(0, 5) \end{aligned}$$

Let's walk through each line of our statistical model. First we have $r_i \sim \text{Normal}(\mu, \sigma)$. Here we are defining the observed values of root length for each individual i (r_i) as a random variable following a normal distribution with parameters mean $= \mu$ and standard deviation $= \sigma$. Remember the tilde symbol (~) defines a relationship as **stochastic**, which means that the observed values r_i are probabalistic rather than being determiend with certainty. The normal distribution with its parameters defines the probability of drawing particulare values of root length. We don't know what those parameter values are, so we will have to estimate them. In the context of Bayesian estimation, this line represents the likelihood.

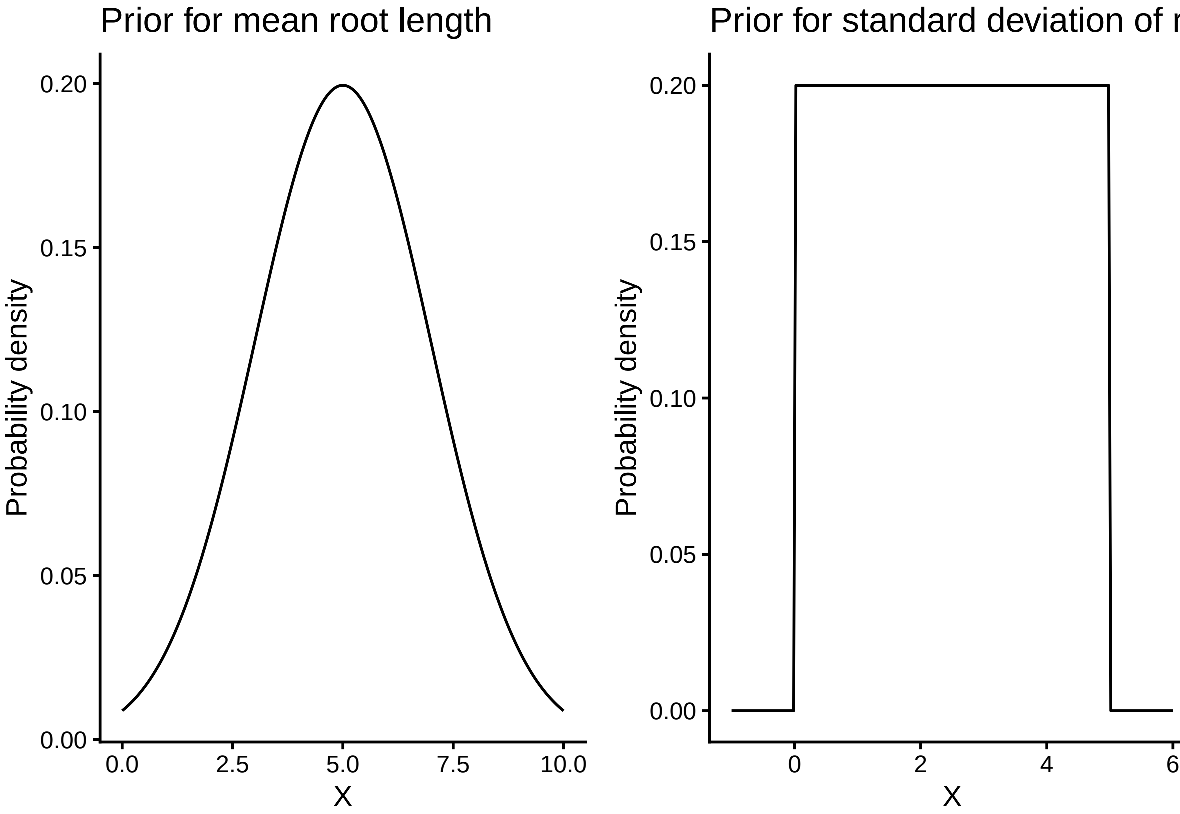
The second and third lines are prior distributions. In Bayesian estimation, every parameter in the statistical model requires a prior. Even though we might be more interested in the mean root length than its variation, we're assuming root length has a normal distribution, and the normal distibution has two parameters (mean and standard deviation). When we assume a variable is drawn from a particular probability distribution, we have to estimate the parmaeters for that probability distribution whether we want them or not. That's why it's useful to differentiate between types of parameters, the estimands being the parameters that we are most interested in.

Now consider what each prior distribution says. The first prior is for the mean root length: $\mu \sim \text{Normal}(5, 2)$. This defines the probability of the mean taking on different values with a normal distribution, specifically a normal distribution with a mean of 5 and standard deviation of 2. Effetively what this means is that - prior to analyzing the data - I think the most plausible values for the mean root lenght are around 5 cm. The mean could be greater or less than 5 cm, but 5 cm is the most likely value (as the mean of the normal distribution). Of course we can say exatly how likely the other values are. Following hte empricail rule, I'm assuming that there's a 95% chance that the mean root length is between 1 and 9 cm. Values outside those bounds collectively have only a 5% probability.

How did I know to use those particular values for the normal prior (mean = 5, standard deviation = 2). This is where domain knowledge is helpful. Based on prior knowledge from people who have worked with ryegrass in this kind of experimental setting (growing plants in petri dishes), we know the mean root lengths are going to be relatively small, likely somewhere in the range of 0-10 cm. The values at the extremes of that range are much less likely than values in the middle - indeed, it's not even possible to have a root length of 0 cm. The normal distribution captures that prior knowledge nicely (albeit imperfectly, as every model is imperfect).

Remember that Bayesian inference combines the prior and likelihood to quantify the posterior distributions for each a parameter. We're going to use `brms` to do just that, but before doing so, it can be very useful to do that's called a **prior predictive simulation**. The idea is that we can use the prior distributions to simulate data to get a sense for what the prior distributions imply about what the data should look like. For the ryegrass example, the idea is to get a sense for the different possible combinations of the mean and standard deviation of root length, and the resulting distribution of root length implied by the priors:

```r
#from https://bookdown.org/content/4857/geocentric-models.html#a-language-for-describing-models
n <- 10000

set.seed(123)

#randomly draw means from the prior
mu.sim <- rnorm(n, mean = 5, sd = 2)

#randomly draw SDs from the prior
sigma.sim <- runif(n, min = 0, max = 5)

#randomly draw values of root length from the combined means and SDs
r.sim <- rnorm(n, mean = mu.sim, sd = sigma.sim)
```


Prior for mean root length

Prior for standard deviation of r

Figure 9.2: TODO: caption.

```
#plot the simulated root lenth distribution
df <- data.frame(r.sim = r.sim)

ggplot(df, aes(x = r.sim)) +
  geom_density(fill = "lightblue", color = "darkblue") +
  labs(title = "Implied distribution of root length from priors",
       x = "Root length (cm)",
       y = "Density") +
  theme_minimal()
```

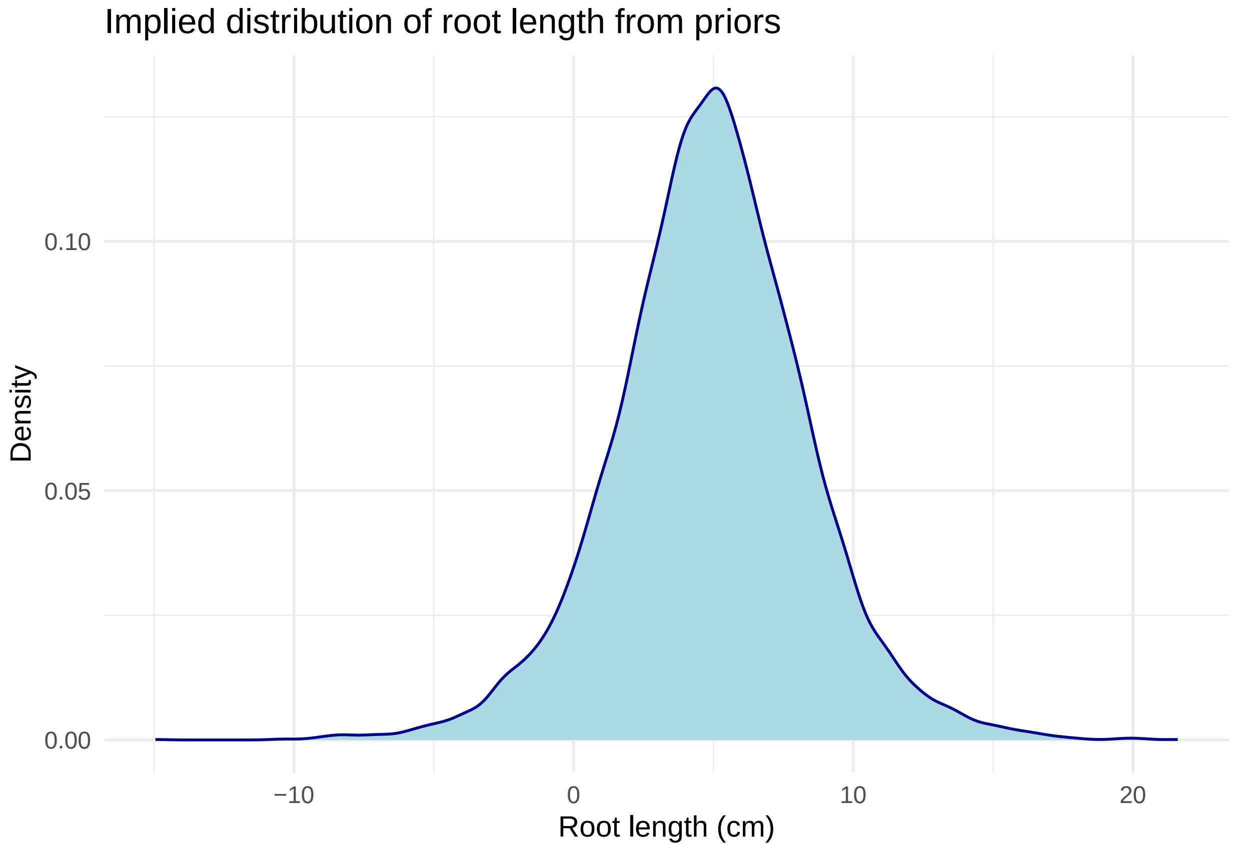


Figure 9.3: TODO: caption.

We can see the distribution of root length implied by the priors has a mean right around 5 cm as expected, so that's good. But what's not good is that the variation around that mean is unrealistic. The priors are implying a non-trivial chance of seeing root lengths that are 0 or negative! Good thing we did this prior predictive simulation. This is exactly why you'd do such a thing; to see if the priors you assumed are realistic. Clearly the priors we're using can be improved. Let's tighten things up with a revised statsitical model:

$$
\begin{aligned}
&r_i \sim \text{Normal}(\mu, \sigma) \\
&\mu \sim \text{Normal}(5, 1) \\
&\sigma \sim \text{Uniform}(0, 3)
\end{aligned}
$$

Do you see what changed? The normal prior for the mean now implies a 95% chance of the mean being 3-7 cm, and we've reduced the upper bound of the standard deviation from 5 cm to 3 cm. Let's see if that implies a more realistic distribution of heights:

```r
#from https://bookdown.org/content/4857/geocentric-models.html#a-language-for-describi
n <- 10000

set.seed(123)

#randomly draw means from the prior
mu.sim <- rnorm(n, mean = 5, sd = 1)

#randomly draw SDs from the prior
sigma.sim <- runif(n, min = 0, max = 3)

#randomly draw values of root length from the combined means and SDs
r.sim <- rnorm(n, mean = mu.sim, sd = sigma.sim)

#plot the simulated root lenth distribution
df <- data.frame(r.sim = r.sim)

ggplot(df, aes(x = r.sim)) +
  geom_density(fill = "lightblue", color = "darkblue") +
  labs(title = "Implied distribution of root length from priors",
       x = "Root length (cm)",
       y = "Density") +
  theme_minimal()
```

There are still some negative values implied by the priors, but now they are quite rare. One of the reasons we continue to see some very small proportion of negative root lengths is that we're assuming a normal distribution for root length, and the normal distribution can have negative values. As we'll see later in the book, there are other probability distributions that may be more effective. In this case, it would be helpful to use a probability distribution that has a bell-shaped curve, but that does not allow negative values. Stay tuned. For now, this prior distribution will suffice. Let's proceed with estimation in `brms`:

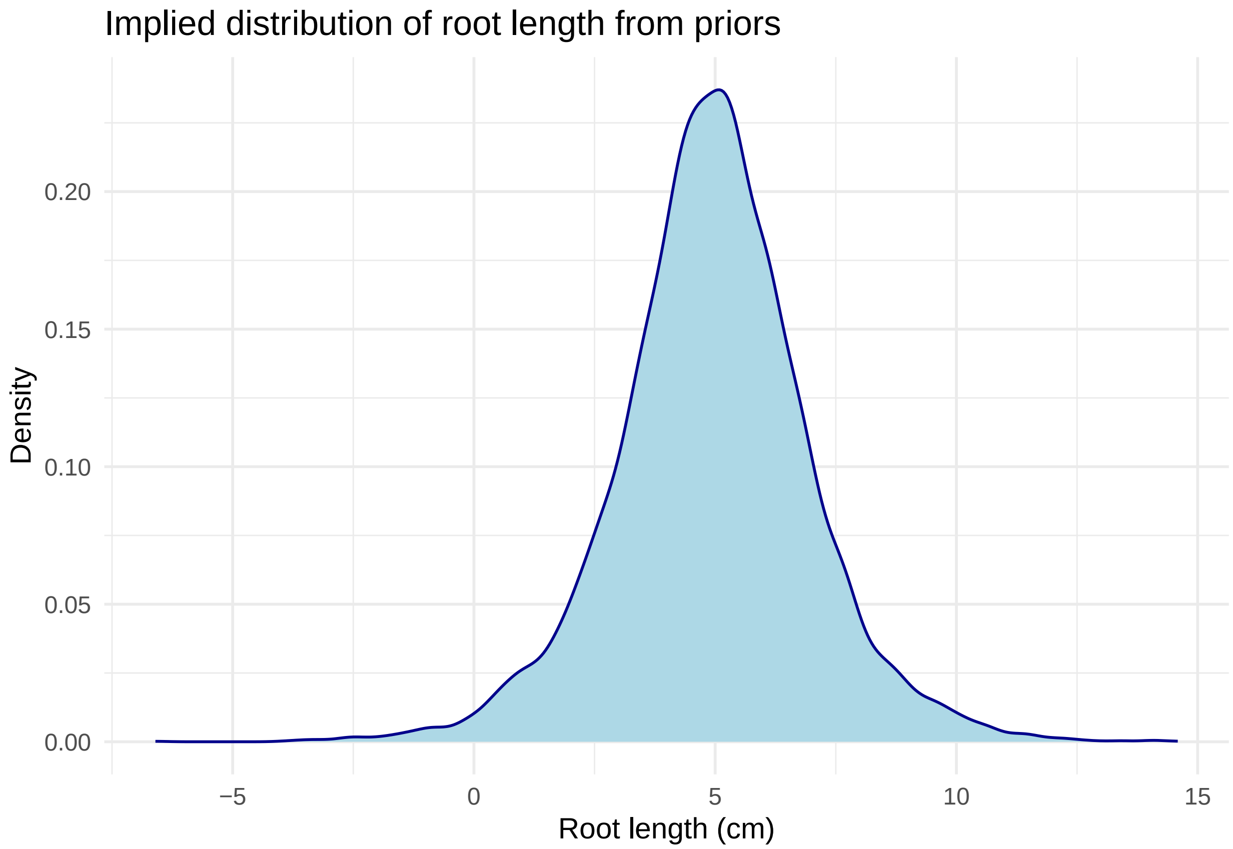


Figure 9.4: TODO: caption.

```r
#specify model formula
m1.formula <- bf(rootl ~ 1,
                family = gaussian) #defines root length as a normal random var

#specify priors
m1.prior <- c(prior(normal(5, 1), class = Intercept),
             prior(uniform(0, 3), class = sigma, lb=0, ub=3))

#compute the posterior
m1 <- brm(data = d,
         formula = m1.formula,
         prior = m1.prior,
         refresh = 0,
         seed=123)
```

```
## Running /Library/Frameworks/R.framework/Resources/bin/R CMD SHLIB foo.c
## using C compiler: ‘Apple clang version 17.0.0 (clang-1700.0.13.3)’
## using SDK: ‘MacOSX15.4.sdk’
## clang -arch arm64 -std=gnu2x -I"/Library/Frameworks/R.framework/Resources/include" -
## In file included from <built-in>:1:
## In file included from /Library/Frameworks/R.framework/Versions/4.5-arm64/Resources/l
## In file included from /Library/Frameworks/R.framework/Versions/4.5-arm64/Resources/l
## In file included from /Library/Frameworks/R.framework/Versions/4.5-arm64/Resources/l
## /Library/Frameworks/R.framework/Versions/4.5-arm64/Resources/library/RcppEigen/inclu
##   679 | #include <cmath>
##       |           ^~~~~~~
## 1 error generated.
## make: *** [foo.o] Error 1
```

```r
print(m1)
```

```
##  Family: gaussian
##   Links: mu = identity
## Formula: rootl ~ 1
##    Data: d (Number of observations: 9)
##   Draws: 4 chains, each with iter = 2000; warmup = 1000; thin = 1;
##          total post-warmup draws = 4000
##
## Regression Coefficients:
##           Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## Intercept     5.45      0.59     4.28     6.63 1.00     2069     1990
##
## Further Distributional Parameters:
##       Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
```

```
## sigma     2.25      0.40     1.51     2.94 1.00     1568     1287
##
## Draws were sampled using sampling(NUTS). For each parameter, Bulk_ESS
## and Tail_ESS are effective sample size measures, and Rhat is the potential
## scale reduction factor on split chains (at convergence, Rhat = 1).
```

Let's walk through this. We've defined the likelihood with the `bf` function, where the formula `rootl ~ 1` tells `brms` that we want to fit a model with only an intercept, which in this case is a fancy way of saying we want the overall meam (this will make more sense later this chapter!). We're assuming `rootl` is a normal random variable with `family = gaussian`. We then define our priors. In `brms`, each parameter getting a prior needs a distribution for the prior (`normal` for the mean, `uniform` for the stanard deviation in this case), and we specify the type of parameter with the `class` argument, here being an `Intercept` for the mean and `sigma` for the standard deviation. After fitting this model, we see the mean of the posterior distribution for root length is 5.45, and the 95% credible interval is 4.28-6.63. In other words, there's a 95% probability that the mean root length is between 4.28 and 6.63. We can execute the `plot` function on our model object to see a graph of the posterior distributions and a plot that helps us diagnose whether the model is converged:

```r
plot(m1)
```

The left panel shows the posterior distributions for the mean (`b_Intercept`) and standard deviation (`sigma`) parameters, and the right side of the panel shows **trace plots** for each panel. Trace plots allow you to view the value of each parameter for each iteration of the model in each chain. What we're looking for here is relative consistency in the parameter values among the chains, that is **convergence** of the parameter values. These trace plots indicate solid convergence because the values for each chain overlap extensively and hover around a common value. Numerically, the `Rhat` values near 1 also indicate convergence.

Remember that we can also draw samples from the posterior distribution to compute any quantity of interest. For example, suppose we want to estimate the probability that the mean root length is greater than 5 cm. We just need to extract the samples from the posterior and find the proportion of values greater than 5 for the mean:

```r
#posterior samples
m1.post <- as.data.frame(as_draws_df(m1))
head(m1.post)
```

```
##   b_Intercept    sigma Intercept    lprior      lp__ .chain .iteration .draw
## 1    5.051680 2.925563  5.051680 -2.018886 -25.07558      1          1     1
```

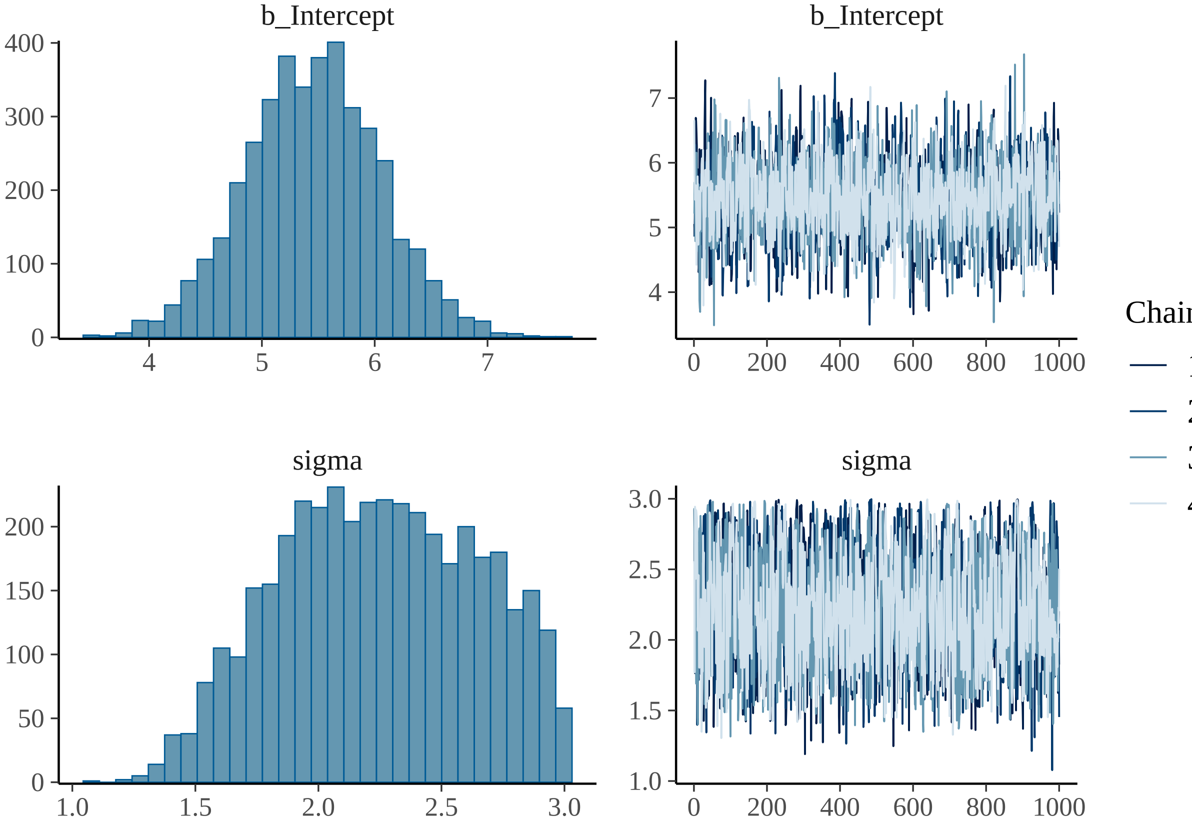


Figure 9.5: TODO: caption.

```
## 2    4.882915 2.430660  4.882915 -2.024405 -22.87511      1          2     2
## 3    5.939409 1.763796  5.939409 -2.458796 -22.51164      1          3     3
## 4    5.695199 2.873104  5.695199 -2.259202 -24.50022      1          4     4
## 5    6.694626 2.061946  6.694626 -3.453430 -24.31155      1          5     5
## 6    6.645777 2.094947  6.645777 -3.371841 -24.11954      1          6     6
```

```
#probability that the proportion infected is greater than 10%
mean(m1.post$b_Intercept > 5)
```

```
## [1] 0.77825
```

We see there's a 78% chance that the mean root length is >5 cm.

9.2 Linear model

9.2.1 Basic structure of the linear model

As it turns out, the researchers who generated the data we just analyzed were not simply interested in describing the mean growth rate of ryegrass. Because ryegrass can be invasive, they were interested in understanding the effect of a new herbicide on ryegrass growth. Each petri dish with ryegrass was randomly assigned an herbicide concentration, and they measured root length as an index of plant growth. This is a simple experimental design in which all other resource levels were controlled (e.g., water, light, nutrients). Because there were no concerns about post-treatment bias (e.g., non-random dropout), we can represent the scientific model with a simple DAG:

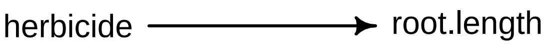


Figure 9.6: Initial DAG for the causal effect of greenspace on mental health.

Recall that our DAG is a scientific model of factors affecting root length. Like any model, it's a simplified representation of the system in that we are proposing the only non-trivial cause of root length variation is herbicide concentration. In reality we know there are plenty of other factors that affect plant growth - water, light, nutrients, etc. Those factors were controlled in this experiment, such that the different petri dishes were assigned identical levels of resources regardless of herbicide concentration. Because of measurement error, the resource levels won't be perfectly identical. Some petri dishes will inevitably receive a few microliters more or less of water, for example. That variation may well affect root length, but because the impact is expected to be so miniscule, we leave causes like that out of the DAG. Again, models are simplified representations of reality.

Given our scientific model, how should we analyze the data? We need a statistical model that captures the nature of the relationship between root length and herbicide concentration. As we saw in Chapter 3, when we have two variables that are quantitative, we can use a scatterplot to visualize the association between those variables. Let's start there:

```r
ggplot(d, aes(x = conc, y = rootl)) +
  geom_point() +
  labs(,
    x = "Herbicide concentration (mM)",
    y = "Root Length (cm)"
  ) +
  scale_y_continuous(limits = c(0, 10)) +
  theme_classic()
```

We can see there were three levels of herbicide concentration assigned to three replicates for nine total observations. Visually it looks like there is a negative relationship between root length and herbicide concentration, root length decreases as herbicide concentration increases. We need a statistical model to estimate that association. Linear models are extremely useful for this task.

In our initial statistical model, we assumed that the root length values were drawn from a common normal distribution with a single mean and standard deviation. The scatterplot above suggests that's not a good assumption. It looks like the average root length is high when little herbicide is applied, whereas average root length is high when a lot of herbicide is applied. We need a statistical model that allows the root length values to be drawn from distributions that have different means, where the mean root length depends on the herbicide concentration. Let's revise the statistical model to do just that:

$$
\begin{aligned}
&r_i \sim \text{Normal}(\mu_i, \sigma)\\
&\mu_i = \alpha + \beta x_i\\
&\alpha \sim \text{Normal}(5, 2)\\
&\beta \sim \text{Normal}(0, 1)\\
&\sigma \sim \text{Uniform}(0, 3)
\end{aligned}
$$

Again let's walk through the components:

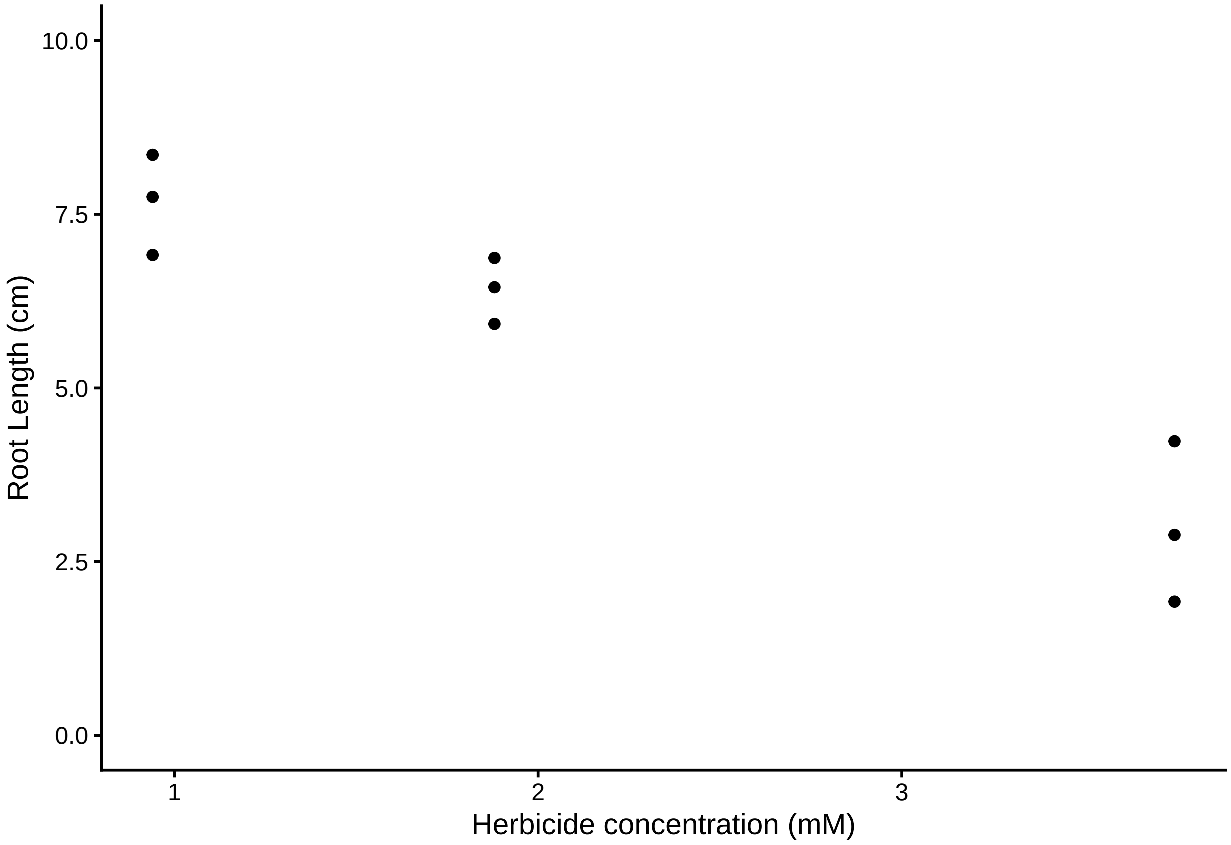


Figure 9.7: TODO: caption.

- **Likelihood** ($r_i \sim \text{Normal}(\mu_i, \sigma)$: Remember the likelihood is simply the probability of the data given the parameter values. How likely are the observed root length values given the mean and standard deviation for the ryegrass distribution? But there's one big change here. Did you notice the mean parameter now has a subscript (*i)*? Rather than assuming we have a single normal distribution that describes all the root length values, we're now saying that the observed root length for individual i is drawn from a unique normal distribution with its own mean.

- **Linear model** ($\mu_i = \alpha + \beta x_i$): Our first linear model! This model defines how the unique mean for each individual i is determined. The model says the mean root length for each individual i is a linear function of two parameters: an intercept (α) and a slope (β). You might remember from basic algebra that a line is defined by a simple equation $y = mx+b$. That's exactly what we have here, just with different labels for the parameters. What is each component of the linear model saying? The slope (β) represents the expected change in the mean root length (μ_i) for each 1-unit change in x_i. The value x_i here represents the herbicide concentration that each individual i receives. Thus, the slope represents the expected change in the mean root length when the herbicide concentration increases by 1 mM. The intercept (α) represents the expected mean root length when the herbicide concentration is 0. That should make sense. When $x_i = 0$, the slope term in the linear model simply drops out, leaving just the intercept. Together, the intercept and slope determine the mean root length for the distributions from which each observed root length i is drawn.

- **Priors:** Remember that each parameter in a statistical model must have a prior distribution, reflecting our belief about the possible values of those parameters prior to analyzing the data. We may not be interested in all the parameters in the model. If our research question is whether the herbicide affects root length, the estimand is the slope (β), but we still need to estimate the other parameters to estimate the slope. Let's walk through the prior for each parameter:

    - **Intercept** (α): Here we are specifying a normal prior for the intercept with a mean of 5 and standard deviation of 2. That reflects our prior belief that there's a 95% probability the mean root length is between 1 and 10 cm when there's no herbicide applied. In this case we increased the standard deviation of the prior back from 1 to 2 to allow for a wider prior distribution given that we are no longer estimating the grand mean, but the mean when no herbicide is applied.
    - **Slope** (β): What is the expected change in mean root length with a 1 mM increase in herbicide concentration? Our normal prior with mean = 0 and standard deviation = 1 implies that the most likely value of the slope is 0, and there's a 95% chance that the change in root length is between -2 and 2 cm as herbicide concentration increases

by 1 mM. Biologically, the researchers very likely expect the root length will decrease as herbicide concentration increases. If much of literature supports such a negative effect, then it may be wise to use a prior that has more probability weighted towards negative effects. On the other hand, this is a new herbicide, and there's a question about whether it works, so it's possible that there's no effect. Given that possibility, we choose to center the prior around 0. Even if we center the prior for the slope around 0, we would still want to use an appropriate standard deviation to limit the range of slopes to values we think are realistic. For example, it wouldn't make sense to allow for a slope that allows a 100-cm change in root length per one unit increase in mM, when the range in root length values is 0-20 cm. So we restrict the variance in the prior distribution to a range of effect we think is plausible.

- **Standard deviation** (σ). The standard deviation represents the expected variation in root length values around the expected mean. Any deviation in root length from the expected mean predicted by herbicide concentration represents variation that can't be explained by herbicide concentration. For example, imagine the expected mean root length is 4 cm when the herbicide concentration is 1 mM. Not every plant with 1 mM herbicide applied will have exactly 4 cm root length. There will be deviations around the expected mean of 4 cm. Those deviations are called **residual errors** (or just "residuals"). Residual variation is always expected in systems that have multiple causes. Some of the variation around the expected mean based on herbicide concentration could be due to minor variation in water, light availability, or other resources. Some of the residual variation may simply be measurement error, and some may simply be random, not having obvious causes). The standard deviation parameter specifies the expected magnitude of the variation in observed root length round the expected mean based on herbicide concentration. Because there's no i subscript on teh standard deviation, we're assuming a common magnitude of residual error no matter what the mean root length may be. In this case, we've retained the same uniform prior distribution that we used in our more simple analysis that assumed a common normal distribution for all values of root length.

9.2.2 Fitting the linear model in brms

Let's go ahead and use a prior predictive simulation to see what the priors imply about the relationship between root length and herbicide concentration. Here all we do is simulate values of the intercept and slope from the priors, then plot them. I've limited the number of simulations to N = 100 to ensure that we can visualize the lines in the resulting graph.

```
n <- 100

set.seed(123)

#intercept
alpha.sim <- rnorm(n, mean = 5, sd = 2)

#slope
beta.sim <- rnorm(n, mean = 0, sd = 1)

#values of x (herbicide concentration)
x_vals <- seq(0, 4, length.out = 100)

# Create a data frame with all lines
lines_df <- expand.grid(x = x_vals, id = 1:n) %>%
  mutate(y = alpha.sim[id] + beta.sim[id] * x)

# Plot using ggplot
ggplot(lines_df, aes(x = x, y = y, group = id)) +
  geom_line(alpha = 0.3, color = "blue") +  # Adjust transparency and color
  theme_minimal() +
  labs(x = "Herbicide concentration (mM)",
       y = "Root length (cm)") +
  theme_classic()
```

The graph of the prior predictive simulations shows the priors allow for both positive and negative relationships between root length and herbicide concentration. The priors also allow for a range in the expected rooth length when the herbicide concentration is 0 (the intercept). One thing I don't like about these priors is that in some simulations they allow for relationships with negative expected values of root length, which of course is not possible. But the vast majority of the simulations appear realistic, and for our purposes this is sufficient to combine with the data to estimate the posterior.

To estimate the posterior in `brms`, we need to use the formula to specify the linear model. The formula we'll use here is `rootl ~ 1 + conc`, where `1` represents the intercept, and `conc` represents the slope for the effect of herbicide concentration. We use the `+` operator to add the slope for `conc` to the model. Then we just need to make sure each parameter has a prior. Slope parameters are denoted `class = b` when defining priors in `brms`. Here's the code to estimate the posterior:

```
#specify model formula
m2.formula <- bf(rootl ~ 1 + conc,
                 family = gaussian) #defines root length as a normal random var

#specify priors
```

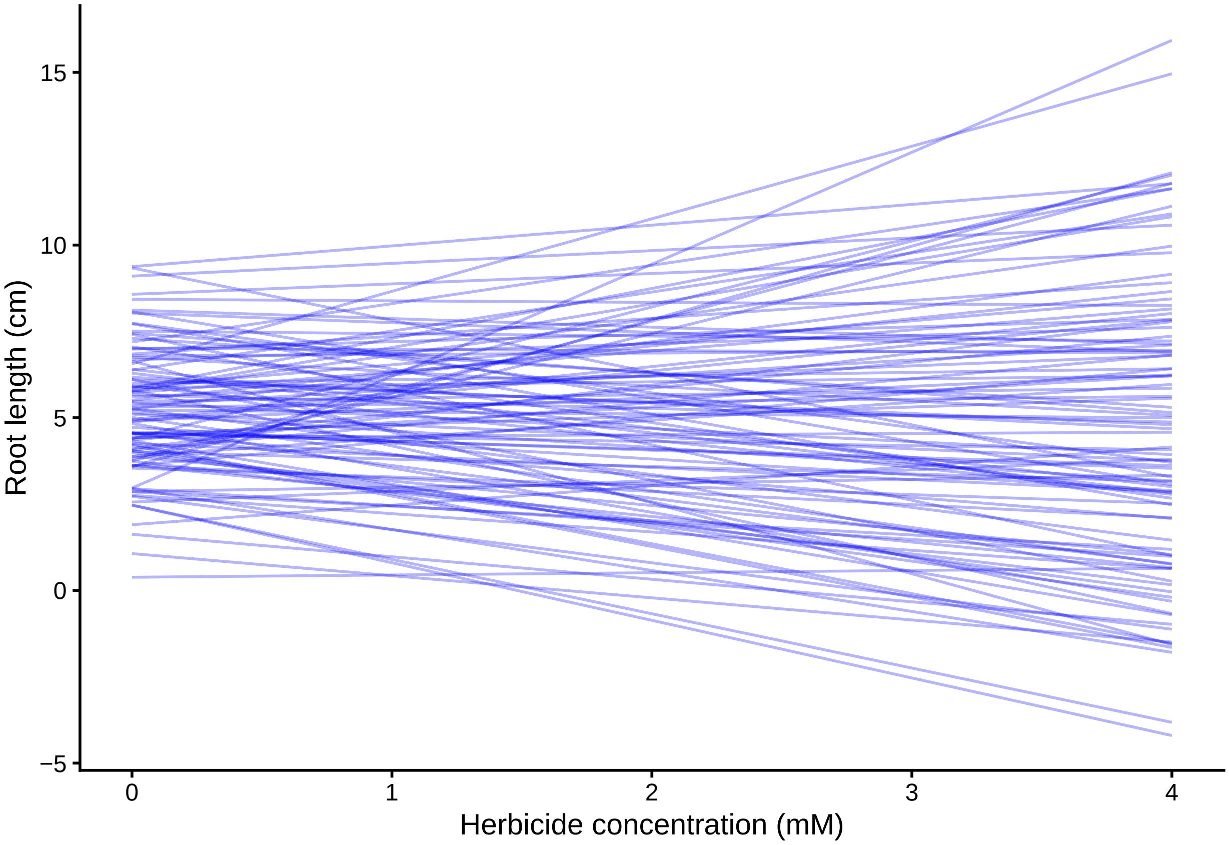


Figure 9.8: TODO: caption.

```
m2.prior <- c(prior(normal(5, 2), class = Intercept),
              prior(normal(0, 1), class = b),
              prior(uniform(0, 3), class = sigma, lb=0, ub=3))

#compute the posterior
m2 <- brm(data = d,
         formula = m2.formula,
         prior = m2.prior,
         refresh = 0,
         seed=123)
```

```
## Running /Library/Frameworks/R.framework/Resources/bin/R CMD SHLIB foo.c
## using C compiler: ‘Apple clang version 17.0.0 (clang-1700.0.13.3)’
## using SDK: ‘MacOSX15.4.sdk’
## clang -arch arm64 -std=gnu2x -I"/Library/Frameworks/R.framework/Resources/include" -
## In file included from <built-in>:1:
## In file included from /Library/Frameworks/R.framework/Versions/4.5-arm64/Resources/l
## In file included from /Library/Frameworks/R.framework/Versions/4.5-arm64/Resources/l
## In file included from /Library/Frameworks/R.framework/Versions/4.5-arm64/Resources/l
## /Library/Frameworks/R.framework/Versions/4.5-arm64/Resources/library/RcppEigen/inclu
##   679 | #include <cmath>
##       |          ^~~~~~~
## 1 error generated.
## make: *** [foo.o] Error 1
```

```
print(m2)
```

```
##  Family: gaussian
##   Links: mu = identity
## Formula: rootl ~ 1 + conc
##    Data: d (Number of observations: 9)
##   Draws: 4 chains, each with iter = 2000; warmup = 1000; thin = 1;
##          total post-warmup draws = 4000
##
## Regression Coefficients:
##           Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## Intercept     9.05      0.74     7.32    10.36 1.00     2046     1725
## conc         -1.54      0.30    -2.06    -0.86 1.00     2056     1547
##
## Further Distributional Parameters:
##       Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## sigma     1.00      0.36     0.55     1.93 1.00     1615     1636
##
## Draws were sampled using sampling(NUTS). For each parameter, Bulk_ESS
```

```
## and Tail_ESS are effective sample size measures, and Rhat is the potential
## scale reduction factor on split chains (at convergence, Rhat = 1).
```

How do we interpret the output? Remember that the summary for `brms` models shows the mean (`Estimate`), standard deviation (`Est.Error`) and 95% credible interval (`l-95% CI` and `u-95% CI`) for each parameter. We're also given metrics to evaluate whether the parameters have converged to consistent values, an `Rhat` near 1 implying convergence. Based on the summary output, we can see the mean of the posterior for the intercept is 9.05, implying the most likely value of root length is 9.05 cm when no herbicide is applied. The line for `conc` supplies summary statistics for the slope for the effect of herbicide concentration on root length. We see the most likely value is -1.54, implying that for every one unit increase in herbicide concentration, root length declines by 1.54 cm on average. Notably the 95% credible interval is (-2.06, -0.86), suggesting the posterior distribution for the slope is broadly negative. We can confirm as much by plotting the posterior distributions:

```r
plot(m2)
```

Indeed, we see a posterior distribution that is entirely negative for the slope. This is strong evidence that the herbicide has a negative effect on root length, and that the suppression of plant growth increases with increasing herbicide concentration. We can also see from the traceplots that there is excellent convergence of the three parameters in our model (the third being the residual error).

Plots of the posterior distributions and tables summarizing those posterior distributions are helpful for summarizing the output for simple models like ours. But usually it's even more helpful to visualize the output of our model graphically. Our primary interest is in the relationship between root length and herbicide concentration, so we should make a plot that shows what the posterior distribution implies about that relationship. Lets re-create our scatterplot for root length and herbicide concentration, but now we'll add a line to the graph representing the association between root length and herbicide based on the posterior means. To do so, we're going to first use the `fitted` function to predict the expected values of root length for different values of herbicide concentration. Let's start with that:

```r
#values of herbicide concentration for which to predict root length
x <- seq(min(d$conc), max(d$conc), length.out=100)

#predict values of root length for each value of herbicide
y <- fitted(m2, newdata=data.frame(conc = x))
fit <- cbind.data.frame(conc = x, y)
head(fit)
```

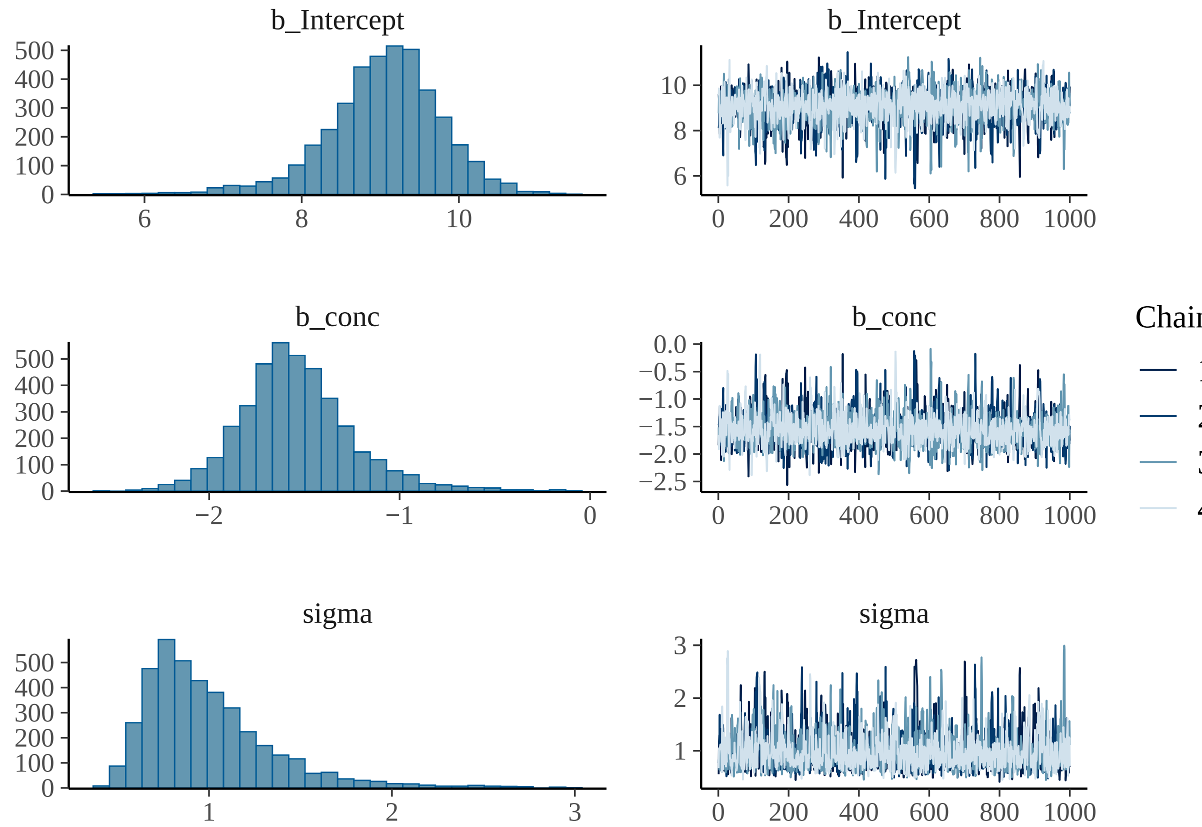


Figure 9.9: TODO: caption.

```
##        conc Estimate Est.Error     Q2.5    Q97.5
## 1 0.9400000 7.598360 0.5021902 6.480100 8.530201
## 2 0.9683838 7.554646 0.4959052 6.456589 8.478159
## 3 0.9967677 7.510932 0.4896868 6.433725 8.428845
## 4 1.0251515 7.467218 0.4835377 6.410807 8.375627
## 5 1.0535354 7.423504 0.4774606 6.383898 8.315237
## 6 1.0819192 7.379791 0.4714582 6.354950 8.265914
```

What is the `fitted` function doing? It's taking each value of `x` and plugging it into the linear model formula to compute the expected mean root length. For example, when herbicide concentration is $x = 0.94$, the expected mean root length is $y = 9.05 - 1.54 * 0.94 = 7.6$ based on the posterior mean for the intercept (9.05) and slope (-1.54). But remember with Bayesian inference the estimate is not a single point, but an entire distribution. The `fitted` function computes the expected mean root length from the values of x across every sample for the posterior distribution, and it provides summary statistics of the variation around the posterior mean, namely the standard deviation (`Est.Error`) and a 95% credible interval (`Q2.5` and `Q97.5`).

Now let's make our scatterplot and add the posterior mean predictions. All we do here is take the code for our original scatterplot, and we add the function `geom_line` to add the prediction line:

```r
ggplot(d, aes(x = conc, y = rootl)) +
  geom_point() +
  geom_line(data = fit, aes(x = conc, y = Estimate)) +
  labs(,
    x = "Herbicide concentration (mM)",
    y = "Root Length (cm)"
  ) +
  scale_y_continuous(limits = c(0, 10)) +
  theme_classic()
```

It's worth bearing in mind that this prediction line represents the posterior mean, so it doesn't communicate the uncertainty about our estimate. To see what I mean, let's add prediction lines for a selection of the posterior samples to get a sense for the variation To do so, we need to first extract the samples with `as_draws_df`:

```r
m2.post <- as_draws_df(m2)
head(m2.post)
```

```
## # A draws_df: 6 iterations, 1 chains, and 6 variables
##   b_Intercept b_conc sigma Intercept lprior lp__
## 1         9.6   -1.8  0.56       5.6   -5.4  -17
```

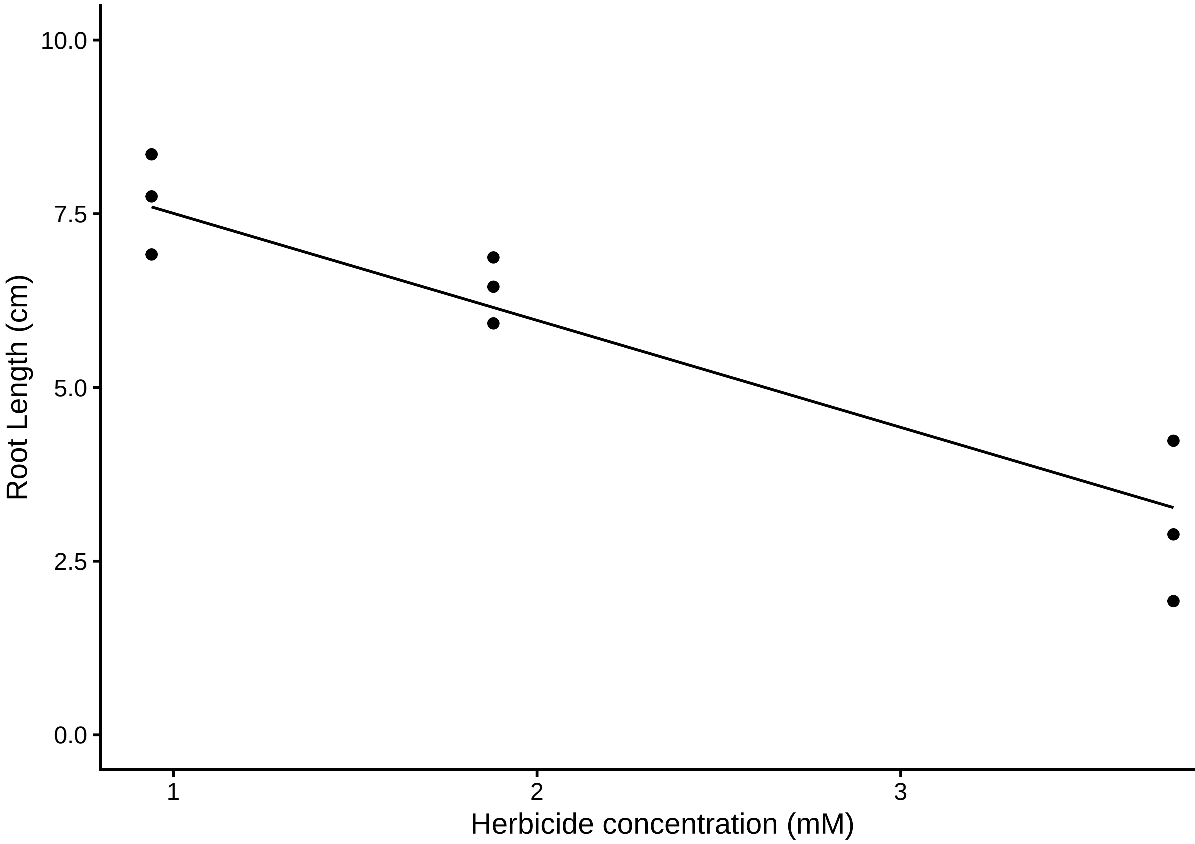


Figure 9.10: TODO: caption.

```
## 2         9.4   -1.5  1.18       6.2   -4.9  -17
## 3         8.2   -1.3  1.30       5.4   -4.5  -18
## 4         7.8   -1.1  0.88       5.4   -4.3  -18
## 5         9.8   -1.9  0.81       5.7   -5.4  -16
## 6         9.7   -1.8  1.02       5.8   -5.3  -16
## # ... hidden reserved variables {'.chain', '.iteration', '.draw'}
```

```
ggplot(d, aes(x = conc, y = rootl)) +
  geom_abline(data = m2.post,
              aes(intercept = b_Intercept, slope = b_conc),
              linewidth=0.1, alpha=0.1) +
  geom_line(data = fit, aes(x = x, y = Estimate), color="firebrick") +
  geom_point(color="firebrick") +
  labs(,
    x = "Herbicide concentration (mM)",
    y = "Root Length (cm)"
  ) +
  scale_y_continuous(limits = c(0, 10)) +
  theme_classic()
```

This plot shows the prediction line for every single posterior sample from our model. I leveraged the `geom_abline` function to make this easy, as it automatically adds a line to the graph based on a supplied intercept and slope. I made the line width small and the lines transparent (`alpha = 0.3`) so that you could see the the points, the posterior mean prediction, and where most of the lines are concentrated. What we see is that the posterior mean prediction is in the center of the cluster, and the variation around it represents uncertainty. The more variation in the predictions from individual samples, the more uncertainty we have about the estimated relationship. Including predictions from each draw of the posterior can effective for displaying uncertainty, but an alternative would be to plot a credible interval at a particular level of probability around the posterior mean prediciton. Here's the same plot but with a 95% credible interval for the prediction line:

```
ggplot(d, aes(x = conc, y = rootl)) +
  geom_smooth(data = fit,
              aes(x = conc, y = Estimate, ymin = Q2.5, ymax = Q97.5),
              stat = "identity",
              fill = "grey70", color = "black", alpha = 1, linewidth = 1/2) +
  geom_point(color="firebrick") +
  labs(,
    x = "Herbicide concentration (mM)",
    y = "Root Length (cm)"
  ) +
  scale_y_continuous(limits = c(0, 10)) +
```

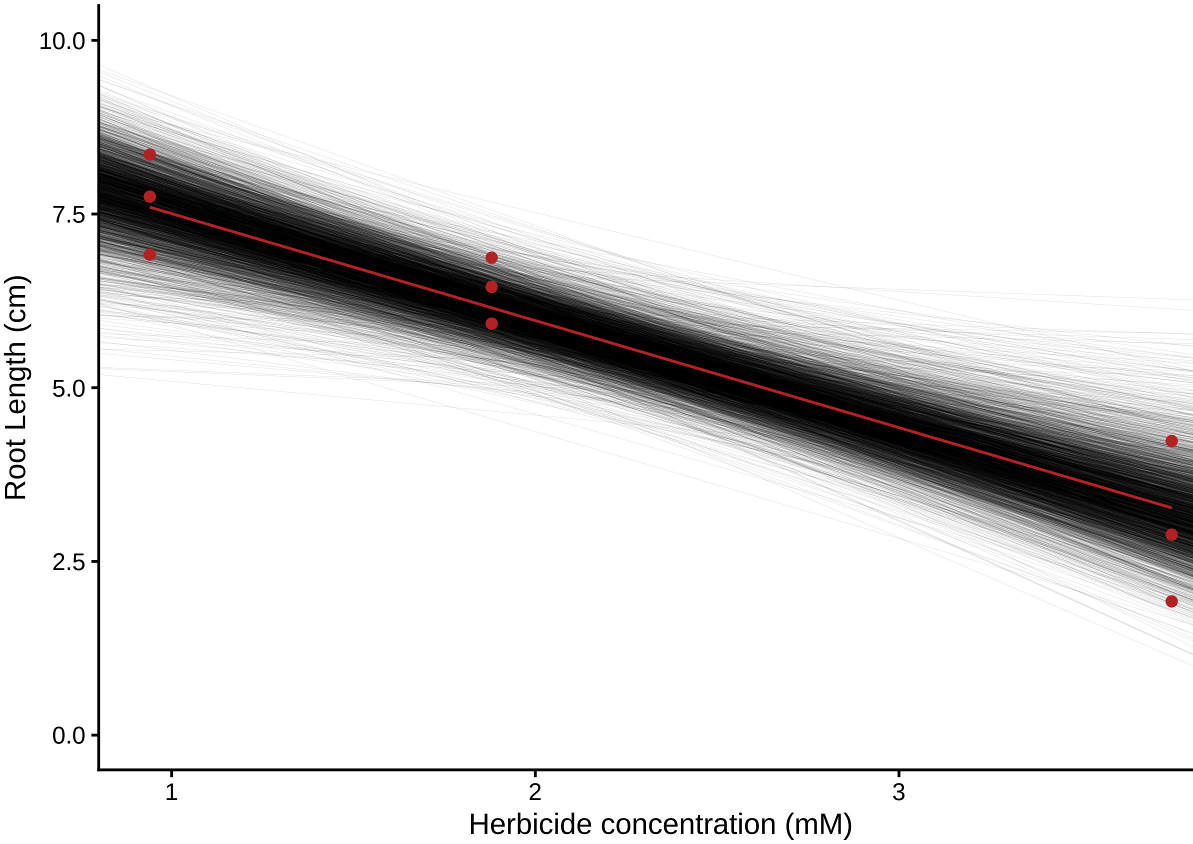


Figure 9.11: TODO: caption.

```
  theme_classic()
```

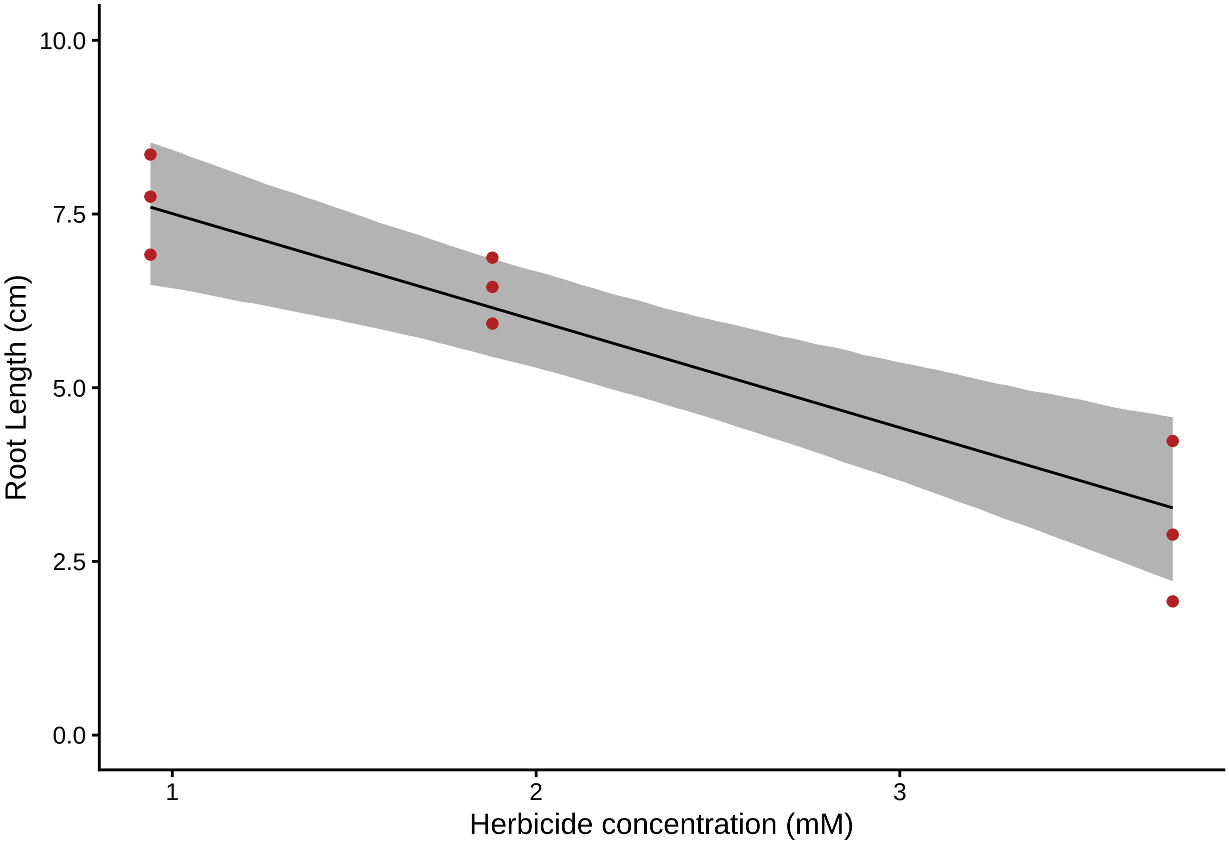


Figure 9.12: TODO: caption.

Here we're drawing on the predictions we made with the `fitted` function, which computed the 95% credible interval. The `geom_smooth` function adds the prediction line and a shaded area for the 95% credible interval.

It's important to note that the variation in the prediction lines and the credible interval we just plotted both represent uncertainty about the *expected mean* value of the response variable at each value of the explanatory variable. In the context of our statistical model, these represent uncertainty about the values of μ_i. What if we wanted to represent uncertainty about the observed values of the response variable? We can clearly see that not all the points are exactly the same as the mean root length at a given herbicide concentration. There is residual variation in root length unexplained by herbicide concentration. How do we represent uncertainty about the individual values of root length?

To quantify uncertainty about the individual values of root length, we need to consider the standard deviation parameter, which represents the variation in root length values around the expected mean. In `brms`, the `fitted` function we used before only makes predictions about the mean of the response variable, and it quantifies uncertainty only about that mean. The`predict` function allows us to quantify uncertainty about the individual values of the response variable, collectively considering the mean and the standard deviation. It works much like the `fitted` function:

```r
y <- predict(m2, newdata=data.frame(conc = x))
pred <- cbind.data.frame(conc = x, y)
head(pred)
```

```
##        conc Estimate Est.Error     Q2.5    Q97.5
## 1 0.9400000 7.585479  1.133356 5.198950 9.701865
## 2 0.9683838 7.526634  1.166065 5.125505 9.812272
## 3 0.9967677 7.506209  1.185859 5.172469 9.809129
## 4 1.0251515 7.450557  1.161147 5.058929 9.727346
## 5 1.0535354 7.448916  1.154815 5.082123 9.682222
## 6 1.0819192 7.378764  1.155527 5.023748 9.561848
```

Notice that although the estimated values are similar to the estimates from `fitted`, the credible intervals are much wider here because they consider the uncertainty in the mean and the individual observations around the mean. We call this interval the

$$predictioninterval$$

. Let's go ahead and plot it:

```r
ggplot(d, aes(x = conc, y = rootl)) +
  geom_ribbon(data = pred,
              aes(x = conc, y = Estimate, ymin = Q2.5, ymax = Q97.5),
              fill = "grey83") +
  geom_smooth(data = fit,
              aes(x = conc, y = Estimate, ymin = Q2.5, ymax = Q97.5),
              stat = "identity",
              fill = "grey70", color = "black", alpha = 0.3, linewidth = 1/2) +
  geom_point(color="firebrick") +
  labs(,
    x = "Herbicide concentration (mM)",
    y = "Root Length (cm)"
  ) +
  scale_y_continuous(limits = c(0, 10)) +
  theme_classic()
```

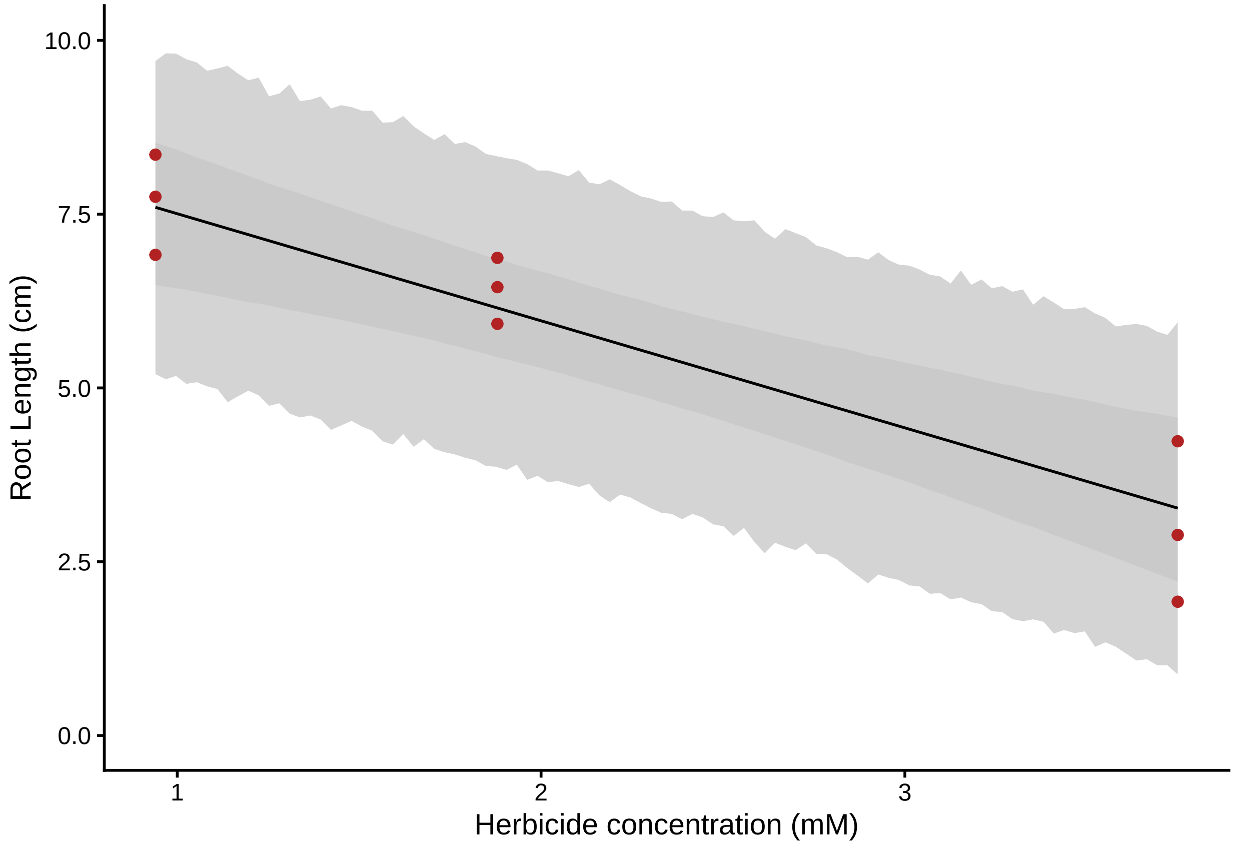


Figure 9.13: TODO: caption.

Here we've used the `geom_ribbon` function to add the prediction interval to the last version of our figure. This figure now shows the observed data (red points), the posterior mean prediction line (solid black line), uncertainty in the predicted mean (95% credible interval in dark gray shading), and uncertainty in the individual observations (95% prediction interval in light gray shading). Whereas the credible interval shows uncertainty about the expected mean, the 95% prediction interval in light gray shows where we would expect 95% of the root length values to occur at any particular value of herbicide concentration.

9.3 Linear models with categorical predictors

Linear models are flexible and can be modified to accomodate different types of data. So far we've looked at a simple model when we have a continuous response variable and a continuous predictor variable. In this section we look at what happens when the response is continous and the explanatory variable is categorical.

9.3.1 Binary explanatory variables

Let's revisit the dataset on color morphology of red-backed salamanders. These salamanders have two primary color morphs, striped and unstriped, that have been shown to vary in a number of other phenotypic traits. In this section let's examine if color morphology affects the size of adult salamanders, measured as the snout-vent-length (SVL). First we load the data:

```
tail <- read.csv("data/plci_tails.csv")
```

We begin by plotting the tail autotomy data and comparing between morphs, using the

```
ggplot(tail, aes(x = morph, y = length.cm)) +
  geom_jitter(width = 0.1, shape = 1, aes(color = morph)) +
  scale_color_manual(values = c("striped" = "red", "unstriped" = "slategray")) +
  labs(x = "Sex", y = "Snout-vent-length (cm)", color = "morph") +
  theme_classic() +
  theme(legend.position = "none")
```

We need to fit a model that allows us to estimate the mean SVL for each color morph and compare the mean between morphs. Here's our statistical model:

$$\begin{aligned}
&l_i \sim \text{Normal}(\mu_i, \sigma)\\
&\mu_i = \alpha_j\\
&\alpha_j \sim \text{Normal}(3.5, 0.5)\\
&\sigma \sim \text{Exponential}(3)
\end{aligned}$$

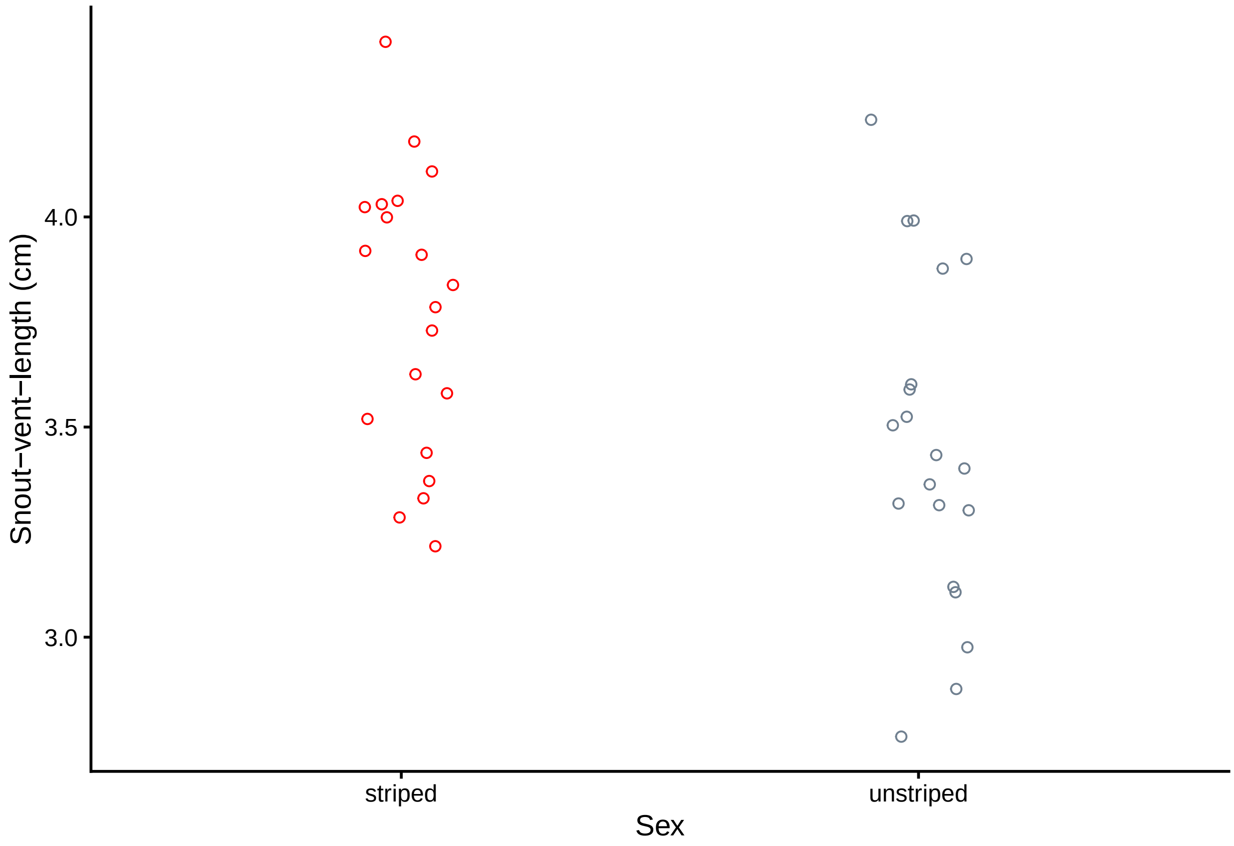


Figure 9.14: TODO: caption.

We assume the lengths (l) for each individual i follow a normal distribution with mean μ_i and standard deviation σ. The second line says the expected mean for each individual i is defined by the mean for its morph, α_j, where j represents an index for each color morph. Thus each individual i is either striped or unstriped, and we assume separate meeans for each morph. We assume a normal prior for the means of each sex with mean = 3.5 cm and standard deviation = 0.5 lb, reflecting the prior belief of a ~95% probability that the mean SVL will be between 2.5 and 4.5 for each morph. For the standard deviation parameter we assume an exponential prior. This is the first time we've seen an exponential distribution. The exponential distribution is handy for standard deviations because it is bounded at zero and has only positive values. The probabiity density is greatest at 0, and density declines with greater values, with the rate of decline specified by a single rate parameter (greater values indicate steeper rates of decline). The exponential function is considered a regularizing prior for standard deviations because it favors smaller values. Here we choose a an exponential prior with rate parameter 1 for the standard deviation of SVL.

A simple way to code categorical explanatory variables is by specifying a numerical index value for each category. This is simple a way of representing each category as an integer value. Let's define the striped morph as 1 and unstriped as 2:

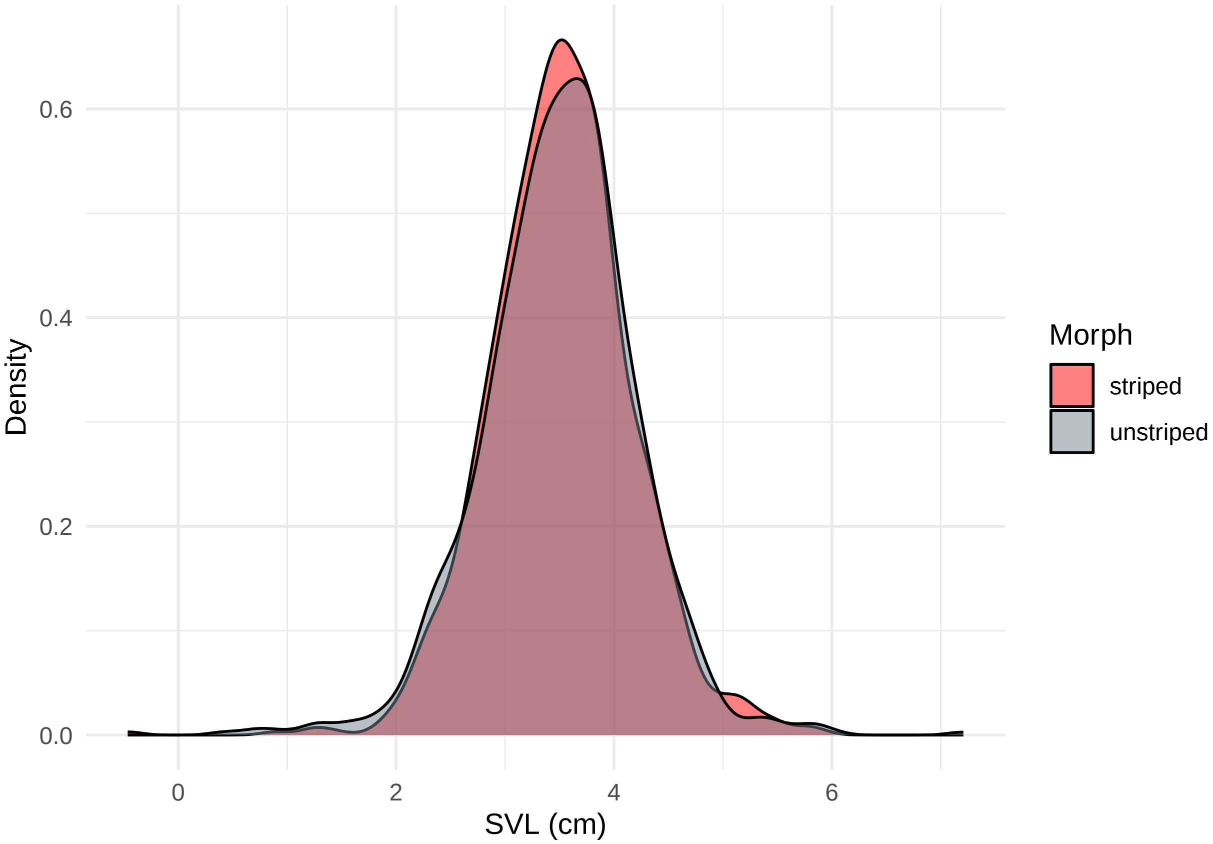
```r
tail$morph.i <- ifelse(tail$morph == "striped", 1, 2)
tail$morph.i <- factor(tail$morph.i)
```

One way to handle categorical predictor variables*index variable*. An index variable represents each category as an integer value. We also define the index variable, `morph.i` as a factor variable, which is required by `brms`.

We begin the analysis with a prior predictive check:

```r
set.seed(123)  # for reproducibility

#draw 1000 values of the mean striped length from the prior
striped_mu <- rnorm(1000, mean = 3.5, sd = 0.5)

#draw 1000 values of the mean festriped weight from the prior
unstriped_mu <- rnorm(1000, mean = 3.5, sd = 0.5)

#draw 1000 values of the standard deviation in weight from the prior
sample_sigma <- rexp(1000, 3)

#draw values of individual weights for stripeds and festripeds
prior_striped_l <- rnorm(1000, striped_mu, sample_sigma)
prior_unstriped_l <- rnorm(1000, unstriped_mu, sample_sigma)
```

```r
#combine to a dataframe
df <- data.frame(prior_striped_l = prior_striped_l,
                 prior_unstriped_l = prior_unstriped_l)

ggplot() +
  geom_density(aes(x = df$prior_striped_l, fill = "striped"), alpha = 0.5) +
  geom_density(aes(x = df$prior_unstriped_l, fill = "unstriped"), alpha = 0.5) +
  scale_fill_manual(values = c("striped" = "red", "unstriped" = "slategray")) +
  labs(x = "SVL (cm)", y = "Density", fill = "Morph") +
  theme_minimal()
```



We see the priors imply no difference in the distribution of individual SVLs, but certainly they allow for it. The only issue here is that the priors imply a few negative values for SVL, which of course doesn't make sense. However those observations are extremely rare, and we proceed to fit the model:

Let's go ahead look at how we fit the model with brms. We already have the data organized in the dataframe tail, so we can can get right to defining our model and priors.

All we want here is the estimation of a mean for each group, so we don't want to estimate a typical intercept. To suppress the intercept estimation in brms, we use the 0 + ... syntax. Adding 0 + morph.i syntax tells brms to compute a separate intercept for each index value of the morph.i factor variable. As such, when we define the priors for the means, we do so by specifying class = b (rather than Intercept).

```r
#specify model formula
m2.formula <- bf(length.cm ~ 0 + morph.i,
                family = gaussian)

#specify priors
m2.prior <- c(prior(normal(3.5, 0.5), class = b),
              prior(exponential(3), class=sigma))

#compute the posterior
m2 <- brm(data = tail,
         formula = m2.formula,
         prior = m2.prior,
         refresh = 0,
         seed=123)
```

```
## Running /Library/Frameworks/R.framework/Resources/bin/R CMD SHLIB foo.c
## using C compiler: ‘Apple clang version 17.0.0 (clang-1700.0.13.3)’
## using SDK: ‘MacOSX15.4.sdk’
## clang -arch arm64 -std=gnu2x -I"/Library/Frameworks/R.framework/Resources/include" -
## In file included from <built-in>:1:
## In file included from /Library/Frameworks/R.framework/Versions/4.5-arm64/Resources/l
## In file included from /Library/Frameworks/R.framework/Versions/4.5-arm64/Resources/l
## In file included from /Library/Frameworks/R.framework/Versions/4.5-arm64/Resources/l
## /Library/Frameworks/R.framework/Versions/4.5-arm64/Resources/library/RcppEigen/inclu
##   679 | #include <cmath>
##       |          ^~~~~~~
## 1 error generated.
## make: *** [foo.o] Error 1
```

```r
plot(m2)
```

We see the plot function returns posterior distributions for three parameters, including the mean for each morph and the standard deviation. The means for each morph are labeled with the appropriate index level, specifically morph.i1

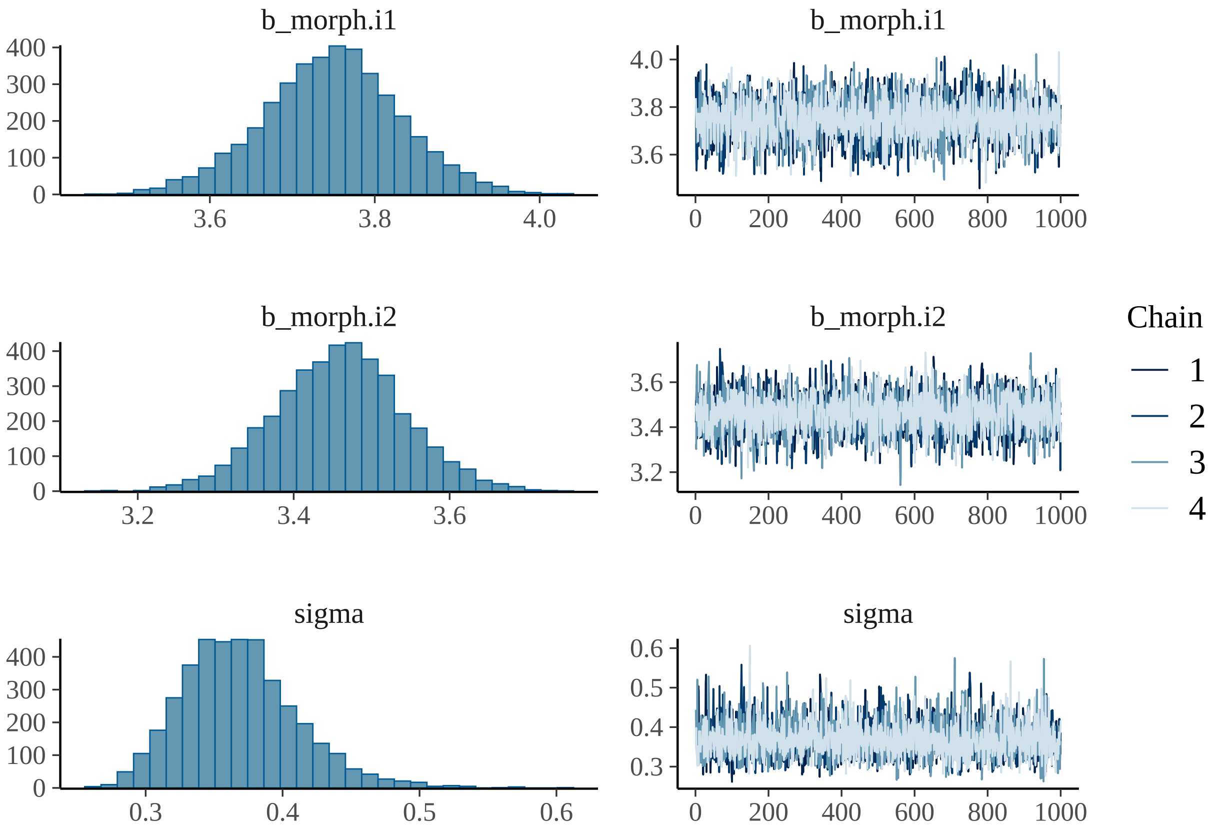


Figure 9.15: TODO: caption.

for striped and `morph.i2` for unstriped. The traceplots look pretty good, so we turn our attention to the model summary:

```
print(m2)
```

```
##  Family: gaussian
##   Links: mu = identity
## Formula: length.cm ~ 0 + morph.i
##    Data: tail (Number of observations: 40)
##   Draws: 4 chains, each with iter = 2000; warmup = 1000; thin = 1;
##          total post-warmup draws = 4000
##
## Regression Coefficients:
##          Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## morph.i1     3.75      0.08     3.58     3.91 1.00     3509     2932
## morph.i2     3.46      0.08     3.29     3.62 1.00     3670     2422
##
## Further Distributional Parameters:
##       Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## sigma     0.37      0.04     0.30     0.47 1.00     3474     2613
##
## Draws were sampled using sampling(NUTS). For each parameter, Bulk_ESS
## and Tail_ESS are effective sample size measures, and Rhat is the potential
## scale reduction factor on split chains (at convergence, Rhat = 1).
```

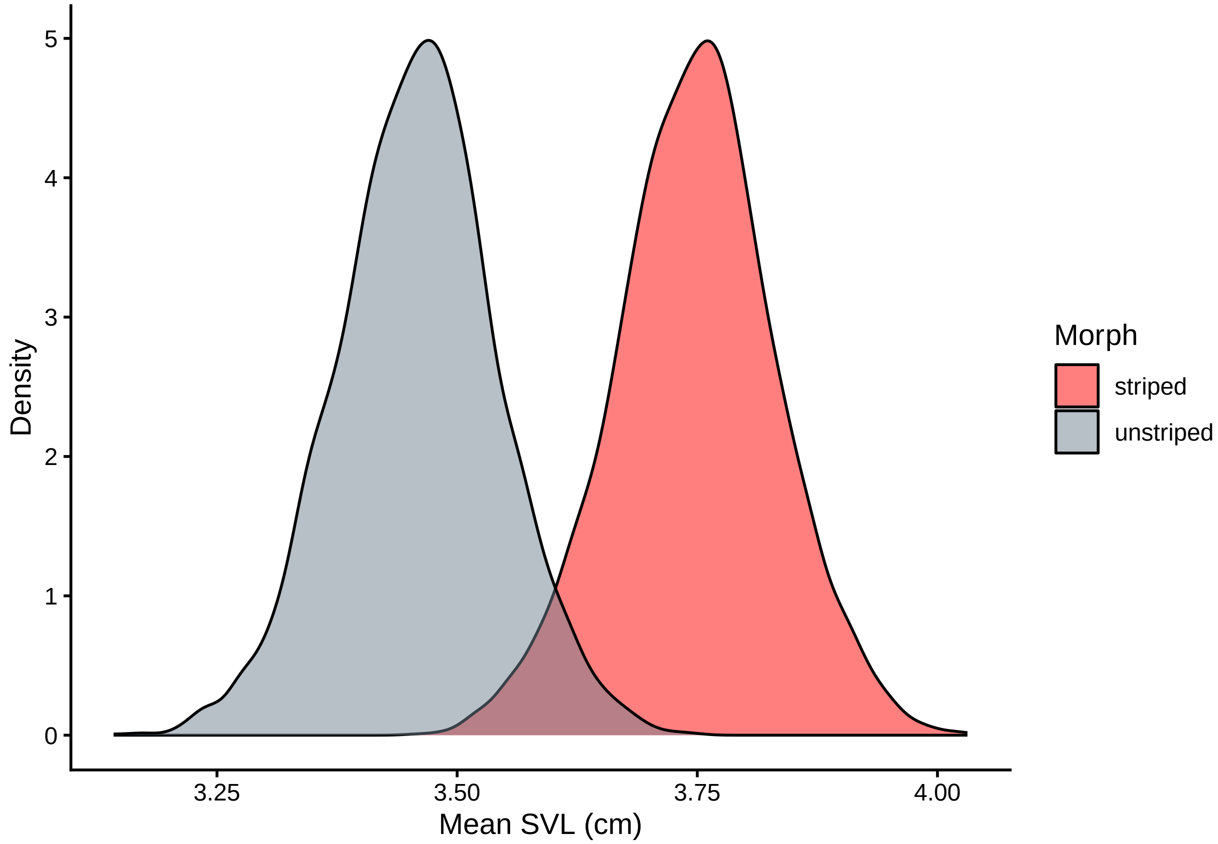
It looks like the mean SVL of the posterior for striped morph is 3.75 g, with a 95% credible interval from 3.58 to 3.91 g. In contrast, the mean of the posterior for unstriped morph is 3.46 cm, with a 95% credible interval of 3.29 to 3.62.

Is there a difference in the mean SVL between color morphs? The posterior distributions overlap somethwat, we we can see by the 95% credible intervals. But we don't need to visually look for overlap of posterior distributions. We can use the samples of the mean SVL for each morph to derive values that allow us to make the dcomparison between morph explicit. Let's first extract the samples and plot the posterior distributions for hte mean of each morph.

```
#extract the posterior samples
m2.post <- as_draws_df(m2)
head(m2.post)
```

```
## # A draws_df: 6 iterations, 1 chains, and 5 variables
##   b_morph.i1 b_morph.i2 sigma lprior lp__
## 1        3.9        3.5  0.37  -0.83  -19
## 2        3.8        3.5  0.37  -0.69  -18
## 3        3.8        3.6  0.40  -0.69  -18
## 4        3.8        3.3  0.35  -0.56  -18
## 5        3.7        3.6  0.35  -0.50  -17
## 6        3.7        3.5  0.42  -0.69  -18
## # ... hidden reserved variables {'.chain', '.iteration', '.draw'}
```

```
#plot
ggplot() +
  geom_density(aes(x = m2.post$b_morph.i1, fill = "striped"), alpha = 0.5) +
  geom_density(aes(x = m2.post$b_morph.i2, fill = "unstriped"), alpha = 0.5) +
  scale_fill_manual(values = c("striped" = "red", "unstriped" = "slategray")) +
  labs(x = "Mean SVL (cm)", y = "Density", fill = "Morph") +
  theme_classic()
```



We can see in the plot what we thought was apparent in the credibel intervals: the bulk of the posterior distribution for mean SVL for striped is greater than unstriped, but there is some overlap. If we are really interested in the difference between the mean SVL of each morph, we should quantify that difference directly from the samples and examine the posterior distribution of the difference in means. This is as simple is adding a new column to `m2.post` as the difference in means between the morphs, then we can summarize that posterior distribution. This is called a **contrast**.

```
#calculate the contrast as mean striped - mean unstriped
m2.post$morph.delta <- m2.post$b_morph.i1 - m2.post$b_morph.i2

#plot
ggplot(m2.post, aes(x = morph.delta)) +
  geom_density(fill = "slategray", alpha = 0.5) +
  labs(x = "Difference in mean SVL (cm; Stirped - Unstriped)", y = "Density") +
  theme_classic()
```

```
#mean and credible interval
mean(m2.post$morph.delta)
```

```
## [1] 0.2888998
```

```
quantile(m2.post$morph.delta, probs=c(0.025, 0.975))
```

```
##       2.5%      97.5%
## 0.06143877 0.51822613
```

We mean of the derived posterior difference in means between morphs is 0.29, with a 95% credible interval of 0.06 to 0.52 cm. Thus, There's a 95% probability that the mean SVL for striped morph is at least 0.06 cm greater than the unstriped morph. We could also compute the probability of the mean for each morph being greater than the other:

```
#probability of mean striped > mean unstriped
mean(m2.post$morph.delta>0)
```

```
## [1] 0.99125
```

```
#probability of mean untriped > mean striped
mean(m2.post$morph.delta<0)
```

```
## [1] 0.00875
```

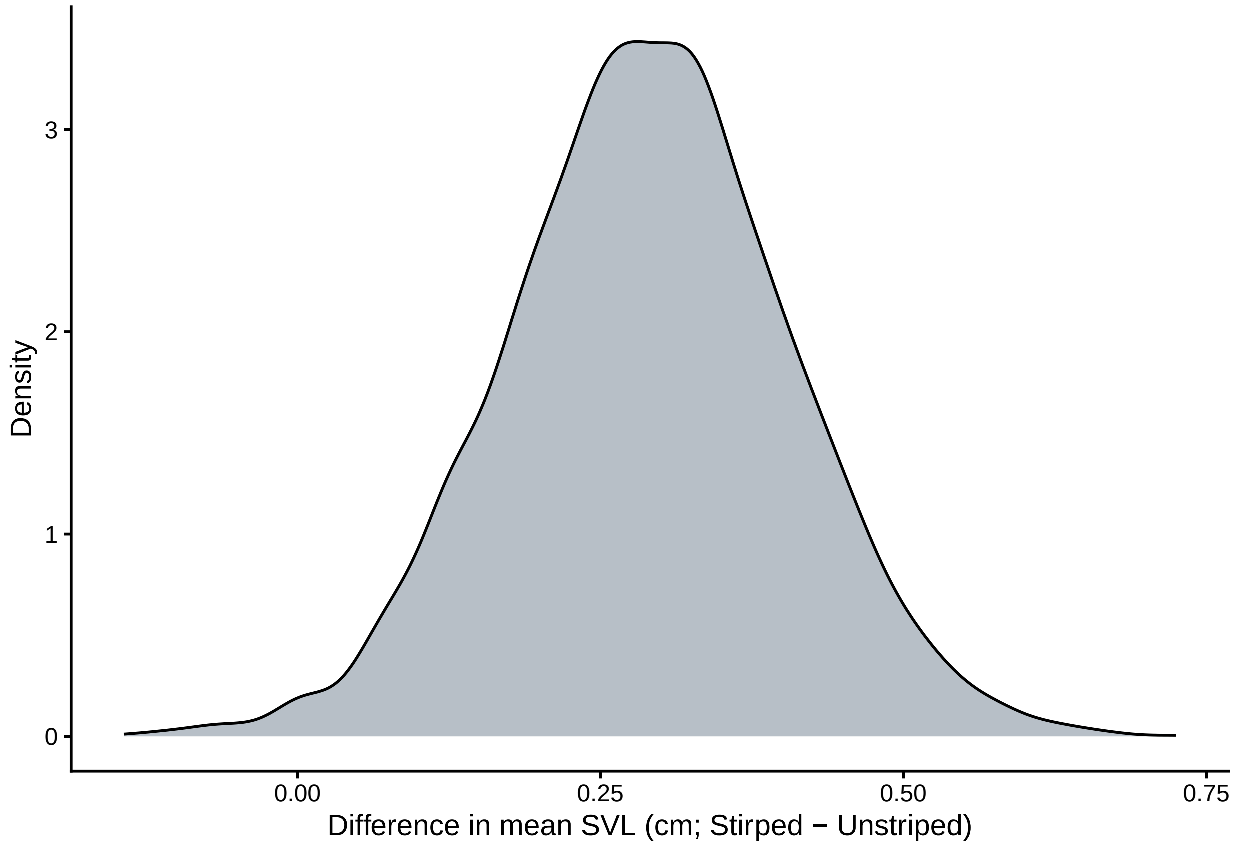
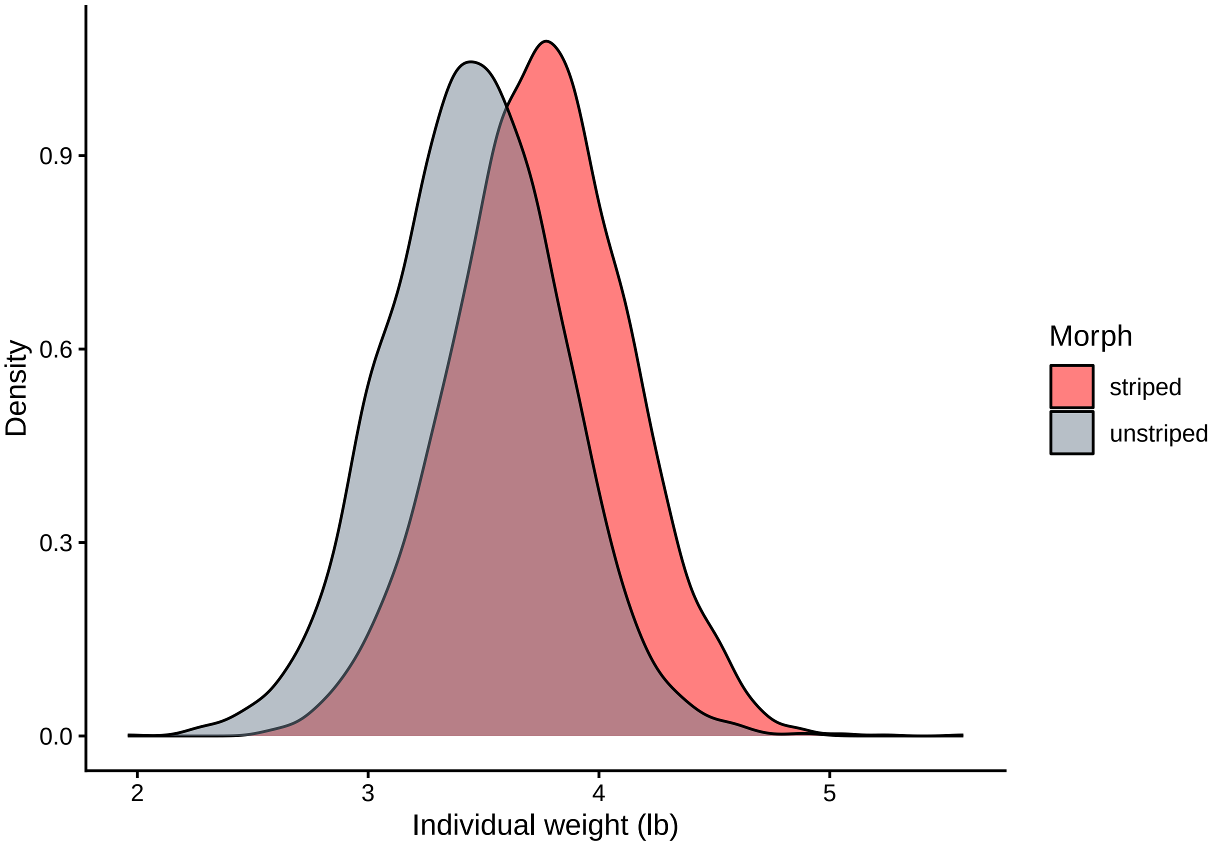


Figure 9.16: TODO: caption.

We find a 99% chance the mean SVL for striped is greater than unstriped.

In addition to examining the posterior distribution for the mean SVL, we can also examine the posterior distribution for the individual SVL values.

```r
set.seed(123)

# Simulate one weight per posterior sample
post_s_w <- rnorm(nrow(m2.post), mean = m2.post$b_morph.i1, sd = m2.post$sigma)
post_u_w <- rnorm(nrow(m2.post), mean = m2.post$b_morph.i2, sd = m2.post$sigma)

# Create data frame for plotting
df_post <- data.frame(post_s_w = post_s_w,
                      post_u_w = post_u_w)

ggplot() +
  geom_density(aes(x = df_post$post_s_w, fill = "striped"), alpha = 0.5) +
  geom_density(aes(x = df_post$post_u_w, fill = "unstriped"), alpha = 0.5) +
  scale_fill_manual(values = c("striped" = "red", "unstriped" = "slategray")) +
  labs(x = "Individual weight (lb)", y = "Density", fill = "Morph") +
  theme_classic()
```



Here the `rnorm` function is generating a random SVL for each color morph drawn from a distribution specified by the sampled mean and standard deviation from the posterior distributions for striped and unstriped morphs. Plotting the density of those 10,000 SVLs for each morph allows us to visualize the posterior distributions of individual SVL by morph. We can clearly see that the center of the distributions is greater for the striped than unstriped morph (as we saw previously), but that individual weights overlap quite a bit between color morphs.

We can also quantify a contrast for the individual SVLs. This gives us the posterior distribution of differences in *individual* SVLs between color morphs:

```r
#posterior distribution for difference in individual weights (contrast)
df_post$l_contrast <- df_post$post_s_w - df_post$post_u_w

#plot
ggplot(df_post, aes(x = l_contrast)) +
  geom_density(fill = "slategray", alpha = 0.5) +
  labs(x = "Difference in individual SVL (cm; Striped - Unstriped)", y = "Density") +
  theme_classic()
```

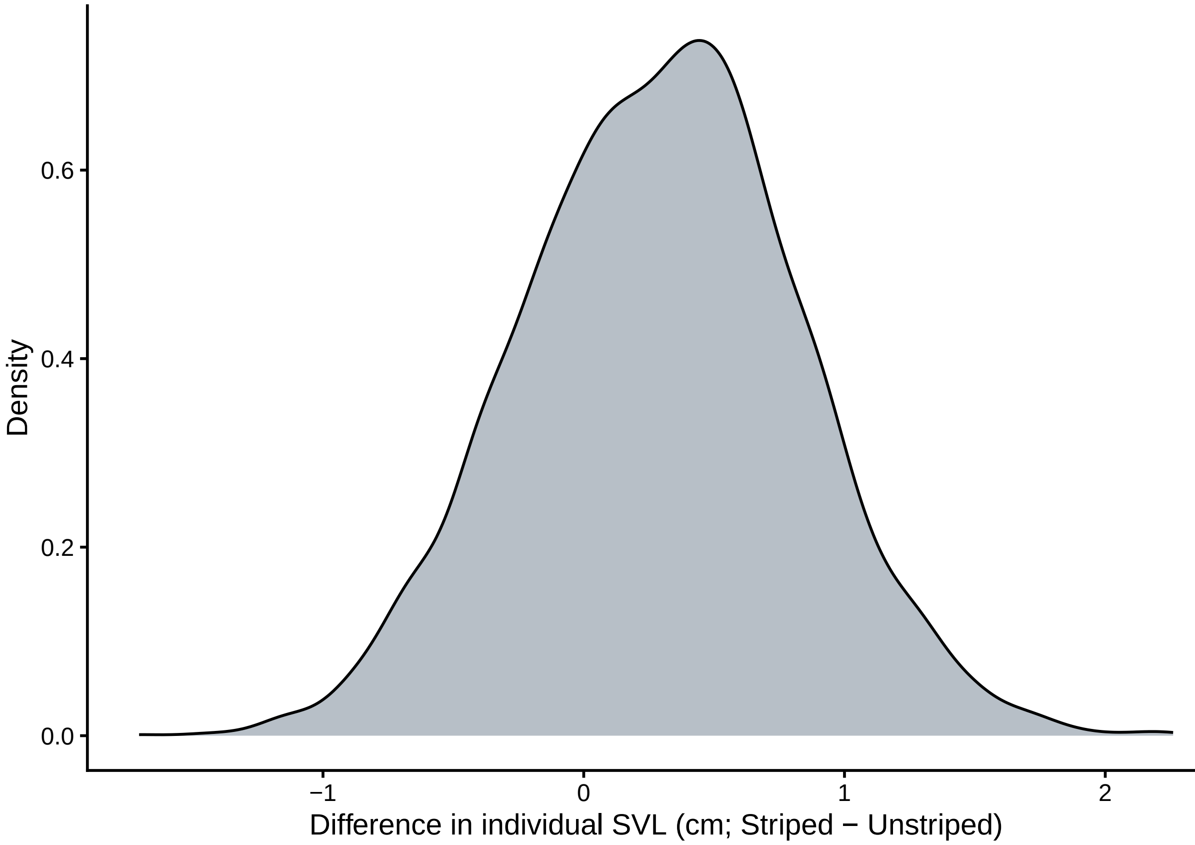


Figure 9.17: TODO: caption.

```
#proportion above zero
sum(df_post$l_contrast > 0)/nrow(df_post)
```

```
## [1] 0.70675
```

```
#proportion below zero
sum(df_post$l_contrast < 0)/nrow(df_post)
```

```
## [1] 0.29325
```

This contrast indicate that if we randomly select one striped morph and one unstriped morph from their respective SVL distributions, we can expect the striped morph to have a longer SVL than the unstriped morph 71% of the time, whereas we expect the unstriped morph to have a greater SVL 29% of the time.

9.3.2 Categorical predictors with more than two categories

Planned

Chapter 10

Graphical Causal Models

I've made the case that science starts with a clear research question, and we've explored some characteristics of what defines a good research question. I've also suggested that many (maybe most) scientific research studies have a primary goal of testing a causal hypothesis. In this chapter, we will take a closer look at causal questions. Designing scientific studies to address causal questions can be extremely challenging. Overcoming those challenges requires a clearly stated scientific model of the causal effects being hypothesized. Describing your scientific model and stating your causal assumptions with a graph is the focus of this chapter, and a theme that we will come back to as we explore the particulars of research design and statistics in the remainder of the book.

10.1 Directed acyclic graphs (DAGs)

There are two general approaches to **causal inference**: the potential outcomes framework (Neyman 1923, Rubin 2005) and the graphical causal model framework (Pearl 2000). In this book, we will use graphical causal models to guide our approach to causal inference. I find that the graphical representation of causal models provides for a smoother entry to the ideas of causal inference for students in an introductory statistics course.

The graphical causal modeling framework uses **directed acyclic graphs (DAGs)** to visualize the causal assumptions of a hypothesis. Let's work through an example. Suppose you are interested in the causal effect of living near urban greenspace on mental health. In cities, greenspaces are simply areas where natural vegetation occurs, such as forest. This is a forward causal question with a clearly defined cause (greenspace) and effect (mental health). The hypothesis is that living near greenspace reduces the risk of mental health disorders, perhaps by alleviating anxiety, or encouraging physical activity. Causal effects of one variable on another can involve multiple mechanisms.

Let's assume that we can measure whether or not people live near greenspace. We define `greenspace` as a nominal variable where individuals live either "near greenspace" or "not near greenspace". In reality, proximity to greenspace can be measured on a continuum, but for now let's keep it simple and define proximity categorically. Let's also assume that `mental.health` is a binary, categorical variable. People either have a mental health disorder, or they don't.

Our causal hypothesis is represented as a DAG in Figure 10.1. This kind of graph has nodes that represent variables, and arrows between nodes that illustrate the flow of causation. Based on how we've talked about our research question to this point, we have a really simple DAG. The nodes are greenspace and mental health, and there's a directional arrow from greenspace to mental health. The direction of the arrow shows the direction of causality. This DAG implies that greenspace directly influences mental health. Because there are no other variables between these variables, we define the effect of greenspace on mental health as a **direct effect**.

```r
dag1 <- dagitty("dag {greenspace -> mental.health}")
coordinates(dag1) <- list(x = c(greenspace = 0, mental.health = 1), y = c(greenspace =
drawdag(dag1)
```

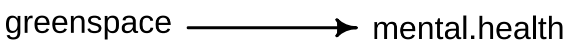


Figure 10.1: Initial DAG for the causal effect of greenspace on mental health.

Our hypothesis is that greenspace *reduces* the risk of mental health disorders, but note there's no indication of the nature of the causal effect in the DAG. DAGs simply illustrate assumed causal effects. DAGs do not assume anything about the form of the causal relationship between variables (e.g., positive vs. negative relationship, linear vs. non-linear relationship).

Drawing DAGs

In this chapter and throughout the book I will create figures of DAGs in R with the `dagitty` and `rethinking` packages. The relationship between variables are defined by the `dagitty` function, with the arrow `->` identifying the direction of causality between variables. You can use the `coordinates` function to customize the arrnangement of the variables in the plot along x- and y-axes.

I also highly recommend the dagitty website, where you can draw DAGs directly in your internet browser.

Life is rarely so simple as assumed in our initial DAG. Notably, the DAG assumes that greenspace is the only significant cause of mental health. The emphasis on significant is purposeful. In reality, there may be hundreds of causes of mental health. Most outcomes in biology are like this with multiple possible causes. However, DAGs focus on the most important causes. If there's a particular mutation on a gene that increases the risk of a mental health disorder by 0.001%, we probably don't need to include that in the DAG because the effect is trivial. Additionally, because we are focusing on mental health as an outcome, we don't necessarily need to include variables that are causally affected by mental health. Overall, some discretion about the variables to include in DAGs is warranted, because otherwise DAGs would be too complex to be useful (Huntington-Klein 2022). All models are simplifications.

Suppose that we collect some idea, and we find that 18% of individuals who don't live near greenspace have been diagnosed with a mental health disorder, whereas 11% of individuals who live near greenspace have a mental health disorder. The absolute risk [1] of a mental health disorder is 7% lower for individuals who live near greenspace. Is that difference caused by greenspace? Not necessarily! Why not? We need to consider the possibility that there is a common cause of greenspace and mental health. In other words, the people who live near greenspace differ in other ways from the people who don't live near greenspace, and these differences might contribute to mental health outcomes. One such possibility is socioeconomic status (SES). SES is very likely a cause of living near greenspace because wealthy people can afford to live near greenspace. SES might also be a cause of mental health because wealthy people have good access to healthcare. Figure **??** adds SES to the DAG.

```
dag2 <- dagitty("dag {greenspace -> mental.health;
                      SES -> greenspace;
                      SES -> mental.health}")
coordinates(dag2) <- list(x = c(greenspace = 0, SES = 1, mental.health = 2),
                          y = c(greenspace = 0, SES = -1, mental.health = 0))
drawdag(dag2)
```

[1] **Absolute risk** is the probability that an event occurs, here expressed as a percentage. Absolute risk can be contrasted with **relative risk**, which is the proportional risk of one outcome relative to another. For example, if the absolute risk of a mental health disorder is 18% for the "near greenspace" group and 11% for the "not near greenspace" group, then the relative risk of living near greenspace is 0.11/0.18 = 0.61. This means that it is 0.61 times less likely to be diagnosed with a mental health disorder when living near greenspace relative to not living near greenspace. Relative risk and absolute risk are often reported in the medical literature, but note that relative risk doesn't tell you anything about the absolute risk. One useful way of comparing risk between groups on the absolute risk scale is the **absolute risk reduction**, which just the difference in absolute risk between groups. That's the 7% that I referenced in the text.

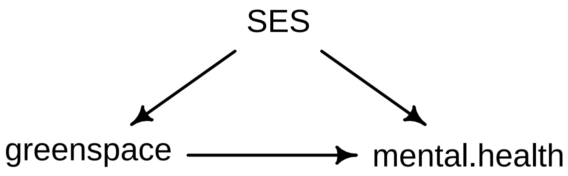


Figure 10.2: DAG with the confounding variable SES added.

The new DAG in Figure 10.2 clearly identifies SES as a common cause of both greenspace and mental health. This means the causal relationship between greenspace and mental health is confounded by SES. We call SES a **confounding variable** because it affects both the explanatory and response variables, ultimately confusing our interpretation of the relationship between greenspace and mental health. Remember that the risk of having a mental health disorder was 7% lower for people who live close to greenspace. It's possible that difference is due to a causal effect of greenspace on mental health, but it's also possible that some or all of the risk difference is due to the confounding effect of SES. Confounding variables can create associations between variables that are not causal.

Let me show you what I mean with the help of a data simulation. The risk probabilities of having a mental health disorder that I previously mentioned were generated by simulating data in R. Here's the code, followed by an explanation:

```r
set.seed(122)

## simulate socioeconomic status (ses)
ses <- rbinom(n = 1000, size = 1, prob = 0.5)

## simulate green space access (grn) based on ses
grn <- rbinom(n = 1000, size = 1, prob = ifelse(ses == 1, 0.8, 0.2))

## simulate mental health status (mnt) based on ses
mnt <- rbinom(n = 1000, size = 1, prob = ifelse(ses == 1, 0.1, 0.2))
```

I assumed that we have a dataset of $n = 1000$ people, and for each person, I first randomly determined if they are of high or low socioeconomic status. I did this using a function called `rbinom`. The arguments of the `rbinom` function include the number of observations (`n = 1000`), the number of trials (`size = 1`), and the probability of "success" (`prob = 0.5`). What this means is that for each of the 1000 individuals, I determine a single time whether they are of high or low SES, each with a probability of 0.5. This produces the variable `ses`, a vector of 0 and 1 values, where 0 represents "low ses" and 1 represents "high ses". Because I set the probability of success to 0.5, there should be about a

50/50 split of low and high ses individuals. It won't necessarily be exactly 50%, because the function is drawing values randomly. It's basically like flipping a coin 1000 times and counting heads and tails.

Once the ses variable was generated, I then generated the greenspace access variable, called `grn`. This was also generated with the `rbinom` function, where a 1 represents "near greenspace" and 0 represents "not near greenspace". I used the `ifelse` function to specify that the probability of being near greenspace was 0.8 for individuals of high ses and 0.2 for individuals of low ses. Finally, I generated a variable for mental health outcome, `mnt`, where 1 represents individuals who have a mental health disorder, and 0 represents individuals who don't have a mental health disorder. I assumed the probability of a mental health disorder was 0.1 for individuals of high ses and 0.2 for individuals of low ses. The only other part of the code chunk above is the `setseed` function, which allows you to simulate the exact same dataset that I simulated, as long as you use the same `setseed` number (122).

Now I want you to notice something important. Because we are simulating data, I had to make some assumptions about what the probability of a mental health disorder would be for individuals of high and low ses and individuals who are near or not near greenspace. In this simulation, I assumed that mental health risk is *only* affected by SES. There is no direct effect of greenspace access on mental health risk based in this simulation. Let's see what happens when we compare the probability of a mental health disorder between greenspace access based on this simulated dataset. We can do that with the `table` function:

```r
table(mnt, grn)
```

```
##    grn
## mnt   0   1
##   0 421 436
##   1  90  53
```

Here we see a **contingency table** showing the number of individuals in each category of mental health status and greenspace. Greenspace, the explanatory variable, is located at the top of the table, where the column for `grn` = 0 represents not near greenspace, and the column for `grn` = 1 represents individuals who live near greenspace. The rows represent the categories of mental health status, where `mnt` = 0 are the individuals who don't have a mental health disorder, and `mnt` = 1 are the individuals who have a mental health disorder. You can determine the totals in each category manually, or you can wrap the table function in a function called `addmargins`:

```r
addmargins(table(mnt, grn))
```

```
##      grn
## mnt      0    1  Sum
##   0    421  436  857
##   1     90   53  143
##   Sum  511  489 1000
```

With the addmargins function, we can see there were 511 people who don't live near greenspace and 489 people who do live near greenspace. We can also see the totals for mental health disorders: 143 people out of the 1000 have a mental health disorder. With this information, we can now compute the probability of a mental health disorder for the two categories of greenspace. This is exactly what a researcher might be tempted to do to assess whether living near greenspace causally affect mental health risk:

```
p.mnt.g0 <- 90/511 #mental health risk when greenspace = 0
p.mnt.g0
```

```
## [1] 0.1761252
```

```
p.mnt.g1 <- 53/489 #mental health risk when greenspace = 1
p.mnt.g1
```

```
## [1] 0.1083845
```

So we see exactly what I told you earlier. Of the 1000 people in the dataset, 18% of people not near greenspace had a mental health disorder, and 11% of people near greenspace had a mental health disorder (I'm rounding to two decimal places). One might be tempted to infer that the causal effect of greenspace is the 7% difference between these probabilities. But it's not! I know it's not, **because I simulated the data under the assumption greenspace access has no effect on mental health risk**. The only effect on mental health risk was SES. So why is there a 7% difference in risk probability with respect to greenspace? That difference is driven entirely by the confounding effect of SES in this simulation. High SES increases the chance that an individual lives near greenspace, and high SES decreases the chance of a mental health disorder. We can see this in the data:

```
addmargins(table(grn, ses))
```

```
##      ses
## grn      0    1  Sum
##   0    393  118  511
##   1     81  408  489
##   Sum  474  526 1000
```

```
81/474 #probability of near greenspace when ses is low
```

```
## [1] 0.1708861
```

```
408/526 #probability of near greenspace when ses is high
```

```
## [1] 0.7756654
```

```
addmargins(table(mnt,ses))
```

```
##       ses
## mnt       0    1  Sum
##   0     376  481  857
##   1      98   45  143
##   Sum   474  526 1000
```

```
98/474 #mental health risk when ses is low
```

```
## [1] 0.2067511
```

```
45/526 #mental health risk when ses is high
```

```
## [1] 0.08555133
```

Here we see that the probability of living near greenspace was 17% for low SES and 78% for high SES, and the probability of a mental health disorder was 21% for low SES and 9% for high SES. In this dataset, the people of low SES with a greater risk of a mental health disorder also tend to not live near greenspace.

OK - Who cares? Well, if you want to design a study to investigate whether or not greenspace has a causal effect on mental health, you'd get the wrong answer if you simply compared the raw risk of mental health disorders between people who do and do not live near greenspace. The DAG makes this clear, identifying SES as a confounding variable. If you want to get the right answer, the DAG shows us that we need to take into consideration the socioeconomic status of people. If we don't take SES into consideration, then the relationship we observe between greenspace and mental health will include a mix of the real causal effect of greenspace on mental health *and* the noncausal association via SES. ***DAGs make our causal assumptions clear and inform how we should design a study and conduct analyses to isolate the causal effects of interest***.

Later in the book we will learn more about what it means to take a variable "into consideration" when estimating a causal effect of some other variable, but let me briefly show you with this example. Basically, we need to look at the relationship between greenspace and mental health risk *while holding SES constant*. In other words, we need to compare the mental health risk between greenspace levels within the different categories of SES. Let's do this and see what we find. First, let's combine our variables into a dataframe, and then subset the dataset into two datasets, one for people of low SES, and another for people of high SES:

```
d <- cbind.data.frame(ses, grn, mnt) #create a data frame with all 3 variables
lo.ses <- d[d$ses == 0,]
hi.ses <- d[d$ses == 1,]
```

Now, let's compare the mental health risk between greenspace levels within each dataset. First, the individuals of low SES:

```
addmargins(table(lo.ses$mnt, lo.ses$grn,
                 dnn=c("mnt", "grn"))) #dnn argument adds names to the table
```

```
##      grn
## mnt     0   1 Sum
##   0   310  66 376
##   1    83  15  98
##   Sum 393  81 474
```

```
83/393 #mental health risk for not near greenspace
```

```
## [1] 0.2111959
```

```
15/81 #mental health risk for near greenspace
```

```
## [1] 0.1851852
```

Here we see a less than 3% difference in mental health risk between the levels of greenspace access. We'll learn that the difference we observe here is not a real causal effect; rather it's due entirely to what we'll call **sampling error**. More on that later.

How about the individuals of high SES?

```
addmargins(table(hi.ses$mnt, hi.ses$grn,
                 dnn=c("mnt", "grn")))
```

```
##      grn
## mnt     0   1 Sum
##   0   111 370 481
##   1     7  38  45
##   Sum 118 408 526
```

```
7/118 #mental health risk for not near greenspace
```

```
## [1] 0.05932203
```

```
38/408 #mental health risk for near greenspace
```

```
## [1] 0.09313725
```

Again, only about a 3% difference in mental health risk between the levels of greenspace access, but this time in the opposite direction (greater magnitude of risk when near greenspace). In other words, when we hold SES constant and look for an effect of greenspace access, we find no consistent effect.

This is a nice example showing how the causal assumptions about a system can be communicated in a DAG, which then informs how one proceeds to analyze the data. We will also examine how a DAG can inform how the data should be collected in the first place. More on that in the next chapter. For now, let's formalize some aspects about the structure of DAGS.

10.2 Three causal structures in DAGs

There are three main types of causal structures that are identifiable in DAGs. Let's work through each. In each case, assume there is an explanatory variable X and a response variable Y, and the primary interest is in understanding the causal effect of X on Y.

10.2.1 The fork: confounders

Forks have the structure $X \leftarrow Z \rightarrow Y$, where Z is a confounding variable. You've already seen the fork: $greenspace \leftarrow SES \rightarrow mental.health$. Here SES is a common cause of greenspace and mental health, and I showed how a confounding variable can create a noncausal association between the explanatory and response variables. Because SES increases greenspace access and reduces the likelihood of mental health disorders, the risk of a mental health disorder will be lower for people who live close to greenspaces than those far from greenspaces. In other words, confounders can generate spurious relationships between the explanatory and response variable, meaning they have to be taken into account when trying to understand the causal effect of X on Y.

Confounders can also mask patterns in the data generated from real causal effects. Suppose a researcher is interested in the causal effect of sunscreen use on skin cancer risk: *sunscreen* → *cancer*. The researcher collects data on both variables from medical records and finds no relationship; the probability of skin cancer is the same regardless of the use of sunscreen. Surely sunscreen reduces the chance of skin cancer, right?

When building a DAG, ***it is essential to include any variable that causally affects at least two other variables in the DAG***. Can you think of any variables that would causally affect sunscreen use and skin cancer risk? Here's a likely one: sun exposure. Sun exposure is a known risk factor for skin cancer, and people vary in their sun exposure based on where they live, work, and their lifestyle. Sun exposure might also causally affect sunscreen use. Perhaps people who have high sun exposure are more likely to use sunscreen. Do you see the issue here? Let's look at the DAG in Figure 10.3:

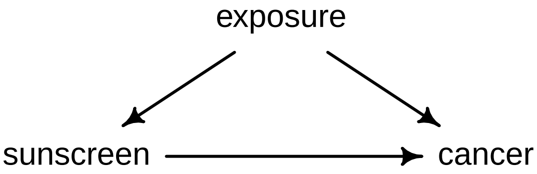


Figure 10.3: DAG for effects of sunscreen use and sun exposure on skin cancer risk.

If exposure increases the use of sunscreen and increases the risk of skin cancer, then that can generate an unexpected pattern of skin cancer being more common with sunscreen use! But let's say that sunscreen does have a direct causal effect of reducing skin cancer risk. If both forces are at play - the direct effect of sunscreen reducing cancer risk, and the confounding effect of sun exposure increasing sunscreen use and cancer risk - then you might not find any relationship at all between cancer risk and sunscreen use. In other words, sometimes confounders will **mask** a true causal effect, specifically if the pattern of the association between X and Y generated by the confounder is the opposite direction of the pattern of the association between X and Y generated by the direct effect.

Identifying confounders on back-door paths: Confounders can be identified by finding **back-door paths** from the explanatory variable to the response variable. A back-door path is any path in the DAG that starts with an arrow pointing into the explanatory variable and ends with an arrow pointing into the response variable. Forks create backdoor paths: *sunscreen* ← *exposure* → *cancer*.

Back-door paths can include more than two pathways. Consider a researcher trying to understand the causal effect of exercise on recovery time from a respiratory virus. Of course there are multiple potential common causes of exercise and viral recovery time. Figure 10.4 shows a DAG that includes the expected causal effect of exercise on recovery time, but also general health knowledge and vaccination status. See if you can find the back-door path between exercise and recovery.

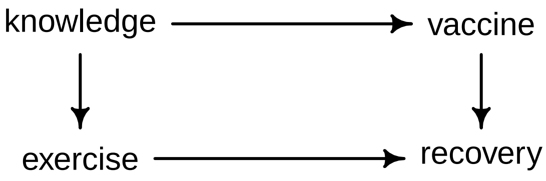


Figure 10.4: DAG for causal effect of exercise on recovery time from a respiratory virus.

The back-door path is *exercise* ← *knowledge* → *vaccine* → *recovery*. The confounder here is health knowledge. People with a lot of health knowledge might exercise more and be more likely to get vaccinated, and vaccination time can reduce recovery time. If that's the case, health knowledge confounds the relationship between exercise and recovery time. In this case, the back-door path includes more than one intervening node between the explanatory and response variable.

Dealing with confounders: When there's a back-door path identified in a DAG that will confound the relationship between the explanatory and response variable, that pathway must be blocked or controlled. The options for blocking back-door paths involving confounders include study design and statistical analysis and will be addressed in coming chapters.

10.2.2 The pipe: mediators

Let's continue examining the last DAG on viral recovery time to illustrate the next causal structure: the **pipe**. The pipe has a standard structure of $X \to Z \to Y$. The variable Z is called a **mediator**, because the causal effect of X on Y is mediated (at least in part) by Z. You've probably noticed this structure in some of the DAGs we've explored so far. For example, the path *knowledge* → *exercise* → *recovery* is a pipe. Here, the causal effect of health knowledge is mediated by exercise. Essentially, the causal effect of knowledge is transmitted to viral recovery via exercise level. The path *knowledge* → *vaccine* → *recovery* is also a pipe. In this case, vaccination is a mediator transmitting the causal effect of knowledge to recovery.

When the causal effect of a variable X on Y involves a mediator Z, we call the causal effect of X on Y an **indirect effect**. A causal effect of X on Y can involve multiple indirect effects. Thus, one can examine the indirect effect of knowledge on recovery via exercise, or the indirect effect of knowledge on recovery via vaccination. Note that there is no direct effect of knowledge on recovery in this model.

Causal effects of a variable X on Y can also involve direct and indirect effects. Consider an ecologist asking whether the amount of forest habitat in a landscape causally affects bird species diversity. The researcher hypotheses that forest can affect species diversity in two ways. First, forest area can directly affect species diversity because the amount of forest affects the amount of resources available for birds (e.g., food, nesting sites). Second, forest area can indirectly affect species diversity by affecting the degree of habitat fragmentation. When forest area declines, the remaining forest becomes spatially fragmented, and fragmentation can have a direct effect on diversity by limiting the immigration of new species. Figure 10.5 shows a DAG reflecting these ideas.

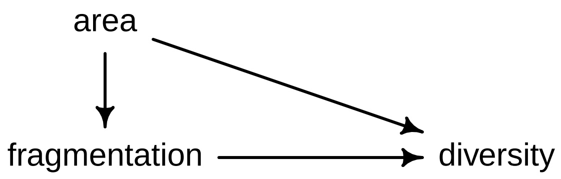


Figure 10.5: DAG for causal effect of forest area and fragmentation on bird diversity.

In this case, forest area has a direct effect on diversity and an indirect effect on diversity via the mediator fragmentation. The sum of a variables direct and indirect effects is called the **total effect**.

Dealing with mediators: When the causal effect of X on Y involves a pipe, generally there is no need to do anything about the mediator. A mediator is an important component of the causal effect of the explanatory variable of interest. If I want to know the total effect of forest area on bird species diversity, then I should make sure that area can affect bird diversity in all the ways I hypothesize, including the direct effect and the indirect effect via fragmentation. If I block the effect of fragmentation on diversity as part of the research design or analysis, then I am blocking part of the causal effect of forest area on diversity. This is called **post-treatment bias**. The "bias" part of this phrase means that you would get the wrong answer if you wanted to know the total effect of area on diversity but blocked the effect of fragmentation. If you want to know either the total (or indirect effect) of a treatment variable (another term for an explanatory variable), then you need to let the causal effect of that treatment variable be transmitted through its mediators.

In some circumstances it is OK to block a mediators effect. For example, suppose I was specifically interested in the direct effect of forest area on bird diversity, independent of its effect via fragmentation. In that case, I would want to design the study or analysis in a way to block the pipe involving fragmentation, leaving only the direct effect of area on diversity.

10.2.3 The inverted fork: colliders

Imagine a researcher asks whether people who have serious illnesses (e.g., heart disease, cancer, autoimmune disorders) are more likely to be infected with COVID-19 than people who don't have serious illnesses. The researcher conducts the study by examining patient records from a local hospital and finds a surprising result: there's a **negative association** between serious illnesses and COVID-19. In other words, people who have serious illnesses appear **less** likely to have COVID-19 than those without a serious illnesses. But there's one problem with this analysis: it's not correct! Let's explore why with a simulated dataset.

Assume that serious illness status and COVID-19 status are binary categorical variables, and additionally assume that having a serious illness **increases** the probability of having COVID-19. Let's also assume that having a serious illness or COVID-19 increases the probability of someone being admitted to the hospital. Figure **??** shows a DAG reflecting these assumptions.

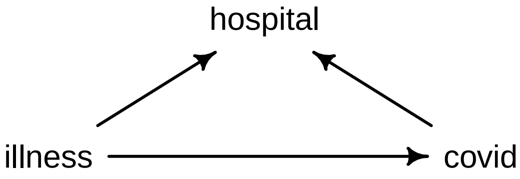


Figure 10.6: DAG for the causal effect of serious illness on COVID infection status.

And now here's the R code to generate some data based on the stated assumptions, using the `rbinom` function to simulate binary data just like in the example on greenspace and mental health outcomes.

```
set.seed(123)

## simulate illness status
ill <- rbinom(n = 1000, size = 1, prob = 0.2)

## simulate covid status based on illness status
cov <- rbinom(n = 1000, size = 1, prob = ifelse(ill == 1, 0.2, 0.1))

## simulate hospital status based on illness status and covid status
```

```r
hos <- rbinom(n = 1000, size = 1, prob = ifelse(ill == 1 & cov == 1, 0.29,
                                         ifelse(ill == 1 & cov == 0, 0.25,
                                         ifelse(ill == 0 & cov == 1, 0.05, 0.01))))
```

Here are the quantitative assumptions we've made in this simulated dataset of 1000 people:

1. The baseline probability of someone having a serious illness is 20%.
2. Serious illness increases the risk of COVID infection. The probability of a COVID infection is 20% for those with a serious illness and 10% for everyone else.
3. Serious illness and COVID both increase the chance of being admitted to the hospital. The baseline probability of someone without a serious illness or COVID being admitted to the hospital is assumed to be 1%. A serious illness increases the risk by 24%, and a COVID infection increases the risk by 4%. Thus, if someone has COVID but no serious illness (`ill == 0 & cov == 1`), their chance of being in the hospital is 5%. If someone has a serious illness but not COVID (`ill == 1 & cov == 0`), their chance of being in the hospital is 25%. If someone has a serious illness and COVID (`ill == 1 & cov == 1`), their chance of being in the hospital is 29%.

Now let's replicate the researcher's analysis by filtering the dataset to only include hospitalized patients, and then we will examine the relationship between serious illness and COVID status:

```r
d <- cbind.data.frame(ill, cov, hos) #create a data frame with all 3 variables
hos.patients <- d[d$hos == 1,] #extract the observations for the hospital patients

addmargins(table(hos.patients$ill, hos.patients$cov,
                 dnn = c("ill", "cov")))
```

```
##      cov
## ill    0  1 Sum
##   0    8  4  12
##   1   47 12  59
##   Sum 55 16  71
```

From the filtered dataset, we can see that we have a total of 71 hospital records. Of the 71 patients, 47 patients have a serious illness but not COVID, 4 patients have COVID but not a serious illness, and 12 patients have a serious illness and COVID. Does having a serious illness increase the risk of COVID?

If we simply compute the probability of COVID for people with and without a serious illness, here's what we find. The probability of COVID is $12/59 = 20\%$

for people with a serious illness and $4/12 = 33\%$ for people without a serious illness. That's right. Among hospitalized patients, people without a serious illness appear more likely to have COVID than those with a serious illness, the opposite of the true relationship in the general population based on our simulated model.

This counterintuitive result arises from **collider bias**. Take another look at the DAG and note that both serious illness and COVID status causally affect hospital admission: $Illness \rightarrow Hospital \leftarrow COVID$. This causal structure is called an inverted fork: $X \rightarrow Z \leftarrow Y$, where the variable Z is called a **collider** because it is causally affected by both X and Y. Hospital admission is a collider, where a patient is most likely to be admitted if they have a serious illness or COVID.

Dealing with colliders: Colliders only create problems when a researcher blocks or controls for the collider Z in the pathway $X \rightarrow Z \leftarrow Y$ as part of the research design or analysis. That's exactly what's happening with our example case of filtering the dataset to only individuals who were admitted to the hospital. When conditioning on hospital admissions in this way, a spurious relationship is generated between serious illness and COVID status.

Think about it this way. When we restrict the analysis to hospitalized patients, the two causes of hospitalization compete in a away to explain why patients were admitted. If a patient doesn't have COVID, they must have had a serious illness. If a patient doesn't have a seriuos illness, they must have had COVID. This creates a spurious negative relationship between serious illness and COVID status, even though the true relationship in the general population is just the opposite.

Collider bias is a type of **selection bias** in that it's driven by analyzing relationships between variables within a certain group. The world is full of these kind of examples. Why do ex-partners tend to either be attractive or intelligent, but not both? In this case the sample is restricted only to the people you dated in the past. If an ex wasn't particularly attractive, they were probably intelligent. If an ex wasn't intelligent, they were probably attarctive. Individuals who were neither attractive nor intelligent never made it into the pool of individuals you'd choose to go on a date with!

So beware of colliders. When you are interested in the causal relationship between X and Y, and X and Y both affect Z, generally avoid filtering the dataset by Z!

10.3 Closing backdoor paths

Drawing a DAG is a really useful way of being explicit about your scientific model, but it can also inform how you should go about designing your study and analyzing data. That's ultimately why we're covering graphical causal diagrams before we jump into statistics. DAGs represent your causal assumptions, and everything else - design, analysis, inference - follows from that. Science before statistics!

Once you have a DAG in hand, you can evaluate it to identify problematic pathways, basically any causal path that confounds the causal effect of interest. The causal pathways of interest start with a particular explanatory variable X and end with an outcome variable Y, and the arrows between X and Y should be forward facing. We will call these pathways **directed paths**, which are causal and of primary interest. This might involve a direct effect of X on Y, $X \rightarrow Y$, indirect effects of X on Y $X \rightarrow Z \rightarrow Y$, or both (the total effect). Problematic pathways are **back-door paths**, which are paths that connect X to Y but start with an arrow pointing into X. A fork structure with a confounder is the classic example. Back-door paths are noncausal and must be blocked. Finally, some paths between X and Y will start with a directed path out of X but end with a collider. If there's a collider on a pathway between X and Y, that pathway is noncausal but is already blocked. No further action is needed.

Let's take a look at an example. Figure 10.7 shows a DAG where the interest is in understanding the causal effect of screen time on obesity in pediatric patients (Williams et al. 2018). The DAG includes three other relevant variables, parent education, physical activity, and self-harm. This DAG represents a scientific hypothesis about the causal relationship between screen time and obesity, but it also includes other relevant pathways that are noncausal but could generate a spurious association between screen time and obesity. To understand how best to proceed with the design and analysis phase of the research, one would start by listing all the pathways between screen time and obesity, then identify the causal pathways of interest and the noncausal pathways. Given a list of all the pathways, causal and noncausal, one can then determine if any action needs to be taken to block the noncausal pathways from generating spurious associations.

Below I list every path connecting the explanatory variable (screen time) to the response variable (obesity). For each path, we identify whether it is directed, back-door, or blocked via a collider. Given this information, one can decide which variables should be controlled during the design or analysis phase, and which variables should be left alone.

- $Screen.time \rightarrow Physical.activity \rightarrow Obesity$: Here we have a pipe. This is a forward causal pathway involving an indirect effect of screen time on obesity via physical activity. The hypothesis here is that the more time a child spends on a screen, the less time they are being physically active, and the more likely they will be obese. This is a causal pathway of interest, so we wouldn't want to block it. It would be a mistake to block or condition on physical activity, because that's assumed to be an essential component of the causal mechanism linking screen to obesity. Blocking physical activity would be an example of post-treatment bias.

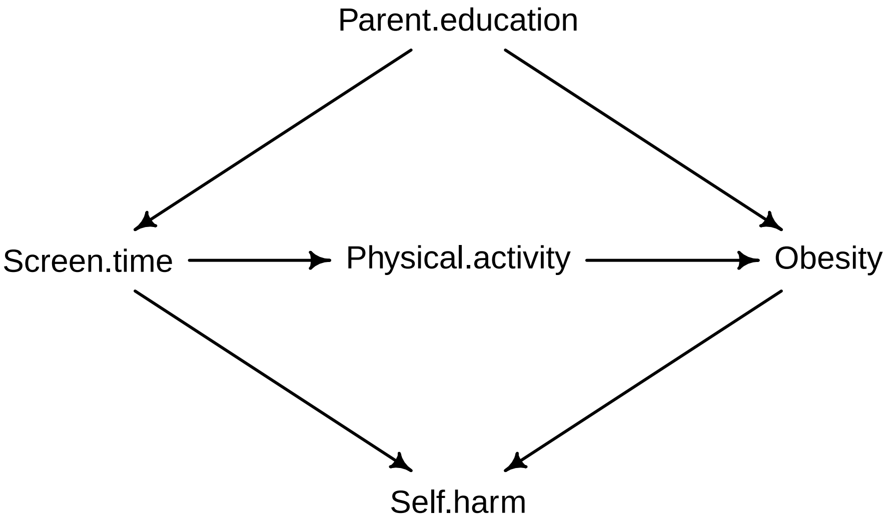


Figure 10.7: DAG for the causal effect of screen time on obesity in pediatric patients.

- *Screen.time* ← *Parent.education* → *Obesity*: This is a back-door path because it involves an arrow pointing into the explanatory variable. The causal structure is a fork, where parent education is a confounder. Thus, this is a non-causal path that will generate a spurious association between screen time and obesity. We need to block the effect of parent education on obesity during the design or analysis phase of the research.

- *Screen.time* → *Self.harm* ← *Obesity*: This is a path involving a collider. Note that both screen time and obesity are assumed to have direct effects on self-harm. Collider paths are blocked by default, so no action is necessary. It would be a mistake to design the research or conduct the analysis in a way that conditions on self-harm (for example, by only analyzing individuals who have a history of self-harm), because that would generate a noncausal association between screen time and obesity.

There you have it. With a DAG in hand, you can identify all the potential paths bewteen the explanatory and response variable, and determine for each whether not they need to be blocked. We will examine the methods of blocking available when we explore study designs and linear models later in the book.

10.4 Like all models, DAGs require assumptions

The great value of using DAGs is that they make causal assumptions crystal clear. Based on those assumptions and the causal structures observed in a DAG, one can design a data colleciton scheme or analyze the data in a way that increases the chance of obtaining an unbiased estimate of the causal effect of interest. But like all models, DAGs are simplifications of nature and have their own assumptions. Here are some important ones:

1. Causal effects in DAGs cannot be bidirectional. They must be **directed** from one variable to another.

2. DAGs cannot have cycles. In other words, you can't have a causal effect structure like this where you start with a causal effect of variable X and have a chain of causal effects that end back at X: $X \rightarrow Y \rightarrow Z \rightarrow X$. Cycles or loops are not allowed. DAGs are most beneficial to represent static causal models. If your scientific hypothesis is more dynamic, involving time and feedback loops, other tools for causal inference will often be necessary.

3. All variables with non-trivial effects should be included. This is most important when two variables have a shared cause, like Z here: $X \leftarrow Z \rightarrow Y$. The reason it's important to include common causes is because such confoudners need to be blocked by design or in the analysis to get an unbiased estimate of the relationship between X and Y. Unobserved or unmeasured confounders will lead to biased conclusions, although there are methods to help address unobserved confounders.

4. Building on the previous point, using a DAG to inform research design and analysis does not magically make any association you find a causal relationship. Leaving non-trivial variables out of the DAG, particularly confounders, will make causal conclusions from the DAG invalid. Domain knowledge becomes extremely important when designing a DAG to ensure that it adequately represents prior knowledge about the science of the research question.

So DAGs have assumptions and limits like any model, but they are a very useful method for an initial exploration of the tools of causal inference. We'll build on this foundation moving forward.

So far we have examined the importance of identifying a clear research question and communicating the causal assumptions related to the research question with a DAG. I've tried to make the case that defining clear research questions and developing an explicit scientific model as a DAG will help inform how one goes about designing research to collect and analyze data. In the next two chapters, we will begin to look at some basic elements of research design given a question and DAG in hand. We'll first explore our options for different types studies we can conduct, and then we'll explore the concept of sampling to collect data and some principles we can apply to maximize the quality of our dataset and the inferences we make from the data. Let's begin with study design.

10.5 Types of Study Designs

Consider the last research question from Chapter 4 on graphical causal models: Does screen time affect obesity in children? The hypothesis here is that screen time increases obesity due to reduced physical activity. Time is a zero sum game. Children who spend four hours on a screen each day are not spending those four hours being physically active, and increased risk of obesity may be a consequence of reduced physical activity. In the terms of a DAG in Figure 10.8, the expectation is that screen time indirectly increases obesity by reducing physical activity, where physical activity is a mediator.

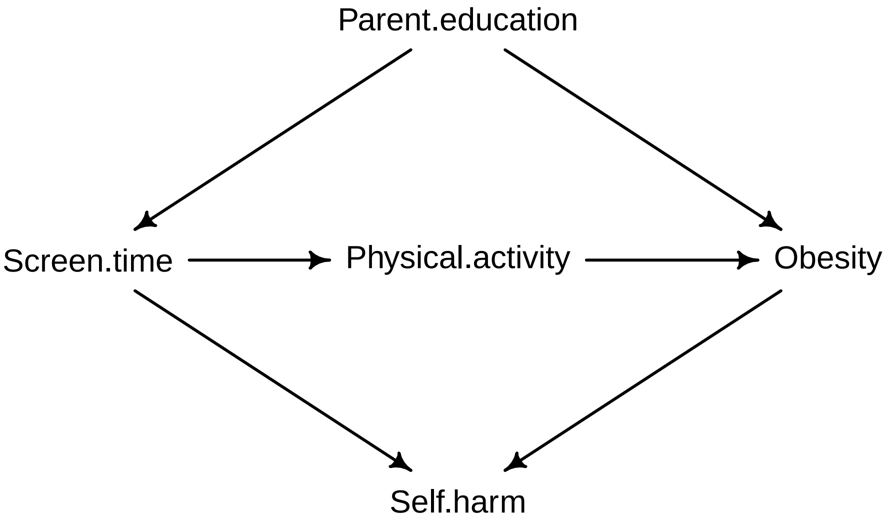


Figure 10.8: DAG for the causal effect of screen time on obesity in pediatric patients.

The DAG from Chapter 4 also includes two additional variables that are causally related the explanatory and response variable. Screen time and obesity are both expected to have direct effects on self-harm. Because self-harm is a collider in this DAG, and the path *Screen.time* → *Self.harm* ← *Obesity* is blocked by default. In other words, when we design the research, we should **not** attempt to do anything additionally in our research design to block the self-harm outcome.

Doing so would actually create a noncausal association between screen time and obesity.

The other relevant path in the DAG is $Screen.time \leftarrow Parent.education \rightarrow Obesity$. This is a back-door path with a confounding effect of parent education. Here the idea is that parents with a lot of education about health will know something about the risks of screen time and the risks of obesity, and they may take actions to minimize screen time and obesity in their children through mechanisms other than physical activity. For example, perhaps the parents who restrict screen time are also likely to restrict their children's diet to the most health foods. If that were the case, you might find a positive association between obesity and screen time. But that positive association wouldn't reflect the causal effect of screen time.

Given this DAG and the potential confounding effect of parental education, how do we proceed to design the study? There are two general types of study designs we can use: experimental and observational research.

10.6 Experimental studies

Experiments are essentially the gold standard for scientific research when the goal is causal explanation. There are two key elements of experiments. First, the researcher manipulates the explanatory variable. In other words, individuals in the experiment are assigned particular values of the explanatory variable, often referred to as the **treatment**. Second, the particular values of the explanatory varaible are **randomly assigned** to individuals in the experiment. This element is called **randomization**, and it is really the defining feature of an experiment.

Why is randomization so important? Let's consider this question in the context of the research question on screen time and obesity. The DAG shows that parent education is a confounding variable. Thus, the concern is that kids who have low levels of screen time have parents with high levels of education, and those parents with high education parent in other ways (besides screen time) to minimize the chance that their kids will be obese. If the researcher is interested in the effect of screen time on obesity, then the confounding effect of parent education must be controlled. And that's what randomization does.

Suppose the researcher enrolls 200 kids into the study. The researcher decides that she will assign three possible levels of screen time: 0, 5, or 10 hours per week. These different levels of screen time will be assigned to each participant randomly. For example, the researcher might have a list of each individual's name (or more likely, an identification code), and for each individual she could pick a piece of paper out of a hat indicating one of the three treatment levels. Or, she can use 21st-century technology and assign one of the three treatment levels in R:

```r
set.seed(123)
## create ID codes for each participant
id <- 1:200

## define the three treatment levels (0 = none, 5 = low, 10 = high)
trt.levels <- c("none", "low", "high")

## randomly assign each individual to a treatment
trt <- sample(trt.levels, 200, replace = TRUE)

## combine into a dataframe
d <- cbind.data.frame(id, trt)
head(d)
```

```
##   id  trt
## 1  1 high
## 2  2 high
## 3  3 high
## 4  4  low
## 5  5 high
## 6  6  low
```

The `sample` function was used to randomly assign each of the 200 participants to one of the three treatment levels [2]. Here's why randomization is the key feature of an experiment: If the treatment levels are randomly assigned to each individual, then there will be no relationship between the screen time of each participant and their parent's education level. A participant with who has highly educated parents will have an equal chance of being assigned to each of the three levels of screen time. In other words, experiments remove the influence of confounding variables on the explanatory variable. If the researcher used an experiment in this way, the DAG would be revised to eliminate the $Parent.education \rightarrow Screen.time$ effect as seen in Figure 10.9.

In this revised DAG, there is no longer a back-door path confounding the relationship between screen time and obesity. The research would randomly assign the levels of screen time, let enough time pass to observe the expected effect, and then record the value of the outcome variable, obesity (likely as the change in obesity from the start to the end of the study).

Random assignment of the explanatory varaible to individuals in an experiment not only helps control for confounding variables the researcher is aware of, but also unobserved confounders. Even with extensive domain knowledge, it's hard to think about every possible confounding variable and to include each one in the

[2] The `replace` = TRUE argument just means that the sampling is done "with replacement". That means if the first person is assigned the "high" treatment, the next person can also be assigned the "high" treatment.

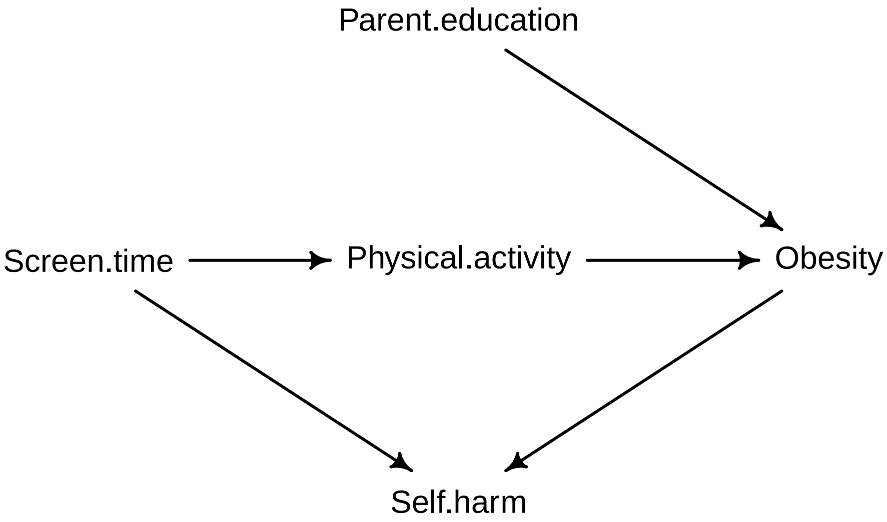


Figure 10.9: DAG for the causal effect of screen time on obesity in pediatric patients when using an experimental study design.

DAG. Randomization breaks the association between the explanatory varaible and unobserved confounders too.

Experiments are sometimes referred to as **randomized controlled trials (RCTs)**, emphasizing the essential component of randomization when attempting to infer a causal effect of an explanatory variable. The "controlled" part of the RCT name often refers to a **control group**, which is a group that does not receive the standard treatment. Control groups are often essential as a baseline for comparison. In the experiment described here on screen time and obesity, the "none" category of screen time is the control group. However, experiments don't always need a control group as traditionally defined. For example, suppose that instead of "none", "low", and "high", we just dropped the "none" category and assigned individuals to one of two treatment levels: low and high screen time. The researcher would still randomly assign these two levels of screen time to participants and then compare the change in obesity between the two groups. The "low" treatment functions as the baseline for comparison. A researcher might not include a traditional control group (complete absence of the treatment) if that's not a realistic value in the target population of interest, such as if the study is being conducted on a population where virtually no children have exactly zero screen time.

Although experiments are the goal standard of causal inference, correct causal inference isn't a guarantee. One problem researchers face is when participants in an experiment don't comply with their treatment instructions. This kind of issue can be common in experiments requiring behavioral compliance of people (e.g., psychology; Rohrer 2023). For example, imagine that some individuals who were assigned to the "none" treatment still give their kids some screen time. When compliance is a potential issue, researchers should collect data on both the treatment level randomly assigned *and* the treatment level actually received. Hopefully participatns are honest, and the person assigned to the "none" treatment level reports their actual screen time.

But even if people are honest and report their actual screen time, the noncompliance introduces significant complication for the analysis. Although the treatment levels were randomly assigned to participants, the noncompliance may not be random. What if noncompliance is affected by parental education? In this case, perhaps the most highly educated parents enforce stricter screen time limitations than the treatment they were assigned, where as parents with less education may be more likely to relax a strict limitation and allow their kids to have more screen time. The DAG in Figure 10.10 below is modified to reflect this possibility, now including a variable for the actual screen time experienced by the participants, and a direct causal effect of parent education on actual screen time. The researcher might have been aware of noncompliance and collected data on *actual* screen time, but it would be a mistake to simply analyze the relationship between obesity and actual screen time. Why? There's a backdoor path from actual screen time to obesity via the confounder, parent education. Even in an experimental context, correctly analyzing the causal effect of screen time in this example would require an analysis that controls for the effect of parental education on actual screen time.

So, experiments aren't a guarantee of safe causal inference, even when treatments are randomly assigned. Moreover, sometimes conducting experiments can be challenging, or even impossible. Imagine you're an economist studying the effect of the minimum wage on employment. It's impossible to randomly assign different minimum wages to different municipalities. If you're a biologist studying the effect of urbanization on biodiversity, you can't randomly assign areas that do and do not become urbanized. Sometimes experiments can't be done because they're unethical. If you're a psychologist studying the effect of parenting style and child personality, you can't randomly assign babies to different parenting styles. In cases like these, researchers have to rely on observational designs, which we turn to next.

10.7 Observational studies

When researchers collect data without randomly assigning levels of the explanatory variable to individuals, they are conducting an ***observational study***. The key element making a study observational is when the researcher has no role in determining how the data arose. The data arose naturally and are observed and recorded by the reseacher. Sometimes you have to take what you can get.

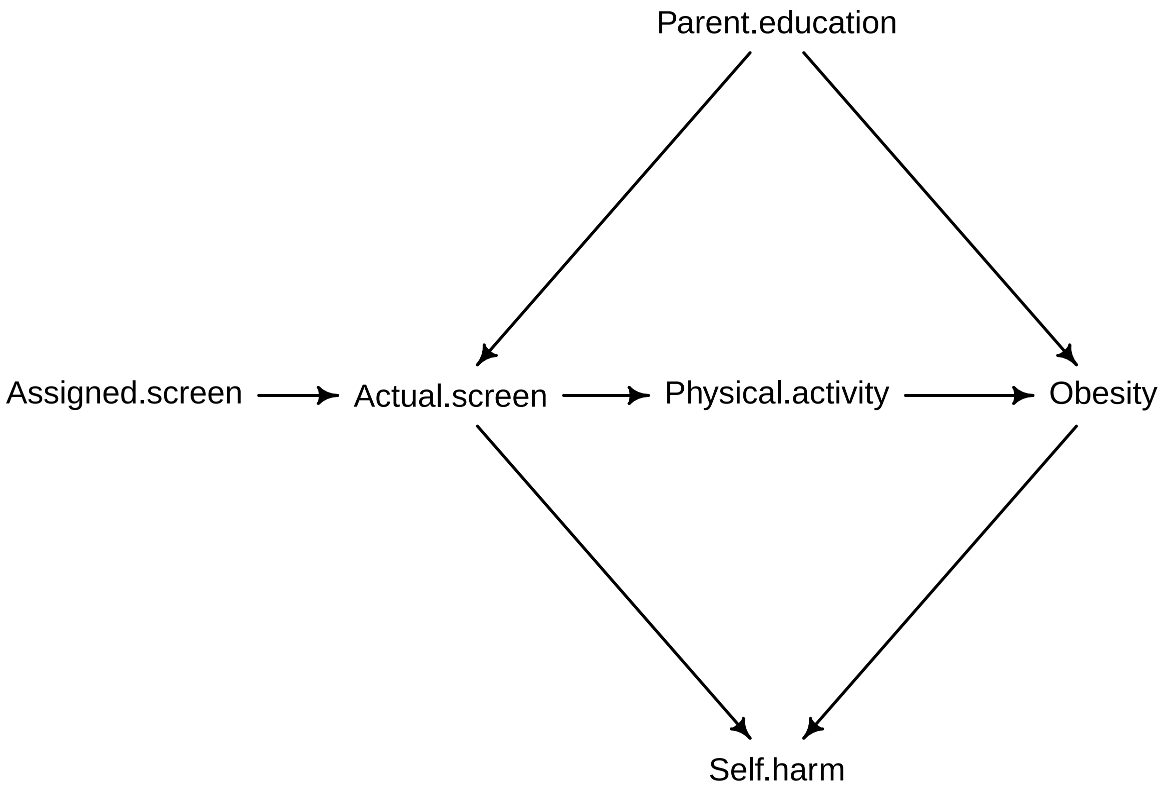


Figure 10.10: Revised DAG including actual screen time.

How would an observational design be conducted to address our research question on screen time and obesity. There's a couple ways such a study could proceed. One approach would be to enroll individuals into a study and track their screen time and obesity over time. This kind of design is called a **prospective study**, which is simply a study that is planned in advance. Another approach would be to analyze existing data on screen time and obesity. Perhaps a dataset was previously collected by these factors by another researcher, or by a medical doctor in their own practice. This kind of design is called a **retrospective study**, referring to the fact that the data being analyzed were generated in the past. Both of these designs have the common element of the data arising naturally on their own without the researcher randomly assigning levels of screen time.

In some ways observational studies are much easier to pull off, particularly if a dataset is already available. But observational designs can present significant challenges in interpretation. Because the levels of the explanatory variable are not randomly assigned, any back-door path involving a confounding variable is unblocked in an observational design. Sure, it might be easy to record screen time and obesity for a bunch of kids, but remember parental education? Kids that have knowledgeable parents might limit their screentime and enforce a diet that minimizes obesity. In that case, one would find a positive relationship between screen time and obesity, not because of a causal effect of screen time, but because of confounding effect of parental education. This is exactly what people mean when they say "correlation is not necessarily causation". In addition to generating noncausal correlations, it's also important to remember that confounders can also mask real relationships.

So how do we proceed to make causal inferences with an observational design? One needs to have a clear scientific model of their system that makes causal assumptions clear, such as with a DAG. With a well-defined DAG that identifies likely back-door paths and confounders, one can then block those paths during the analysis phase. We won't learn the methods to do that until later in the book, but at this point it's important to understand that the researcher must identify potential confounders *and* measure them in order to apply the methods we'll use to block back-door paths during analysis. That's why researchers should start by defining a DAG before the data are collected.

10.7.1 Pros and cons of prospective vs. retrospective studies (planned)

Chapter 11

Causal inference with linear models

I like ice cream. A lot. So I became alarmed when I saw the graph below, suggesting that the number of drownings in the United States is positively related to ice cream sales.

Why would the number of drownings be related to ice cream sales? Well, one hypothesis is that ice cream consumption causally increases risk of drowning. Perhaps eating a lot of ice cream compromises one's ability to swim. The DAG below represents this hypothesis.

The DAG here shows that the number of drownings (D) is indirectly affected by ice cream sales (IC) via the mediators ice cream consumption (C) and swimming performance (SP), which are unobserved. If this DAG was correct, we would expect that the number drownings is related to ice cream sales. We have monthly data available on those measurements as seen in the graph above, but we should build a linear model to more rigorously examine the relationship than by simply eye balling it. Here's our statistical model. Note we're mean-centering the monthly ice-cream sales:

$$
\begin{aligned}
&d_i \sim \text{Normal}(\mu_i, \sigma) \\
&\mu_i = \alpha + \beta(x_i - \bar{x}) \\
&\alpha \sim \text{Normal}(13, 2) \\
&\beta \sim \text{Normal}(0, 2) \\
&\sigma \sim \text{Uniform}(0, 5)
\end{aligned}
$$

For the sake of brevity, I'll skip the presentation of a prior predictive check and move right onto fitting the model. W

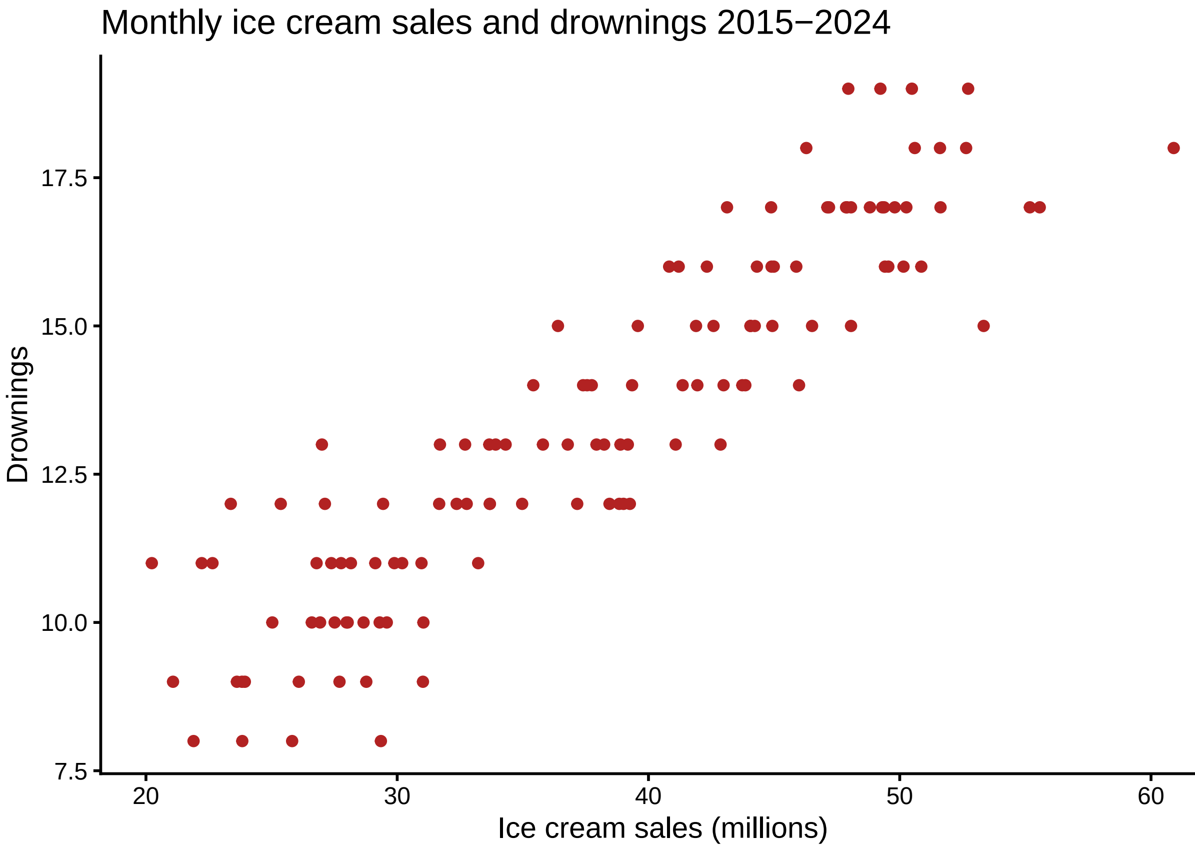


Figure 11.1: TODO: caption.
(#fig:c11c01,)

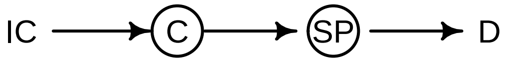


Figure 11.2: Initial DAG for the causal effect of greenspace on mental health.
(#fig:c11c02,)

```r
#mean center ice cream sales
d$sales.c <- d$ice_cream_sales_millions - mean(d$ice_cream_sales_millions)

library(brms)
#specify model formula
m1.formula <- bf(drownings ~ 1 + sales.c,
                family = gaussian)

#specify priors
m1.prior <- c(prior(normal(13, 2), class = Intercept),
              prior(normal(0, 0.25), class = b),
              prior(uniform(0, 5), class = sigma, lb=0, ub=5))

#compute the posterior
m1 <- brm(data = d,
         formula = m1.formula,
         prior = m1.prior,
         refresh = 0,
         seed=123)
```

```
## Running /Library/Frameworks/R.framework/Resources/bin/R CMD SHLIB foo.c
## using C compiler: ‘Apple clang version 17.0.0 (clang-1700.0.13.3)’
## using SDK: ‘MacOSX15.4.sdk’
## clang -arch arm64 -std=gnu2x -I"/Library/Frameworks/R.framework/Resources/include" -DNDEBUG
## In file included from <built-in>:1:
## In file included from /Library/Frameworks/R.framework/Versions/4.5-arm64/Resources/library/Sta
## In file included from /Library/Frameworks/R.framework/Versions/4.5-arm64/Resources/library/Rcp
## In file included from /Library/Frameworks/R.framework/Versions/4.5-arm64/Resources/library/Rcp
## /Library/Frameworks/R.framework/Versions/4.5-arm64/Resources/library/RcppEigen/include/Eigen/s
##   679 | #include <cmath>
##       |          ^~~~~~~
## 1 error generated.
## make: *** [foo.o] Error 1
```

```r
#values
x_vals <- seq(min(d$sales.c), max(d$sales.c), length.out=100)

#mean prediction
y.mu <- fitted(m1, newdata=data.frame(sales.c = x_vals))
fit <- cbind.data.frame(sales.c = x_vals, y.mu)
fit$ice_cream_sales_millions <- fit$sales.c + mean(d$ice_cream_sales_millions)

#individual prediction
y <- predict(m1, newdata=data.frame(sales.c = x_vals))
pred <- cbind.data.frame(sales.c = x_vals, y)
```

```
pred$ice_cream_sales_millions <- pred$sales.c + mean(d$ice_cream_sales_millions)

#now plot; geom_line adds the posterior mean line
ggplot(d, aes(x = ice_cream_sales_millions, y = drownings)) +
  geom_ribbon(data = pred, fill = "grey83",
              aes(x = ice_cream_sales_millions, y = Estimate, ymin = Q2.5, ymax = Q97.
    geom_smooth(data = fit, stat = "identity",
              fill = "slategray3", color = "black", alpha = 1, linewidth = 1,
              aes(x = ice_cream_sales_millions, y = Estimate, ymin = Q2.5, ymax = Q97.
  geom_point(shape = 1, size = 2, color = "firebrick") +
  labs(x = "Ice cream sales (millions)", y = "Drownings") +
  theme_classic()
```

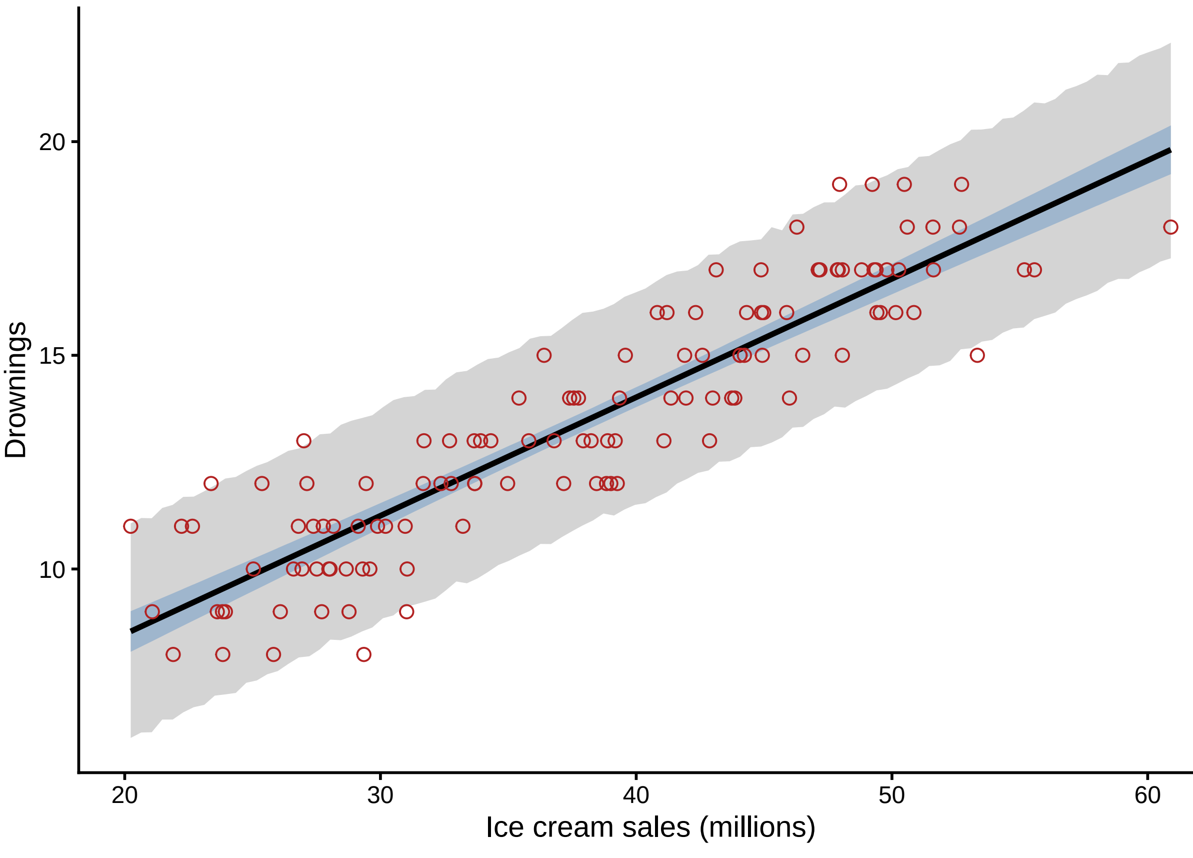


Figure 11.3: TODO: caption.

From the model we see the expected increase in drownings per every one million more ice cream sales is 0.28 with a 95% credible interval of 0.26 to 0.30. In other words, we confirmed what we saw with our eyes. The average number of drownings per month is positively related to ice cream sales. This means ice cream causes drownings, right?

Well, no. Good for you if you've remained skeptical to this point. We've confirmed here that there is indeed a positive association between drownings per month and ice cream sales. But that doesn't mean the association is causal. Just because we have a DAG doesn't make the association we find causal. The scientific hypothesis represented by our DAG might be wrong. And in this case, it's almost certainly wrong. Here's a DAG representing an alternative hypothesis:

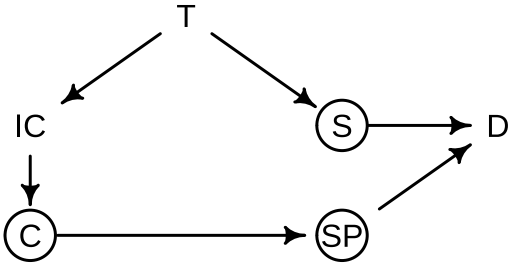


Figure 11.4: Initial DAG for the causal effect of greenspace on mental health. (#fig:c11c04,)

What does this DAG say? First, we see the same causal pathway that we saw before from ice cream sales to drowning via consumption and swimming performance. But now we've added another pathway by which drownings is linked to ice cream sales. The new causal pathway shows that air temperature (T) affects ice cream sales, which makes sense as people tend to eat more ice cream when it's warm. Air temperature also affects the number of swimmers (S), which is an unobserved variable. People are more likely to swim when it's warm! Numerically we expect more drownings when there's more swimmers, so there's a direct effect of swimmers on drownings.

Do you see the issue here? There are two pathways linking ice cream sales to drownings. One is a causal pathway via ice cream consumption and swimming performance, and the other is a backdoor path via temperature and number of swimmers. Temperature here is a confounder! If we want to test the causal effect of ice cream sales on drownings, we would need to adjust for temperature. Recall from Chapter 10 that we can identify adjustment sets necessary when testing the causal effect of one variable on another:

```r
#testing the total effect of ice cream sales on drownings
adjustmentSets(dag2, exposure="IC", outcome="D", effect = "total")
```

```
## { S }
## { T }
```

Here we see we can adjust for temperature or the number of swimmers, but as the number of swimmers is unobserved, we would have to adjust for temperature.

But what does it mean to adjust for a variable? When we do an experiment, we know that randomization breaks the kind of confounding observed in our second DAG. But in an observational design we can't do that. Instead, we approach the process of adjustment with our statistical models. This chapter shows you how to do that.

11.1 Linear models with multiple predictor variables

So we have two DAGs that link ice cream sales to drownings. In the first DAG, there's only one causal pathway between ice cream sales and drownings. Based on that DAG, any relationship we observe between drownings and ice cream sales must be due to that causal pathway. The second DAG adds a backdoor path and suggests that at least part of the association between drownings and ice cream sales may be due to confounding with temperature. If the second DAG is correct, we should still find a relationship bewteen drownings and ice cream sales when we adjust for temperature. Fortunately we can adjust for temperature by including it in a linear model along with ice cream sales to predict drownings.

11.1.1 Generic linear model with two predictors

Remember that the simple linear model has two parameters: an intercept and a slope representing the association between the response and explanatory variables of interest. But we can expand the linear model to include *multiple* explanatory variables. A linear model with multiple predictors is commonly called a **multiple regression** model. For example, here's a multiple regression model with two explanatory variables:

$\mu_i = \alpha + \beta_1 x1_i + \beta_2 x2_i$

Let's walk through what each of these terms represents:

- μ_i: This is the expected mean value of the response variable for any individual with a values of $x1_i$ and $x2_i$.
- α: This is still the intercept term. In our linear model it represents the expected value of the response variable when the values of both $x1$ and $x2$ are 0.

- β_1: This is the slope for the relationship of Y to $x1_i$ *when holding* $x2_i$ (and any other predictor we might add to the model) constant.
- β_2: This is the slope for the relationship of Y to $x2_i$ *when holding* $x1_i$ (and any other predictor we might add to the model) constant.
- $x1_i$: This is the observed value of variable $x1$ for observation i
- $x2_i$: This is the observed value of variable $x2$ for observation i

The model can be expanded to incorporate any number of additional predictor variables, each with a slope β while holding other predictors constant.

11.1.2 Multiple regression model for the ice cream and drownings example

Let's apply this model to our question about the causal effect of ice cream sales on drownings. We want to know if drownings is related to ice cream sales when controlling for the confounder, air temperature. Here's our statistical model:

$$\begin{aligned}
&d_i \sim \text{Normal}(\mu_i, \sigma)\\
&\mu_i = \alpha + \beta_c(c_i - \bar{c}) + \beta_t(t_i - \bar{t})\\
&\alpha \sim \text{Normal}(13, 2)\\
&\beta_c \sim \text{Normal}(0, 0.25)\\
&\beta_t \sim \text{Normal}(0, 0.25)\\
&\sigma \sim \text{Uniform}(0, 5)
\end{aligned}$$

Let's walk through each line of the model to ensure we know what everything means!

- $\hat{d_i}$: This is the expected mean number of drownings for each observation i with values c_i ice cream sales and t_i for temperature. In other words, given specific values of ice cream sales and temperature, this is the predicted mean number of drownings. We could look at the expected number of drownings for any combination of the values of ice cream sales and temperature.
- α: This is the intercept term. Note that we are mean-centering ice cream sales and temperature, so the intercept represents the expected number of drownings when ice cream sales and air temperature are both at their means.
- β_c: This is the slope for the relationship of drownings to ice cream sales *when temperature is held constant.*
- β_A: This is the slope for the relationship of drownings to temperature *when ice cream sales are held constant.*

- c_i: This is the observed value of ice cream sales for each observation i
- t_i: This is the observed value of temperature for each observation i

The only thing that's really new about this model is the interpretation of the slopes. No longer are the slopes simply the expected change in the response variable for each one unit change in the explanatory variable. Rather, the slopes represent the expected change in the response variable per unit change in the explanatory variable *while holding other explanatory variables constant.*

Holding a variable constant is exactly what we mean by *adjusting* for a variable. If we want to test the causal effect of ice cream sales on drownings, we have to adjust for temperature, which means to hold it constant. You can think of temperature being held constant at *any* value. For example, if temperature was held constant at 70 degrees F, how much would drownings change with each unit increase in ice cream sales. *That* is the interpretation of β_c. It's basically allowing us to look at the association between drownings and ice cream sales while shutting down the back-door path through temperature. The model asks, if I know the value of temperature, is there any relationship between drowning and ice cream sales? It doesn't' matter what value temperature is being held constant at. It could be 70 degrees, or 80 degrees, or 30.5 degrees. In this model, the estimated slope for ice cream sales would be the same at any value of temperature. What matters most is that the association between drownings and ice cream - as measured by the slope - is being estimated while holding temperature constant. The slope for temperature is interpreted in the same way. If I know the value of ice cream sales, is there any relationship between drownings and temperature?

11.1.3 Prior predictive check

As usual we should conduct a prior predictive check of our model.

```
n <- 100

set.seed(123)

#intercept
alpha.sim <- rnorm(n, mean = 13, sd = 2)

#slopes
beta.c.sim <- rnorm(n, mean = 0, sd = 0.25)
beta.t.sim <- rnorm(n, mean = 0, sd = 0.25)

#values of c (ice cream sales)
c_vals <- seq(min(d$sales.c), max(d$sales.c), length.out = 100)
```

```r
# Create a data frame with all lines
lines_df <- expand.grid(c_vals = c_vals, sim = 1:n)
lines_df$y <- alpha.sim[lines_df$sim] +
              beta.c.sim[lines_df$sim] * lines_df$c_vals +
              beta.t.sim[lines_df$sim] * 0

#plot
ggplot(lines_df, aes(x = c_vals+mean(d$ice_cream_sales_millions), y = y, group = sim)) +
  geom_line(alpha = 0.3, color = "blue") +
  labs(x = "Ice cream sales (millions)",
       y = "Drownings") +
  theme_classic()
```

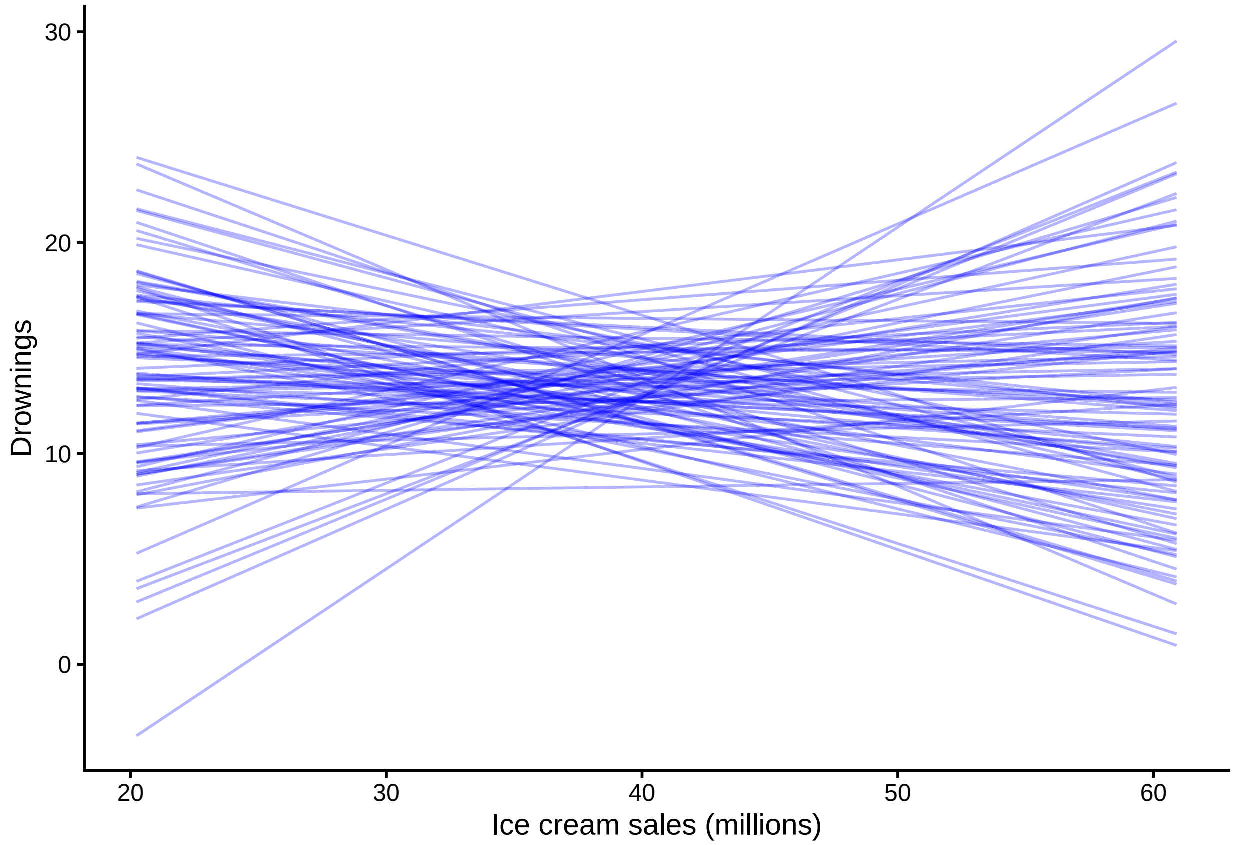


Figure 11.5: TODO: caption.

Compared to the last chapter, note that we now have to include a term in our model for the additional predictor variable. Just as we specify values for the predictor of interest, we have to specify values for the other predictor variables. One common approach is to hold other predictor variables constant at their mean while examining what the priors imply about other model terms. And that's what I've done here. Notice that I'm plotting what the priors imply about the relationship between drownings and ice cream sales (`c_vals`) while setting the value of temperature to 0, which is the mean on the mean-centered scale. Our priors aren't perfect, as we can see some negative values of drownings implied. We could consider tightening up the priors for the slopes, but my sense (without having much domain knowledge here) is that these priors are already pretty conservative, in that they say there's a 95% probability that the change in drownings per unit change in ice cream sales (or temperature) is no more than 0.5 while holding the other predictor variable constant. We'll proceed with these priors.

11.1.4 Fitting the multiple regression model

Let's go ahead and estimate the parameters in the multiple regression model, and then we will focus on interpretation. Specifying a multiple regression model in brms is simple. All we need to do is specify all the predictors in the model, separated by the `+` operator.

```r
#mean center temp
d$temp.c <- d$temperature_F - mean(d$temperature_F)

#specify model formula
m2.formula <- bf(drownings ~ 1 + sales.c + temp.c,
                family = gaussian)

#specify priors
m2.prior <- c(prior(normal(13, 2), class = Intercept),
              prior(normal(0, 0.25), class = b), #applies to all slopes
              prior(uniform(0, 5), class = sigma, lb=0, ub=5))

#compute the posterior
m2 <- brm(data = d,
         formula = m2.formula,
         prior = m2.prior,
         refresh = 0,
         seed=123)
```

```
## Running /Library/Frameworks/R.framework/Resources/bin/R CMD SHLIB foo.c
## using C compiler: ‘Apple clang version 17.0.0 (clang-1700.0.13.3)’
## using SDK: ‘MacOSX15.4.sdk’
## clang -arch arm64 -std=gnu2x -I"/Library/Frameworks/R.framework/Resources/include" -DNDEBUG
## In file included from <built-in>:1:
## In file included from /Library/Frameworks/R.framework/Versions/4.5-arm64/Resources/library/Sta
## In file included from /Library/Frameworks/R.framework/Versions/4.5-arm64/Resources/library/Rcp
## In file included from /Library/Frameworks/R.framework/Versions/4.5-arm64/Resources/library/Rcp
## /Library/Frameworks/R.framework/Versions/4.5-arm64/Resources/library/RcppEigen/include/Eigen/s
##   679 | #include <cmath>
##       |          ^~~~~~~
## 1 error generated.
## make: *** [foo.o] Error 1
```

```r
plot(m2)
```

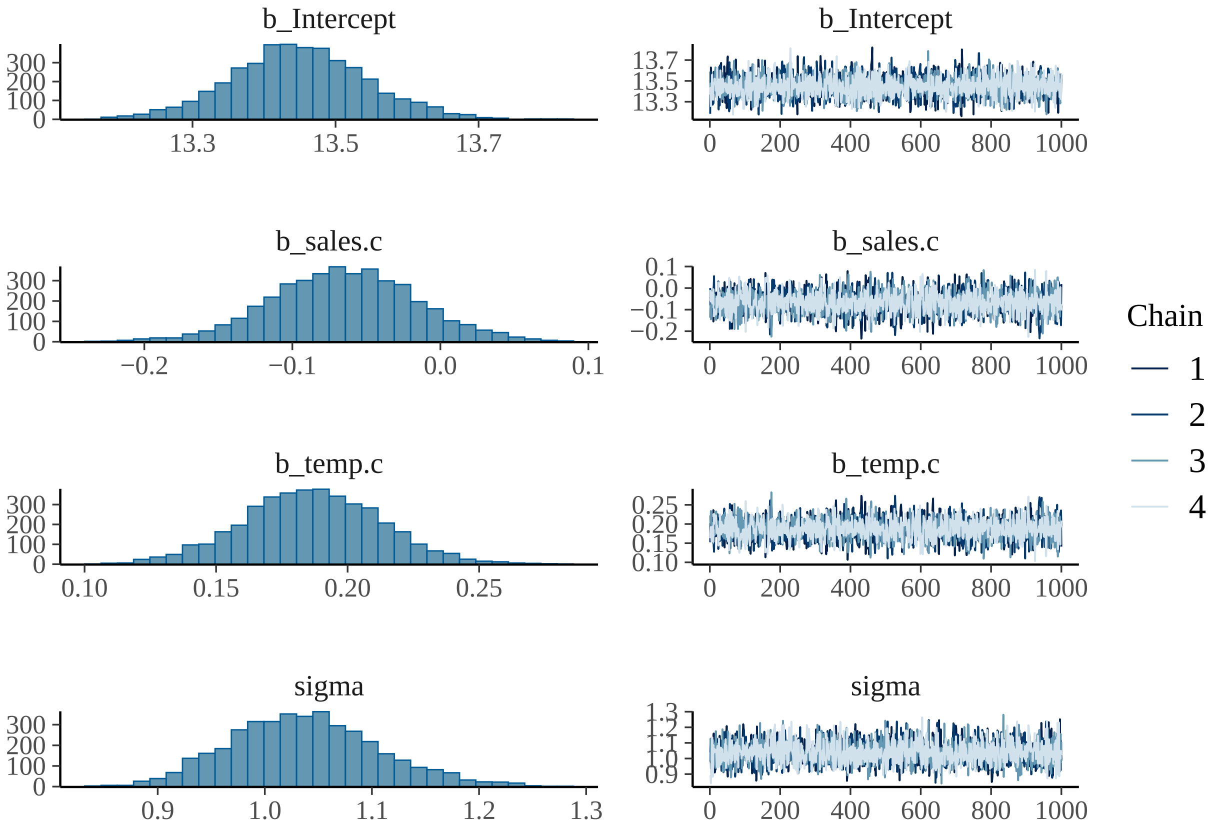


Figure 11.6: TODO: caption.

Our initial plot of the posteriors and the traceplots look good. It looks like each of the four parameters has converged. Let's take a look at the model summary:

```r
print(m2)
```

```
##  Family: gaussian
##   Links: mu = identity
## Formula: drownings ~ 1 + sales.c + temp.c
##    Data: d (Number of observations: 120)
##   Draws: 4 chains, each with iter = 2000; warmup = 1000; thin = 1;
##          total post-warmup draws = 4000
##
## Regression Coefficients:
##           Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## Intercept    13.45      0.10    13.26    13.64 1.00     3122     2018
## sales.c      -0.07      0.05    -0.16     0.03 1.00     2086     1836
## temp.c        0.19      0.03     0.13     0.24 1.00     2095     1763
##
## Further Distributional Parameters:
##       Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## sigma     1.04      0.07     0.91     1.18 1.00     2631     2070
##
## Draws were sampled using sampling(NUTS). For each parameter, Bulk_ESS
## and Tail_ESS are effective sample size measures, and Rhat is the potential
## scale reduction factor on split chains (at convergence, Rhat = 1).
```

Thus looks like the output of a linear model with a single predictor, except now we have a row for the intercept and each predictor in the regression coefficients section. The posterior mean slope for ice cream sales is now -0.07, with the 95% credible interval overlapping 0 (-0.16, 0.03). Conversely, the posterior mean slope for temperature is 0.19, with a 95% credible interval from 0.13 to 0.24, notably all in positive territory.

So what does this mean? The estimated slope for ice cream sales is *very* different than it was when we fit a simple linear model with ice cream sales being the only predictor. In this model, our interpretation is that when we hold temperature constant, there's not a consistent change in drownings when we change ice cream sales. The posterior mean is actually negative, which would imply that when we hold temperature constant, drownings actually *declines* for every unit increase in ice cream sales. But because the posterior distribution broadly overlaps 0, there's a lot of uncertainty about this association when we hold temperature constant. This tells us that although ice cream sales are predictive of drownings (when not accounting for temperature), ice cream sales don't seem to have a *causal* effect on drownings. Phew! I'm ready for some Ben and Jerry's.

What about the coefficient for temperature. Our interpretation there is that when we hold ice cream sales constant, there *is* value in knowing what tmeperature it is. Drowings appear to increase as temperature increases while holding ice cream sales constant. That's of course no surprise. A pistachio gelato on the Italian Riviera in July is sounding pretty good right now.

11.1.5 Prediction plots for multiple regression models

Consider our multiple regression model, which has drownings as the response variable and both ice cream sales and air temperature as the predictor variables. We want to know what the predictive power of ice cream sales is once we condition on air temperature. Of course the regression model output says there's no additional predictive power of ice cream scales when holding temperature constant, but how do we visualize that?

The idea here is that we can use our model to predict values of the response variable given particular values of the explanatory variables. In other words, what would we predict the number of drownings to be given particular values of ice cream sales and air temperature. Because we have a multiple regression model, we can make predictions of drownings for *any* combination of predictor values, even ones we haven't seen.

As an example, consider that the average July temperature is 76 F in the United States. July is also National Ice Cream Month, and let's say it's a banner month for ice cream sales with 60 million ice cream sales. What would the predicted number of drownings be under these circumstances? We can easily compute the predicted drownings when the temperature is 76 F and the number of ice cream sales is 60 for each sample of the posterior, and then summarize the outcome:

```
#extract the posterior samples
m.post <- as_draws_df(m2)
y.pred <- m.post$b_Intercept +
          m.post$b_sales.c*(60-mean(d$ice_cream_sales_millions)) +
          m.post$b_temp.c*(76-mean(d$temperature_F))

#summarize
mean(y.pred)
```

```
## [1] 15.74188
```

```
quantile(y.pred, probs=c(0.025, 0.975))
```

```
##     2.5%    97.5%
## 14.56280 16.94698
```

Here we see the mean of the posterior prediction is 15.74 drownings, with a 95% credible interval of 14.56 to 16.95. I wanted to walk through that example to show you that making predictions of a response variable is as simple as plugging in values of the explanatory variable. We could use the `fitted` function to make a prediction like that much more efficiently:

```
y.pred <- fitted(m2, newdata=data.frame(sales.c = 60 - mean(d$ice_cream_sales_millions)
                                        temp.c = 76 - mean(d$temperature_F)))
y.pred
```

```
##      Estimate Est.Error    Q2.5    Q97.5
## [1,] 15.74188 0.5928937 14.5628 16.94698
```

The most common way of graphically displaying the output of a multiple regression is to predict what would happen to the response variable as we change the value of one predictor while holding other predictors constant. This is essentially what a true experiment attempts to do, but of course not all research questions can be analyzed with an experiment (certainly not this one).

Let's look at an example. What would happen to the number of drownings if we change ice cream sales but hold air temperature constant at its average?

```
#values of mean-centered ice cream sales to predict on
x.pred <- seq(from=min(d$sales.c), to=max(d$sales.c), length.out=100)

#make the predictions
y.pred <- fitted(m2, newdata=data.frame(sales.c = x.pred,
                                        temp.c = 0))
y.pred <- cbind.data.frame(sales.c = x.pred, y.pred)

#put ice cream sales back on actual scale
y.pred$sales <- y.pred$sales.c + mean(d$ice_cream_sales_millions)

#now plot the relationship
ggplot(y.pred) +
  geom_smooth(data = y.pred, stat = "identity",
              fill = "slategray3", color = "black", alpha = 1, linewidth = 1,
              aes(x = sales, y = Estimate, ymin = Q2.5, ymax = Q97.5)) +
  labs(x = "Ice cream sales (millions)", y = "Predicted number of drownings") +
  ylim(0, 25) +
  theme_classic()
```

Here we see that if we hold the air temperature constant at its mean, there appears to be a mild decrease in the number of drownings. But note there's a lot of uncertainty around the posterior mean estimate as indicated by the 95% credible interval in blue. You could fit some pretty flat or even weakly positive slopes in that credible interval.

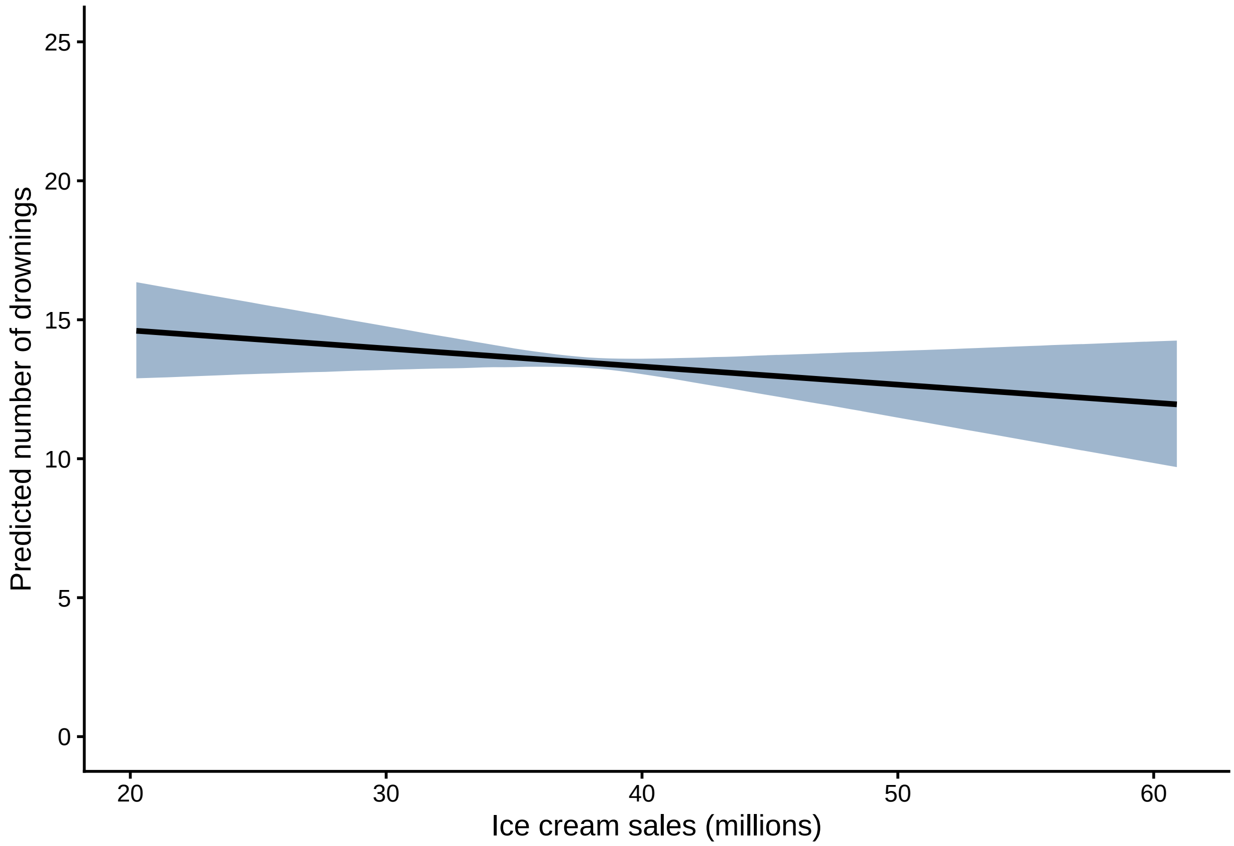


Figure 11.7: TODO: caption.

What if we did the opposite, holding the number of ice cream sales constant at its mean while varying air temperature? Let's see:

```r
#values of mean-centered temperature to predict on
x.pred <- seq(from=min(d$temp.c), to=max(d$temp.c), length.out=100)

#make the predictions
y.pred <- fitted(m2, newdata=data.frame(temp.c = x.pred,
                                        sales.c = 0))
y.pred <- cbind.data.frame(temp.c = x.pred, y.pred)

#put ice cream sales back on actual scale
y.pred$temp <- y.pred$temp.c + mean(d$temperature_F)

#now plot the relationship
ggplot(y.pred) +
  geom_smooth(data = y.pred, stat = "identity",
              fill = "slategray3", color = "black", alpha = 1, linewidth = 1,
              aes(x = temp, y = Estimate, ymin = Q2.5, ymax = Q97.5)) +
  labs(x = "Air temperature (F)", y = "Predicted number of drownings") +
  ylim(0, 25) +
  theme_classic()
```

Here we shoudl expect drownings to strongly increase with air temperature while holding ice cream sales constant. Remember that in our DAG we assumed temperature affects drownings by mediating the number of swimmers or along the ice cream sales pathway. Because we are holding ice cream sales constant, the prediction plot repesents the causal effect of air temperature on drownings via the indirect pathway involving the number of swimmers. We could generate another plot that represents the total effect of air temperature on predicted number of drownings, but we would need a different linear model that estimates the total effect. In the current linear model, the effect of temperature is estimated while holding ice cream sales constant.

11.2 DAG-informed predictors, categorical variables, what multiple regression is doing with predictor residual plot

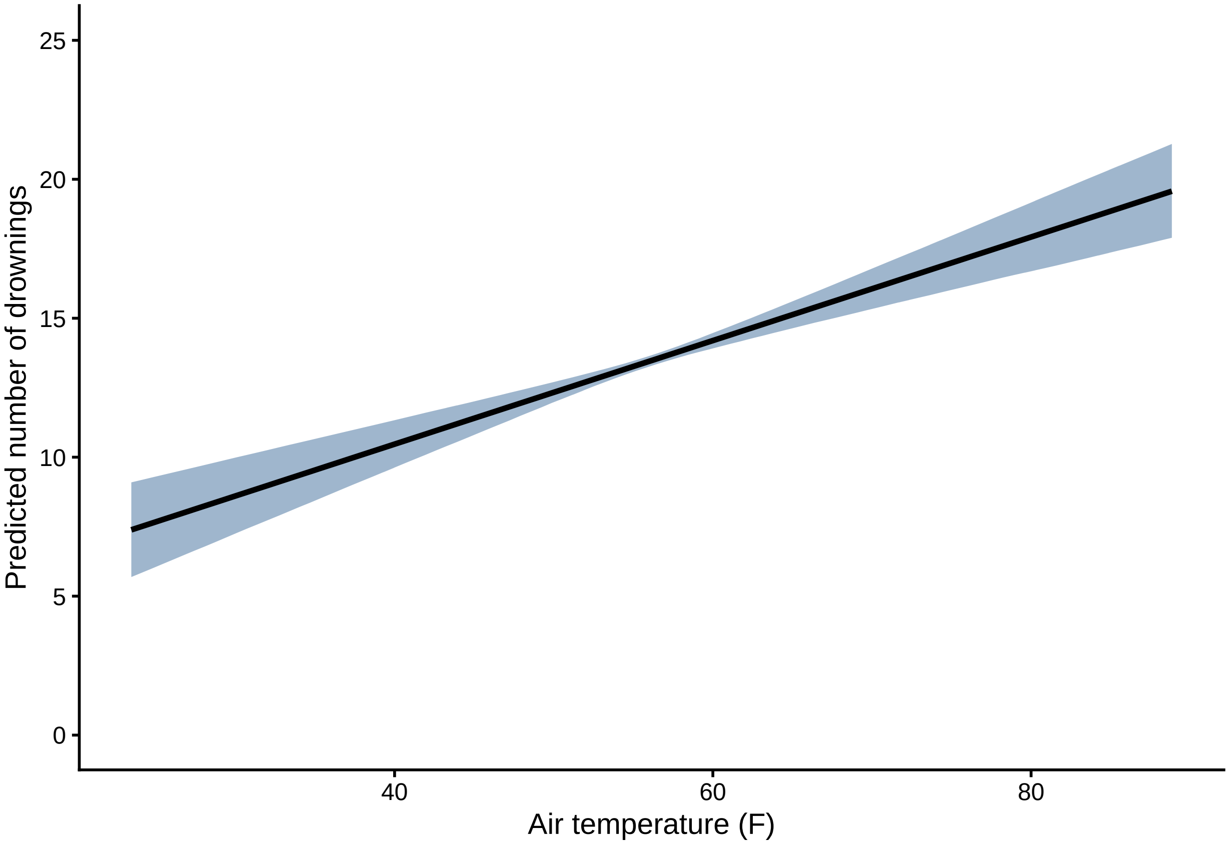


Figure 11.8: TODO: caption.

Chapter 12

Interaction effects

TODO

Chapter 13

Generalized linear models

TODO

Chapter 14

Multilevel models

TODO

Appendix A

Estimation with frequentist inference

Now that we have some basic principles of probability under our belt, we can turn our attention to the fundamental problem of inferential statistics: estimating parameter values from samples and characterizing uncertainty about our estimates. Having some basic understanding of probability was essential to explore the issues related to estimation, precisely because the language in which we will characterize uncertainty about parameter estimates *is* probability.

To keep things simple, we will explore the foundational principles of estimation in the context of the scientific problem we started to look at last chapter: estimating the prevalence of a disease in a population. We will use this example to develop the principles of estimation with different philosophies of inference, namely the frequentist and Bayesian approaches. Those terms should ring a bell from last chapter, as they represent the two different definitions of probability that we looked at.

Statistics education has been dominated by the frequentist approach. And there's good reason for that! Much of the scientific literature is dominated by the frequentist approach. In a way the situation has become a positive feedback loop. Frequentist methods are used by many (most?) professional scientists because those are the methods they are taught, and statistics instructors teach frequentist methods because those are the methods that are used. But Bayesian inference is being used more and more in the scientific literature, and for good reasons. Suffice to say that I think it's time to start teaching both approaches to inference. We'll start with the frequentist approach over the next two chapters, and then turn to Bayesian inference in the next chapter.

A.1 Frequentist estimates are point estimates

Recall the scenario. We have a population of 10,000 people, and we need to estimate the prevalence of a disease to determine if public health interventions will be enforced. Those interventions will be enforced if the prevalence is 10% or greater. We still assume our test is perfect, but we don't have the time or resources to test all 10,000 people, so will randomly sample individuals for testing. We'll also assume that anyone who is randomly selected will comply with the test. I know, these aren't realistic assumptions, but it's useful to make simplifying assumptions to develop first principles.

OK, let's further assume that we randomly sample $N = 100$ people for testing. Out of the 100 tests, we find 8 positives and 92 negatives. Based on this single sample, we **estimate** the prevalence of the infection as

$$\hat{p}_{infected} = \frac{8}{100} = 0.08$$

In other words, based on our sample we estimate that 8% of the population is infected. *Estimate* is a critical word here because we truly don't know what the actual prevalence of the infection is. We are trying to infer the true prevalence - that is, the parameter value - from a sample of data.

In frequentist inference, we try to draw conclusions about parameters in exactly this way. Frequentist inference assumes at the start that there is some true parameter value. We take a sample of data, and we estimate the parameter(s) of interest with the sample. Those parameter estimates are **point estimates**, in that they represent the single best estimate of the parameters of interest.

Because the frequentist approach assumes there is a single true parameter value, there's no way we can talk probabilisticaly about pramater values. For example, one might be tempted to ask "How probable is it that the prevalence of the infection is at least 10% given our sample of 8 of 100 infected?". In other words, we might want to know the conditional probability $P(p \geq 0.1 | \hat{p} = \frac{8}{100} = 0.08)$. But from a frequentist perspective, this doesn't make sense. The true prevalence of the infection is either greater than 0.1 or not regardless of the data we observe in our sample. The parameter p is completely fixed.

OK, but what are we supposed to make of our point estimate that 8% of the population is infected? Is it a good estimate or not? To interrogate the quality of an estimate with frequentist inference, we need to dig deeper and examine the concept of a sampling distribution.

A.2 Sampling distributions

The key element of frequentist inference is that it considers an estimate from a single sample to be only one of many possible outcomes. Just imagine repeating the sampling process over and over again. Every new sample of 100 tests will produce varying point estimates becuase of samplign error. We estimated 8% of the population is infected based on our single sample, but if we were to take another random sample, maybe we'd find the estimate is 9%, or 6%, or 11%. In other words, frequentist estimates are random variables!

When we estimate a parameter with a random sample, is it possible that some estimates are more likely than others? Absolutely. Because the estimate is a random variable, it can be described by a probability distribution. In frequentist statistics, the probability distribution used to describe a sample estimate is called the **sampling distribution**.

Let's assume that the true prevalence of the disease is 11% in our population of 10,000 people. If that's the case, how likely is it that the estimated prevalence from a sample of 100 people will be 8%, as we saw in our sample? In probability terms, what is $P(\hat{p} = \frac{8}{100} = 0.08|p = 0.11)$. You might recognize that in this case, the estimated prevalence $P(\hat{p}$ is a binomial random variable, so we can quantify this probability with the binomial equation:

$$P(X = 8) = \binom{100}{8} 0.11^8 (1 - 0.11)^{100-8} = 0.088$$

Of course we can quantify this easily in R:

```r
dbinom(x = 8, size = 100, prob = 0.11)
```

```
## [1] 0.0880522
```

So we see that when we take a random sample of 100 people from a population with a true prevalence of 11%, the probability of our point estimate being 8% is 0.088. How does that compare to other possible values of the point estimate? Well, we can easily compute the probability of all possible outcomes for the number of positive tests and then plot the resulting distribution (Figure A.1):

```r
#all possible values of positive tests out of N = 100
x <- seq(from = 0, to = 100, by = 1)

#probability of each outcome assuming prevalence is 11%
p.hat <- dbinom(x = x, size = 100, prob = 0.11)

#combine into a data frame
d <- cbind.data.frame(x, p.hat)

#plot the sampling distribution
ggplot(d, aes(x = x, y = p.hat)) +
  geom_col(width = 1, fill = "steelblue", color = "black") +
  labs(x = "Number of positives out of N = 100", y = "Probability") +
  xlim(0,30) +
  theme_classic()
```

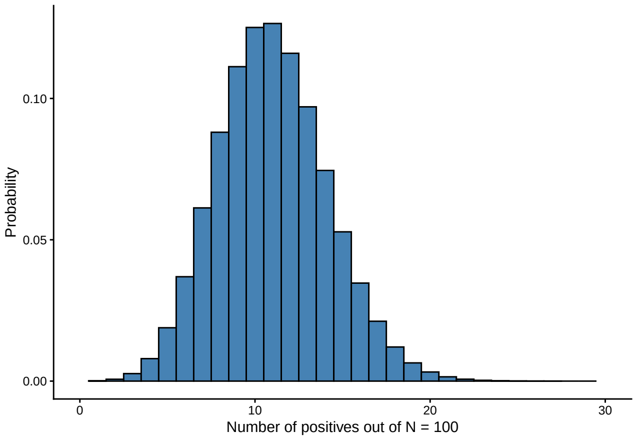


Figure A.1: Sampling distribution of the estimated prevalence of infection based on a sample of N = 100 individuals from a population where the true prevalence of the infection is 11%.

The resulting probability distribution (Figure A.1) shows the probability of each possible point estimate for the prevalence of the infection when we take a random sample of N = 100 from a population where the *true* prevalence is 8%. This is a *sampling distribution*! Sampling distributions show the probability distribution of all possible values for an *estimate* of a parameter when we take a random sample from the target population.

This is an extremely important concept. I have tried to make the case that the primary reason we need statistics is to guide decision-making about hypotheses in light of uncertainty. From a frequentist perspective, **the sampling distribution is an illustration of uncertainty about estimates taken from samples.** In this case, it shows us that when we take a random sample of 100 people, we won't necessarily find 11 positives (to give us an estimate of 11% prevalence) even if the true prevalence of the disease is 11%. It's certainly possible to get an estimate of 11% from a sample of N = 100, but it's almost just as likely to get an estimate of 10%, or 12%. Fundamentally these deviations in our estimates from the true parameter value are caused by random sampling error.

A.2.1 Sampling distributions are centered on the true parameter value

One thing you should observe in the sampling distribution in Figure @ref(fig:a01_chunk02) is that the distribution is centered on the true parameter value, 11%. That is a feature of sampling distributions. The expected value of a sampling distribution (the mean) *is* the true parameter value.

Now, you might be tempted to claim that our estimated prevalence of 8% is an inaccurate estimate. But that's not quite correct. When judging the accuracy of a parameter estimate from a frequentist perspective, you have to think about the distribution of possible estimates (the sampling distribution) rather than the single estimate you observed. Frequentist estimation is based on the idea of long-run frequencies of outcomes based on many random samples. In practice you only observe one, and because of sampling error, your one estimate is very likely to deviate from the truth. But accuracy of an estimator is judged on the expected value of the estimates, not the single estimate you observed.

This is where things get tricky for frequentist estimation. You have a single estimate in hand, but to judge the accuracy of that estimate, you have to think about the collection of possible, yet unobserved estimates that you would see if you repeated your sampling many times. It's an abstract idea! Our single estimate of 8% is indeed lower than the truth, but if you conducted sampling over and over again, you would see some estimates that are greater than the truth too. If estimates were consistently lower than the truth, then the estimates would indeed be biased.

I'm sure your wondering, "If I can't actually observe the sampling distribution, how do I know if my single estimate is accurate?". Good question! To judge the accuracy of your estimator, you really have to focus on aspects of the sampling design, namely the degree to which the observations you drew from the population were a random selection. Random sampling ensures that estimates - on average, across many samples - will be unbiased. In practice, there's nothing that we can quantify based on the concept of a sampling distribution that will tell us if your estimate is biased or not. To judge accuracy of estimates, you have to assess the sampling design, and especially assess whether the observations are a random sample.

A.2.2 Sampling distributions allow us to estimate precision

Sampling distributions are abstract and can't tell us much about whether a single estimate is biased or not, but they can tell us a lot about the precision of our estimates. Indeed, the width of the sampling distribution is a measure of precision. In our example, we can see there's a reasonable chance of seeing anywhere from 5-16 positives out of 100 tests when the true prevalence is 11%.

This variation is analogous to the variation in the number of heads you expect to see out of 10 flips of a fair coin. Although you expect 5 heads, you wouldn't be surprised to get 3, or 6, or 7 heads. In the same respect, we shouldn't be too surprised to see an estimated prevalence of 8% from a sample of N = 100 when the true prevalence is 11%. That variation is a feature of the sampling process, and it gets to the heart of uncertainty. When we sample from populations, there is random error in the outcome, and the width of the sampling distribution illustrates the uncertainty we should feel when we consider whether our sample estimate is any good. The wider the sampling distribution, the more uncertainty, and the less confidence we should feel about our estimate being close to the truth.

What factors affect the precision of estimates as represented by the width of the sampling distribution? Let's look at two: the sample size and variance.

A.2.2.1 Sample size affects precision

We previously identified sample size as one of the key drivers of precision. With the sampling distribution concept, we can interrogate that idea more precisely (pun intended!). We will continue assuming that the true prevalence of the disease is 11%, but we're not going to change the size of the sample that we draw from the population to estimate the prevalence. Let's construct sampling distributions where the sample size is 10, 100, or 1000 people:

```r
#all possible values of positive tests
x10 <- seq(from = 0, to = 10, by = 1)
x100 <- seq(from = 0, to = 100, by = 1)
x1000 <- seq(from = 0, to = 1000, by = 1)

#probability of each outcome assuming prevalence is 11%
p.hat10 <- dbinom(x = x10, size = 10, prob = 0.11)
p.hat100 <- dbinom(x = x100, size = 100, prob = 0.11)
p.hat1000 <- dbinom(x = x1000, size = 1000, prob = 0.11)

#combine into a data frame
d10 <- cbind.data.frame(n = 10, p.hat = x10/10, prob = p.hat10)
d100 <- cbind.data.frame(n = 100, p.hat = x100/100, prob = p.hat100)
d1000 <- cbind.data.frame(n = 1000, p.hat = x1000/1000, prob = p.hat1000)
d_all <- rbind(d10, d100, d1000)
```

And now we can plot the sampling distributions:

```r
#plot the sampling distribution
ggplot(d_all, aes(x = p.hat, y = prob)) +
  geom_col(data = d_all[d_all$n == 10, ],
           fill = "steelblue", color = "black", width = 0.1) +
  geom_col(data = d_all[d_all$n == 100, ],
           fill = "steelblue", color = "black", width = 0.01) +
  geom_col(data = d_all[d_all$n == 1000, ],
           fill = "steelblue", color = "black", width = 0.001) +
  facet_wrap(~ n, ncol = 1, scales = "free",
             labeller = labeller(n = function(x) paste("N =", x)))+
  labs(
    x = "Estimated proportion of positives",
    y = "Probability",
  ) +
  theme_classic() +
  theme(strip.background = element_rect(fill = "gray", color = "black"),
        strip.text = element_text(color = "black", size = 11))
```

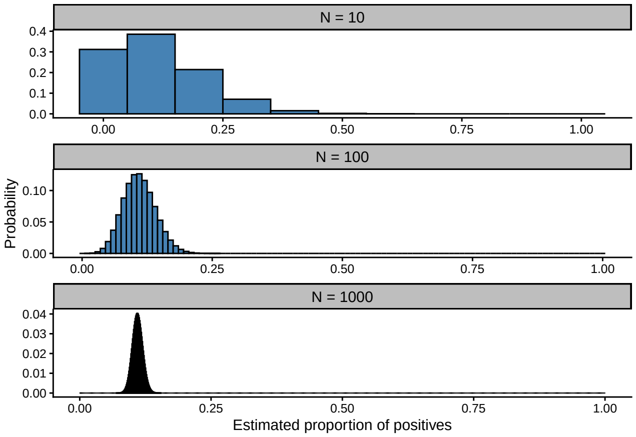


Figure A.2: Sampling distributions of the estimated prevalence of infection based on samples of size N = 10, 100, and 1000 individuals from a population where the true prevalence of the infection is 11%.

What do we notice about the sampling distributions under different sample sizes in Figure A.2? First, each sampling distribution remains centered on the true parameter value of 11% prevalence. The point estimate of a parameter remains an unbiased estimate of the parameter no matter the sample size. In other words, sample size has no effect on the *accuracy* of estimates.

Second, as the sample size increases, the width of the sampling distribution decreases. We expect the estimated prevalence to change much more from sample to sample when the sample size is 10 than when it is 100 or 1000. Greater sample sizes lead to more precise estimates. This should make sense. After all, if you increase the sample size all the way to the size of the target population, there would be no variation at all in estimates from sample to sample.

The **Law of Large Numbers** states this explicitly, that as the sample size N increases, the point estimate ultimately converges on the true parameter value. Let's simulate this process to get a good handle on it. Remember that we assumed a target population of 10,000 individuals. Below we will create a dataset of 10000 individuals with infection status classified as 1 (infected) or 0 (not infected), where 11% of the population is infected:

```
#create infection status for 10000 people with 11% infected
status <- c(rep(1, 1100), rep(0, 8900))

#randomly order the observations
set.seed(124)
status <- sample(status, replace=FALSE)
head(status)
```

```
## [1] 0 1 0 0 0 0
```

```
#confirm the proportion infected is 11%
mean(status)
```

```
## [1] 0.11
```

Now let's quantify the point estimate for the proportion infected as we increase the sample size from $N = 1$ to $N = 10000$. To get a sense for this, we can see above that the first individual is not infected (`status` $= 0$), so based on that individual with a sample size of $N = 1$, the estimated prevalence is 0%. Then we look at the second individual, who is infected (`status` $= 1$), making the point estimate $\hat{p} = \frac{1}{2} = 0.5$ based on $N = 2$. The third individual is not infected, making the point estimate $\hat{p} = \frac{1}{3} = 0.33$, and so on. We can compute the point estimate for each individual observation with the code below:

```
#creates the cumulative sum from each observation
status.cumsum <- cumsum(status)

#cumulative sample size
n <- seq_along(status)

#quantify the point estimates
p.hat.infected <- status.cumsum/seq_along(status)

#create a dataframe
sim.law.large <- cbind.data.frame(status.cumsum, n, p.hat.infected)
head(sim.law.large)
```

```
##   status.cumsum n p.hat.infected
## 1             0 1      0.0000000
## 2             1 2      0.5000000
## 3             1 3      0.3333333
## 4             1 4      0.2500000
## 5             1 5      0.2000000
## 6             1 6      0.1666667
```

Now let's graph the estimated proportion against the sample size:

```
ggplot(sim.law.large, aes(x = n, y = p.hat.infected)) +
  geom_line(color = "steelblue", linewidth=0.7) +  # Line graph for p.hat.infected
  geom_hline(yintercept = 0.11, color = "red", linetype = "dashed") +  # Truth line
  labs(
    x = "Sample Size (n)",
    y = "Estimated Proportion Infected",
    title = ""
  ) +
  theme_classic() +
  theme(
    plot.title = element_text(hjust = 0.5)  # Center the title
  )
```

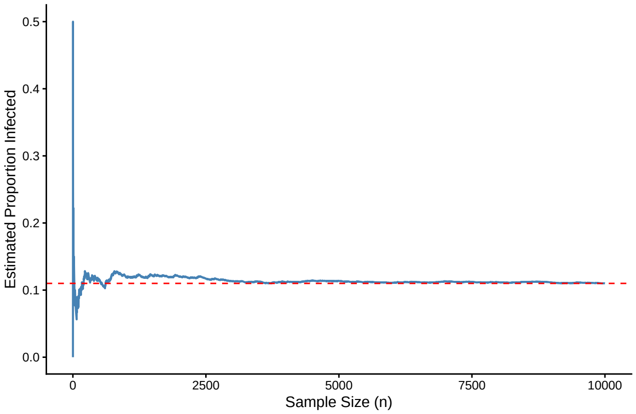


Figure A.3: Illustration of the Law of Large numbers, showing that as the sample size approaches the size of the target population, the point estimate converges on the true parameter value.

Figure A.3 illustrates the Law of Large numbers nicely. We see that the point estimate of the parameter moves around wildly when sample size is small, but eventually the sample size converges on the true value of 0.11. There are deviations between the point estimate and the true parameter value at basically every sample size below the size of the target population, but these deviations get smaller as the sample size increases. This phenomenon is true when trying to estimate any type of parameter, whether it is a proportion, mean, median, variance, etc.

A.2.2.2 Variance affects precision

Sample size is not the *only* factor that affects precision. Precision is also affected by the variance of the underlying random variable. Precision of a point estimate decreases as variance increases. This should make intuitive sense. The idea is that when there's greater variability in the underlying distribution of the random variable, there's greater potential to select extreme observations into the sample, which increases the variability of the point estimate.

We can see this for our example of estimate disease prevalence. Remember that for the binomial distribution, the variance is determined by the true proportion: $p(1-p)$ The variance reaches its maximum value when $p = 0.5$. Figure A.4 compares sampling distributions for when the prevalence is 11% vs. 50% with an identical sample size of N=100.

```r
#all possible values of positive tests
x <- seq(from = 0, to = 100, by = 1)

#probability of each outcome assuming prevalence is 11%
p11 <- dbinom(x = x, size = 100, prob = 0.11)
p50 <- dbinom(x = x, size = 100, prob = 0.50)

#combine into a data frame
d11 <- cbind.data.frame(p = 11, p.hat = x/100, prob = p11)
d50 <- cbind.data.frame(p = 50, p.hat = x/100, prob = p50)
d_all <- rbind(d11, d50)

#plot the sampling distribution
ggplot(d_all, aes(x = p.hat, y = prob)) +
  geom_col(data = d_all[d_all$p == 11, ],
           fill = "steelblue", color = "black", width = 0.01) +
  geom_col(data = d_all[d_all$p == 50, ],
           fill = "steelblue", color = "black", width = 0.01) +
  facet_wrap(~ p, ncol = 1, scales = "free",
             labeller = as_labeller(c(
            `11` = "p = 0.11, N = 100",
            `50` = "p = 0.50, N = 100"
        ))) +
  labs(
    x = "Estimated proportion of positives",
    y = "Probability",
  ) +
  theme_classic() +
  theme(strip.background = element_rect(fill = "gray", color = "black"),
        strip.text = element_text(color = "black", size = 11))
```

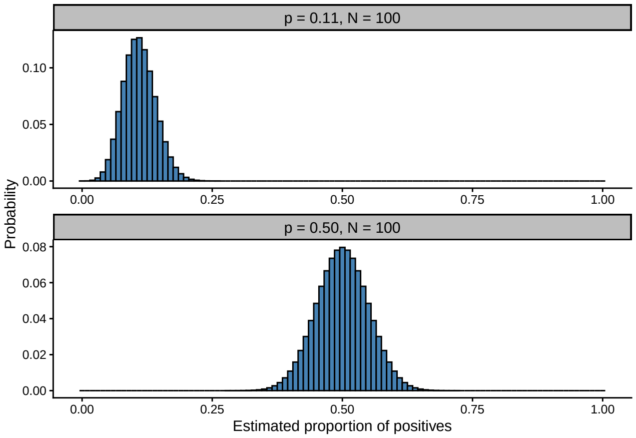


Figure A.4: Sampling distributions of the estimated prevalence of infection based on a sample size of 100 when the prevalence of infection is 11% (low variance) vs. 50% (high variance).

Figure A.4 shows that the sampling distributions are centered in different locations, each being centered on the true parameter value (11% vs. 50%). But what you should also notice is that sampling distribution is wider (less precise) when the prevalence is 50% (higher variance) than when it is 11% (lower variance). The take-home here is that estimates will be less precise when sampling from random variables that have greater variance. Unlike sample size, this isn't something you can control.

This might make more sense with a different type of variable. Imagine that you're estimating the mean height of a group of people. We'll assume height follows a normal distribution where the true mean height is $\mu = 65$ inches. Let's further assume that we draw a sample of 100 people to estimate the height, but let's do so from two populations that differ in the variance of height. In one population, we'll assume the standard deviation is $\sigma = 5$ inches, and in the other population we'll assume the standard deviation is $sigma = 2$ inches. We can visualize these normal distributions and see that there's more variation among individual height when $\sigma = 5$ than when $\sigma = 2$ (Figure A.5)

We now go ahead and generate 10,000 replicate samples of N = 100 from the

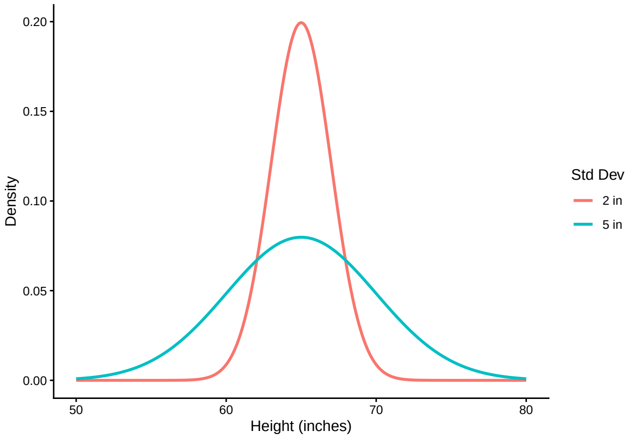


Figure A.5: Assumed distributions of height where the mean height is 65 inches and the standard deviation is either 2 or 5 inches.

underlying populations, one where $\mu = 65$ and $\sigma = 2$ and the other where $\mu = 65$ and $\sigma = 5$.

```
set.seed(123)

#sample sizes to evaluate
stdevs <- c(2, 5)

#empty data frame to store results
sampling_results <- data.frame()

#loop to simulate sampling process at each sample size 10000 times
for (stdev in stdevs) {

  #draw samples 10,000 times and compute sample mean
  sample_means <- replicate(10000, mean(rnorm(n = 100, mean = 65, sd = stdev)))

  #store results
  sampling_results <- rbind(sampling_results,
                            data.frame(stdev = stdev,
                                       sample_mean = sample_means))
}

#plot
ggplot(sampling_results, aes(x = sample_mean)) +
  geom_histogram(aes(y = after_stat(density)), bins=50, fill = "skyblue", color = "bla
  facet_wrap(~ stdev, nrow = 2, scales = "fixed",
               labeller = as_labeller(function(x) paste("SD =", x))) +
  theme_minimal() +
  labs(x = "Sample Mean Height (in)", y = "Probability density" ) +
  theme_classic() +
  theme(
    strip.background = element_rect(fill = "gray", color = "black"),
    strip.text = element_text(size = 10, face = "bold"),
    plot.title = element_text(size = 16, face = "bold", hjust = 0.5)
  )
```

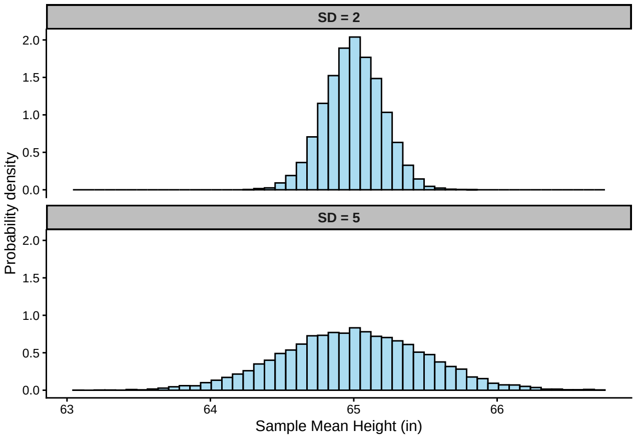


Figure A.6: Sampling distributions of the estimated mean height from samples of N = 100 when the true mean is 65 inches but the standard deviations of height are either 2 vs. 5.

Figure A.6 shows that while both sampling distributions are centered on the true value of 65 inches (that is, the estimates are accurate), the sampling distribution generated from the population with $\sigma = 5$ is much wider than the sampling distribution generated from the population with $\sigma = 2$. In other words, precision is much lower when drawing samples from the population with greater variability of individual height, assuming equivalent sample sizes.

The take-home here is that precision will be greatest and uncertainty will be minimized when drawing large sample sizes from populations that have low variance. While you can control sample size as part of the study design, you can't control the underlying variance of the random variable being studied. Assuming equivalent sample size, estimates from distributions with more variation will more noisy than estimates from distributions with less variation.

A.2.3 Sample size affects the shape of sampling distributions

Going back to the original comparison of sampling distributions at different sample sizes in Figure A.7, you might ahve noticed that the shape of the sampling distribution increasingly resembles a normal distribution as the sample size increases. This happens to be a consequence of the **Central Limit Theorem**, which states that the distribution of a sample estimates will be approximately normal at large sample sizes regardless of the shape of probability distribution for the underlying probability distribution.

Let me illustrate this last point with a different example, this time with a quantitative random variable. Suppose you were interested in estimating the average distance people live from an ice cream shop. We'll assume that the distribution of distances people live from an ice cream shop is skewed to the right. This means most people live close to an ice cream shop, and fewer and fewer people live far from an ice cream shop. This kind of random variable can be described by a **poisson distribution**, which is a probability distribution that often describes things we can count. Let's create a graph of the probability distribution assuming that the true mean distance to ice cream shop is 1 km:

```r
#distances in kilometers
x <- seq(0, 10, by = 1)
prob <- dpois(x = x, lambda = 1) #lambda is the mean distance

#data frame for plotting
d <- cbind.data.frame(x, prob)

# Plot the probability distribution using ggplot2
ggplot(d, aes(x = x, y = prob)) +
  geom_bar(stat = "identity", fill = "skyblue", color = "black", alpha = 0.7) +
  labs(x = "Distance to ice cream shop (km)",
       y = "Probability") +
  theme_classic()
```

We can clearly see in Figure A.7 that the underlying distribution of distances is not normal! It's strongly skewed to the right and represents a poisson distribution. Now let's simulate the process of randomly sampling homes to estimate the mean distance people live from an ice cream shop.

```r
set.seed(123)

#sample sizes to evaluate
sample_sizes <- c(1, 2, 5, 10, 50, 100, 500, 1000)

#empty data frame to store results
```

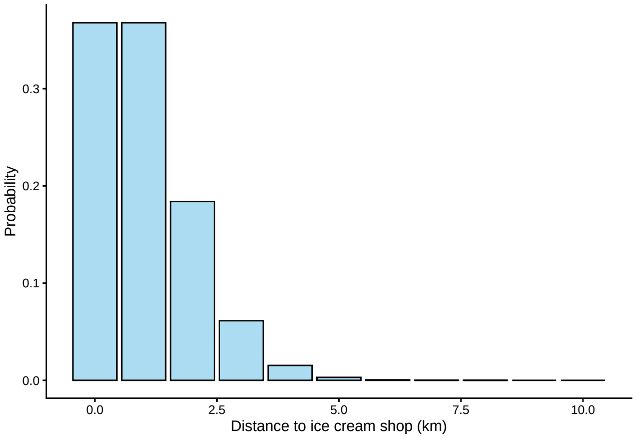


Figure A.7: Probability distribution of distance to ice cream shop. Distance to ice cream shop was assumed to follow a poisson distribution with a mean of 1 km.

```r
sampling_results <- data.frame()
#loop to simulate sampling process at each sample size 10000 times
for (sample_size in sample_sizes) {

  #draw samples 10,000 times and compute sample mean
  sample_means <- replicate(10000, mean(rpois(sample_size, lambda = 1)))

  #store results
  sampling_results <- rbind(sampling_results,
                            data.frame(sample_size = sample_size,
                                       sample_mean = sample_means))
}

#plot
ggplot(sampling_results, aes(x = sample_mean)) +
  geom_histogram(aes(y = ..density..), bins = 30, fill = "skyblue", color = "black", alpha = 0.7)
  facet_wrap(~ sample_size, nrow = 2, scales = "free",
             labeller = as_labeller(function(x) paste("N =", x))) +
  theme_minimal() +
  labs(x = "Sample Mean", y = "Probability density" ) +
  theme_classic() +
  theme(
    strip.background = element_rect(fill = "gray", color = "black"),
    strip.text = element_text(size = 10, face = "bold"),
    plot.title = element_text(size = 16, face = "bold", hjust = 0.5),
    plot.caption = element_text(size = 10, hjust = 0.5)
  )
```

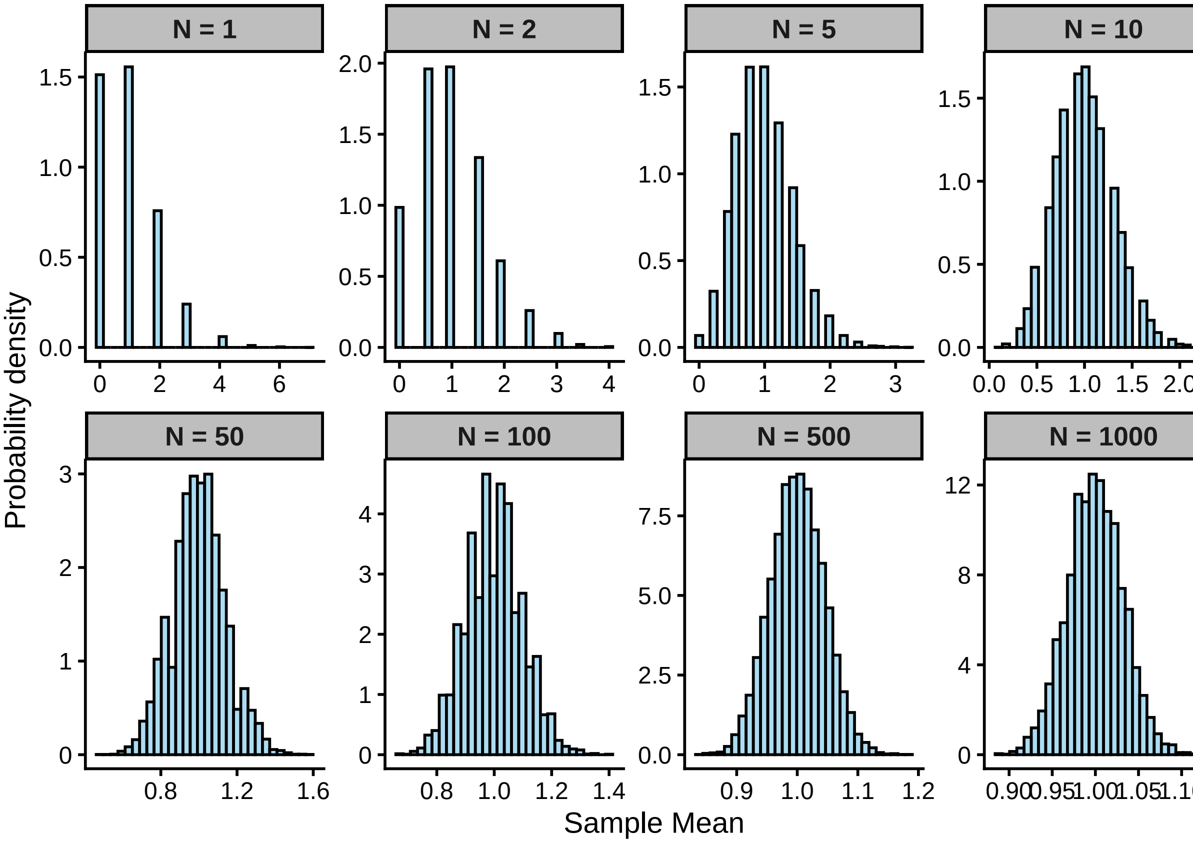


Figure A.8: Sampling distribution for the estimated mean distance people live from an ice cream shop based on samples of size N = 1, 2, 5, 10, 50, 100, 500, and 1000. Note that the scale of the x-axis varies among plots (becoming more narrow as sample size increases) in order to highlight the shape of each sampling distribution.

Figure A.8 shows that as the sample size increases, the shape of the sampling distribution increasingly resembles a normal distribution. This happens even though the underlying probability distribution for the random variable is not normal. Indeed, we can see that when the sample size is only one, the sampling distribution is identical to the distribution of the underlying variable. But it doesn't take many observations for the sampling distribution to take on the classic bell shape of a normal distribution.

A.2.4 Summary points on sampling distributions

That was a lot! But there's a reason for that. Understanding the concept of sampling distributions is really important for understanding how frequentist inference is carried out. Let's wrap up this section by summarizing some fundamental principles of frequentist inference related to sampling:

1. If we take repeated samples from the same population to estimate a parameter, the estimates from different will not be the same due to random sampling error. The distribution of sample estimates given a specific sample size is the sampling distribution.
2. The expected value of the sampling distribution is the true value of the parameter regardless of sample size. Because sample size doesn't affect statistical accuracy, the primary way to minimize bias in estimates is through research design strategies.
3. As sample size increases, the precision of estimates increases, and the more likely a sample estimate will be close to the true parameter value. (Law of Large Numbers)
4. The sampling distribution increasingly approximates a noral distribution as the sample size increases (Central Limit Theorem)

A.3 Quantifying uncertainty: standard error and confidence intervals

Now that we know a) the width of the sampling distribution represents uncertainty about a parameter estimate and b) the sampling distribution is approximately normal under large sample sizes, we can turn to quantifying the degree of uncertainty in estimates from a sample. We'll quantify uncertainty in two ways, the standard error and confidence intervas.

A.3.1 Standard error

Rather than just visually examining the width of a sampling distribution to gauge precision, we can quantify the variation in a sampling distribution. Recall that the standard deviation represents the variation among values for a random variable that follows a normal distribution. If the sampling distribution has a normal distribution, we can simply quantify the standard deviation of the sampling distribution as a measure of precision. The standard deviation of the sampling distribution is called the **standard error** and is quantified as the ratio of the standard deviation of the underlying random variable to the square root of the sample size:

$$SE = \frac{SD}{\sqrt{N}}$$

This is a generic formula to quantify the standard error of an estimate taken from a sample when the sampling distribution is approximately normal. It can be applied to estimates of means, proportions, model coefficients, and more. The standard deviation is quantified differently for these parameters, so you will need to substitute SD for the particular standard deviation formula for the parameter of interest. For example, the standard deviation of proportions is quantified as $\sqrt{p(1-p)}$, so the standard error of a proportion is:

$$SE_{\hat{p}} = \frac{\sqrt{p(1-p)}}{\sqrt{N}}$$

Consider our goal of estimating the prevalence of infection with a sample of 100 individuals when the true prevalence was 11%. Thus, the standard error is

$$SE_{\hat{p}} = \frac{\sqrt{0.11(1-0.11)}}{\sqrt{100}} = 0.0312$$

We know the estimate based on our single sample of N = 100 was 8%, which has a straightforward interpretation. But how do we interpret a standard error of 0.03? In a general sense, higher values for the standard error simply indicate more uncertainty (less precision) about the sample estimate. In other words, we should have more confidence in an estimate with a standard error of 0.03 than an estimate with a standard error of 0.1.

But maybe you have noticed there is a problem here. How can you compute the standard error of a sample estimate when you don't know the true population parameters? Look again at the formula for the standard error of the estimated prevalence. It includes the true prevalence as part of the formula for the standard deviation in the numerator. But we don't know the true prevalence! We're working through this process of estimation precisely because we don't know the true value of the parameter.

So what are we to do? In practice, the best we can do is substitute our estimate(s) for the true parameter value(s) in the formula for the standard deviation. You have to imagine the situation where we really don't know the true prevalence is 11%. All we know is that we've estimated the prevalence to be 8% based on N = 100. We use this information to estimate the standard error as

$$SE_{\hat{p}} = \frac{\sqrt{\hat{p}(1-\hat{p})}}{\sqrt{N}} = \frac{\sqrt{0.08(1-0.08)}}{\sqrt{100}} = 0.0271$$

You can see the estimated standard error is not that different from the true standard error [1].

A.3.2 Confidence intervals

Recall the empirical rule, which states that for a normal distribution approximately two thirds of observations will be within one standard deviation of the mean, and approximately 95% of observations will be within two standard deviations of the mean. Thus, when we obtain a point estimate of a parameter from a sample and estimate the standard error (i.e., the standard deviation of the sampling distribution), we can use that information to quantify a *range* of possible values for the true population parameter at a given level of confidence. Such a range is called a **confidence interval**.

A.3.2.1 Quantifying confidence intervals when you know the standard deviation

Remember the standard normal distribution (Z)? The units of Z in the standard normal are in units of standard deviations. Let's find the value of Z that includes 95% of observations from the mean (which is 0 in a standard normal):

```
#what value of Z has 2.5% of observations below it?
qnorm(0.025, mean = 0, sd = 1, lower.tail=TRUE)
```

```
## [1] -1.959964
```

Here we see that 2.5% of observations are below the value Z = -1.96. Because the normal distribution is symmetric, this means that 2.5% of observations are also above the value Z = 1.96. Figure A.9 shows the standard normal distribution highlighting the middle 95% of observations.

If we think about a standard normal distribution representing a sampling distribution, by definition there is a 95% chance that a point estimate of a parameter from a randomly drawn sample will be between Z = -1.96 standard errors below the true parameter value and 1.96 standard errors above the true parameter value. The Z score for our particular point estimate is

$$Z = \frac{\hat{\mu} - \mu}{\sigma_{\mu}}$$

[1] It's a little awkward that the standard error estimate here is *lower* than the true value. This is actually a known bias under low sample size. Variances tend to be underestimated at low sample size, leading to lower estimates of the standard error at low sample sizes. There are different types of bias-correction factors that can be employed to adjust for this issue, but here we stick with the basic calculation of a standard deviation for a proportion for simplicity.

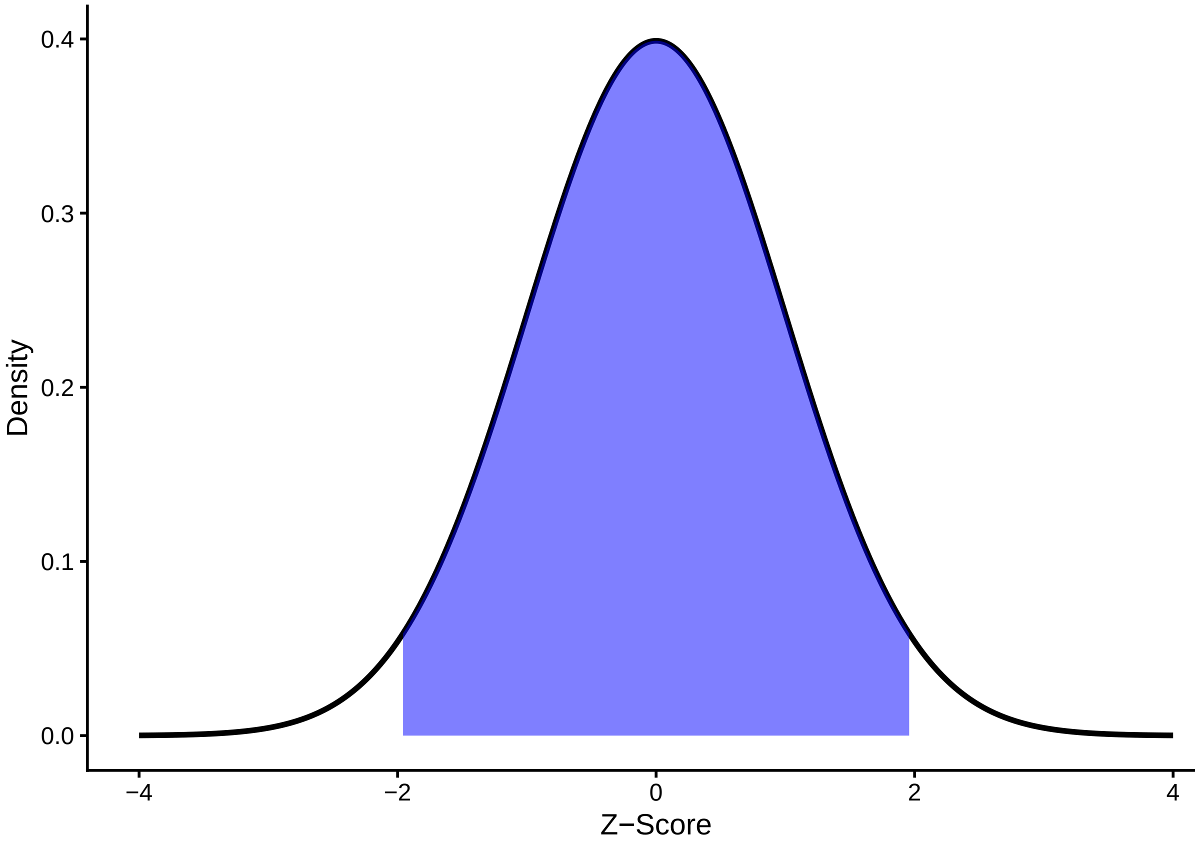


Figure A.9: Standard normal distribution showing hte middle 95% of observations between Z = -1.96 and Z = 1.96.

where $\hat{\mu}$ is the estimated mean, μ is the true mean, and σ_μ is the true standard error. Thus, it must be the case that in 95% of random samples,

$$-1.96 < \frac{\hat{\mu} - \mu}{\sigma_\mu} < 1.96$$

Rearranging this inequality, we obtain a 95% confidence interval:

$$\hat{\mu} - 1.96 * \sigma_\mu < \mu < \hat{\mu} + 1.96 * \sigma_\mu$$

The inequality expresses the idea that we should be 95% confident that a point estimate $\hat{\mu}$ from a random sample will be within the range of values defined by $\hat{\mu} \pm 1.96 * \sigma_\mu$.

Can we apply this to our estimated proportion of infection? Yes, we sure we can. Remember that under the central limit theorem, we expect sampling distributions to be approximately normally distributed under large sample size, and N = 100 is a reasonably large sample size. If our point estimate is 0.08 and the true standard error is 0.03, we can be 95% confident that the true parameter value is within the range $0.08 \pm 1.96 * 0.03$, which corresponds to $0.021 < \mu < 0.139$.

There's no reason we have to quantify confidence intervals at a level of 95%. Indeed we can quantify a confidence interval at any degree of confidence using the following general formula:

$$\hat{\mu} \pm Z_{\alpha/2} * \sigma_\mu$$

where α is the **significance value**. The significance value refers to the probability of observations in the tails, so the level of confidence is $1-\alpha$. For example, $\alpha = 0.1$ for a 90% confidence interval. The quantity of $\pm Z_\alpha * \sigma_\mu$ is called the **margin of error**.

To quantify a 90% confidence interval for our example of infection prevalence, we simply need to find the value of Z leaving 10% of observations in the tails (5% of observations in each tail). Here's how we can do it in R:

```
#90% confidence level
alpha <- 0.1

#Z for 90% confidence level
z <- abs(qnorm(alpha/2, mean = 0, sd = 1, lower.tail=TRUE)) #abs for positive Z

#lower confidence limit: point estimate - z*SE
0.08 - z*0.03
```

```
## [1] 0.03065439
```

```
#upper confidence limit: point estimate + z*SE
0.08 + z*0.03
```

```
## [1] 0.1293456
```

We see that the 90% confidence interval for the point estimate 0.08 and standard error 0.03 is 3% - 13%. In other words, we can be 90% confident that the true proportion infected is between 3% and 13%. Notice that the range of confidence interval narrows as the confidence level decreases.

A.3.2.2 Quantifying confidence intervals when you don't know the standard deviation

Wait a second! The formula to quantify a confidence interval with the standard normal (Z) assumes that we know the standard deviation (and therefore the standard error) with certainty. We only know the true standard error in this case because we simulated the data, but again, we're estimating quantities precisely because we don't know them.

It turns out there's only a slight modification in our approach when we don't know the true standard deviation and standard error. But the approach differs when we're estimating proportions versus means. When we're estimating proportions with an unknown standard error, all we have to do is substitute the estimated standard error for the true standard error. In other words, a 95% confidence interval for a proportion is quantified as

$$\hat{p} - 1.96 * SE_{\hat{p}} < p < \hat{p} + 1.96 * SE_{\hat{p}}$$

Thus, if all we had was our single sample with 8 out of 100 positive tests, the estimated standard error is 0.0271 (from above), and the 95% confidence interval can be quantified as is $0.08 \pm 1.96 * 0.0271$, which corresponds to $0.027 < \mu < 0.133$. That's it! Really straightforward.

What about when we're estimating means? For means, things are slightly more complicated. The reason is that the normal distribution has two parameters, the mean and standard deviation, both of which must be estimated with data. In contrast, the standard deviation of a proportion is quantified directly from the value of the proportion itself. Ultimately this means there is slightly more uncertainty when estimating a standard deviation for normally distributed variables compared to binomially distributed variables. That additional uncertainty gets included as part of the process of quantifying the confidence interval.

How does it work? Rather than using the standard normal distribution, we use a normal probability distribution that has *slightly* heavier tails (i.e., slightly wider) compard to the standard normal distribution. This modified distribution is called a **t-distribution**. The degree to which more probability is added to the tails of a *t*-distribution relative to a standard normal is dependent on sample size. When the sample size is low, the standard deviation is being estimated with less precision, and more probability density is added to the tails to reflect greater uncertainty.

The particular shape of a *t* distribution is specified by the **degrees of freedom**, which in this case is defined as $df = N - 1$. Thus, the appropriate *t*-distribution for a sample of size N = 50 has $df = 50 - 1 = 49$. As the sample size increases, the *t*-distribution converges on the standard normal distribution, which can be seen in Figure @ref(fig:a01__chunk16).

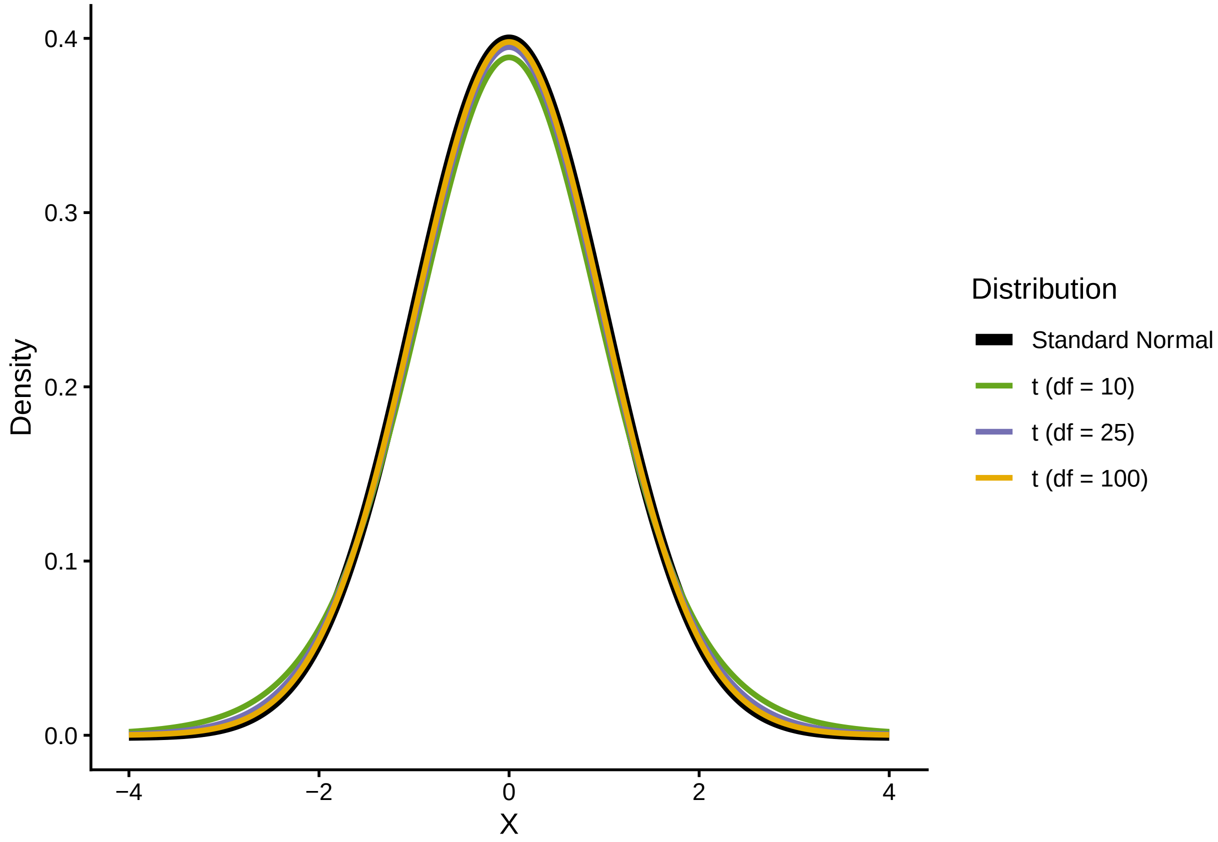


Figure A.10: Comparisong of t-distributions with varying degrees of freedom to the standard normal distribution (Z)

Now that you have a sense for how the *t*-distribution compares to the standard normal, let's look at how confidence intervals are quantified with the *t*-distribution.

$$\hat{\mu} \pm t_{\alpha/2,df} * SE_{\mu}$$

The formula is largely the same as before, except we've replaced the Z with a t distribution at a specified significance level *and* degrees of freedom ($t_{\alpha,df}$), and we've replaced the true standard error with our estimate of the standard error (SE_{μ}).

Let's work through an example. Suppose you're thinking about ice cream again, and you're interested in estimating the average weight of a single scoop of ice cream from your local ice cream parlor. You decide to go to the parlor on 10 randomly selected days and times, and you order a single scoop and weight it. The code chunk below creates a vector of the weights for 10 ice cream cones assuming a true normal distribution of weight with mean 113 g and standard deviation 15 g. A 95% confidence interval is then computed for the estimated weight:

```r
set.seed(123)

#sample size
n <- 10

#ice cream weights (g) in sample of N = 10
ic <- rnorm(n = n, mean = 113, sd = 15)

#estimated mean
ic.mean <- mean(ic)
ic.mean
```

```
## [1] 114.1194
```

```r
#estimated standard deviation
ic.sd <- sd(ic)

#estimated standard error
ic.se <- ic.sd/sqrt(n)
ic.se
```

```
## [1] 4.524195
```

```r
#95% confidence level
alpha <- 0.05

#t for 95% confidence level and df = n-1
t <- abs(qt(alpha/2, df = n-1, lower.tail=TRUE)) #abs for positive t
t
```

```
## [1] 2.262157
```

```
#lower confidence limit: point estimate - t*SE
ic.mean - t*ic.se
```

```
## [1] 103.8849
```

```
#upper confidence limit: point estimate + t*SE
ic.mean + t*ic.se
```

```
## [1] 124.3538
```

We can see that the estimated (`p.hat`) mean weight is 113.02 g, the estimated standard error (`se`) is 3.66, the appropriate t value based on the degrees of freedom is 2.26, and the resulting confidence interval is 104.7 - 121.3 g. You should notice that the t value in this case, 2.26, is greater than the Z of 1.96 for a 95% level of confidence. This reflects added uncertainty for estimating the standard deviation, and therefore the standard error, and ultimately results in a slightly wider confidence interval than compared to the interval based on a known standard error.

A.3.2.3 Interpreting confidence intervals

OK, back to prevalence of the infection. Based on our sample of N = 100 individuals, we estimated the prevalence was 8% with a 95% confidence interval of 2.7 - 13.3%. In other words, we can be 95% confident that true proportion infected is between 2.7% and 13.3%. I am purposely NOT saying that there's a 95% *probability* that the true proportion infected is between 2.7% and 13.3%.

Why can't we interpret our confidence interval in terms of probability? Well, this turns out to be one of the quirks of frequentist estimation. The frequentist definition of probability is about long-run frequencies! In order to determine confidence intervals probabilistically, you have to imagine have hundreds of thousands of confidence intervals from hundreds of thousands of random samples. If we have many 95% confidence intervals from many random samples, then we expect 95% of the confidence intervals to include the true population parameter. But any individual confidence interval? A frequentist would say that the true parmaeter is either in that single confidence interval, or it is not, so there is no probability to speak of for a single iteration.

We can simulate this situation to see exactly what I mean. We'll use the `rbinom` function to simulate 1000 random samples of N = 100 individuals when the probability of infection is 11%, then quantify a 95% confidence interval for each random sample, and finally quantify the proportion of confidence intervals that include 11%.

```
set.seed(124)

#10000 random samples of N = 100
sims <- 1000
alpha = 0.05
n <- 100
x.sims <- rbinom(n = sims, size = n, prob = 0.11)

lcl <- rep(NA, sims)
ucl <- rep(NA, sims)

for(i in 1:sims){
  x <- x.sims[i]
  p.hat <- x/n
  se <- sqrt(p.hat*(1-p.hat))/sqrt(n)
  z <- abs(qnorm(alpha/2, lower.tail=TRUE))
  lcl[i] <- p.hat - z*se
  ucl[i] <- p.hat + z*se
}

contains.truth <- lcl<=0.11 & ucl>=0.11
mean(contains.truth)
```

```
## [1] 0.939
```

```
#plot 20-randomly selected confidence intervals
set.seed(26) # Ensure reproducibility for sampling
indices <- sample(1:sims, 20)
ci.data <- data.frame(
  Interval = 1:20,  # Vertical positions
  LCL = lcl[indices],
  UCL = ucl[indices],
  ContainsTruth = contains.truth[indices]
)

ggplot(ci.data, aes(y = Interval, xmin = LCL, xmax = UCL, color = ContainsTruth)) +
  geom_errorbarh(height = 0.3, size = 1) +  # Plot horizontal confidence intervals
  geom_vline(xintercept = 0.11, linetype = "dashed", color = "black") +  # Add referen
  scale_color_manual(values = c("TRUE" = "black", "FALSE" = "#E6AB02")) +  # Highlight
  theme_classic() +
  labs(
    x = "Proportion Infected",
    y = "Randomly selected 95% CIs",
    color = "Contains true proportion (11%)"
  ) +
  theme(
    legend.position = "top",
    axis.text.y = element_blank(),  # Remove y-axis labels for clarity
    axis.ticks.y = element_blank()
  )
```

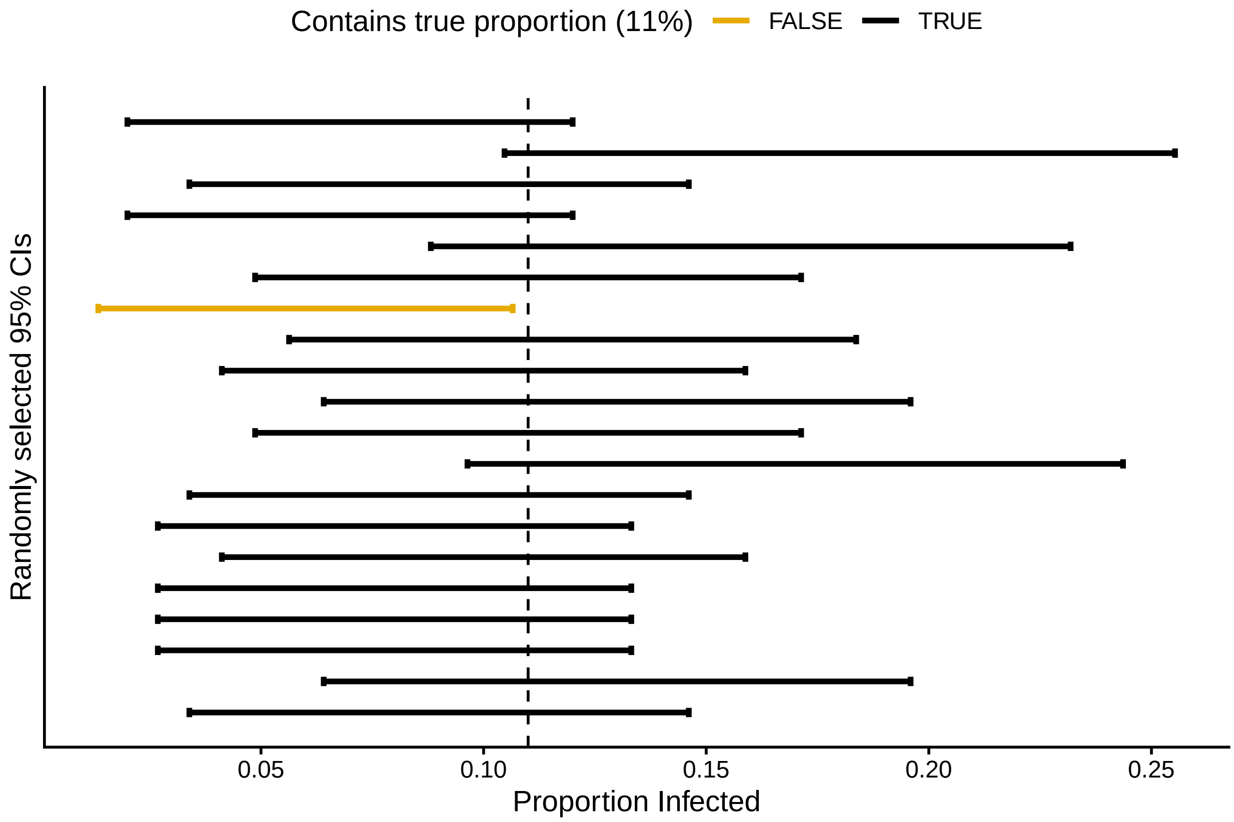


Figure A.11: TODO: caption.

Figure A.11 shows 20 randomly selected 95% confidence intervals for the proportion infected, highlighting confidence intervals that either do or do not include the true value of 11% infected. Note that of the 20 randomly-selected confidence intervals, 19 of 20 contain the truth, which is what we expect for a 95% confidence interval. If we look at the entire sample of 1000 random samples, 93.9% of them contained the true parameter of 11%. Why wasn't it exactly 95%? Well, there's sampling error even in our simulation process because we're not generating an infinite number of samples [2].

So the only time you should interpret frequentist confidence intervals probabilistically is when you have many confidence intervals. Which, of course, is almost never. We usually have a single point estimate and confidence interval based on a single sample. When that's the case, all we can do is say we are 95% confident (or whatever level of confidence is quantified) that the true parameter value is in the interval. This is a source of great frustration and misunderstanding, so if that's how you feel, you are in good company! You might even be asking yourself, what does it even mean to say we are"95% confident" about the true parameter being in a single interval? Good question. To me, it sort of sounds like a strength of belief, almost like a Bayesian interpretation of probability. Stay tuned.

A.3.3 The standard normal approximation does not work well under low sample size

The methods we've used in this section to quantify confidence intervals assume that the sampling distribution for an estimate follows an (approximately) normal distribution. This works well for continuous variables with underlying normal distributions, and it can even work well for random variables that don't have an underlying normal distribution. Indeed, we quantified a confidence interval for a categorical variable (proportion infected) assuming an approximately normal sampling distribution. As we now know from the central limit theorem this works well under reasonably large sample sizes.

So what is a large sample size? Well, there are some general rules of thumb. When estimating proportions for categorical variables, the central limit theorem application generally works when $np \geq 10$ and $n(1-p) \geq 10$. For example, in our example random sample where N = 100 and $p = 0.11$, these conditions are (barely) satisfied: $np = 100 * 0.11 = 11$ and $n(1-p) = 100 * (1-0.11) = 89$. Of course, when you're estimating the proportion, you don't actually know the true proportion! In that case, you can use the estimated proportion to check if the rule of thumb holds.

For our sample where we observed 8 out of 100 infected individuals, the rule would not hold: $100*0.08 = 8$. What then? There are many alternative methods that have been applied for estimating confidence intervals for proportions under low sample size (or extreme values of the proportion near 0 or 1). For example, one approach to quantifying a 95% confidence interval for a proportion when $np < 10$ or $n(1-p) < 10$ is the Agresti-Coull Method. This method makes adjustments to the sample size ($n' = n + 4$) and number of successes ($x' = x + 2$) to compute an adjusted proportion ($p' = \frac{x'}{n'}$), which can then be used to quantify an adjusted standard error and confidence interval by replacing p and

[2] It turns out there's also a little bias due to the tendency to underestimate the standard deviation at low sample size.

n with p' and n' in the formulas above. The `binom` package also has a function, `binom.confint` that can quantify an Agresti-Coull confidence interval directly by setting the `methods` argument to `ac`:

```
#agresti coull confidence interval
binom.confint(x = 8+2, n = 100+4, conf.level=0.95, methods="ac")
```

```
##          method  x   n       mean     lower     upper
## 1 agresti-coull 10 104 0.09615385 0.0513591 0.1697197
```

You can see in this case that the confidence interval is a adjusted a bit higher compared to the interval we quantified.

Appendix B

Decision-making with frequentist estimates

Note: add correct way to get binomial probabilitiy based on likelihood of the observation under the null for p-value

We have randomly sampled 100 people from a population of 10,000 and found that 8 of the 100 tested positive for a viral infection. Our frequentist estimate of the proportion infected is 0.08. We have quantified uncertainty about that estimate represented by the standard error (SE = 0.027) and confidence intervals. Based on the confidence interval, we can be 95% confident that the true proportion infected is between 0.026 and 0.134.

So there we have it. We have an estimate and have quantified uncertainty about the estimate with frequentist methods. What do we do with this information? Recall that our goal is to determine whether the prevalence of the infection is greater than 10%, a threshold that would trigger public health interventions. What should we conclude?

The decision in front of us is the crux of scientific inference. We gather structured observations to test ideas, which we can represent by numerical quantities. In this case, the quantity is a simple proportion. In other cases, we might have a causal hypothesis about how one variable affects another variable. We can represent those ideas with numerical quantities as well, and we'll start doing that in a few chapters. But before we get there, we need to address this question of how we make a *decision* about a scientific idea based on our sample data. In this chapter, we'll take a look at the tools to make decisions about hypotheses when estimates are made with frequentist methods.

B.1 Framework of classical hypothesis testing

B.1.1 State null and alternative hypotheses

Classical hypothesis testing boils down to this. How likely are the observed sample data under the hypothesis that nothing interesting is going on? The idea that nothing interesting is going on is called the **null hypothesis**, abbreviated H_0, which represents a hypothesis of "no effect". Indeed, the word *null* means *nothing.*

Consider some examples of null hypotheses in practice. If you're comparing the effect of a drug on blood pressure relative to a control (e.g., placebo), the null hypothesis is that there is no effect of the drug. If you're examining the effect of greenspace around a person's home on their mental health status, the null hypothesis is that there's no effect of green space. If you're examining the effect of pollution on the number of fish, the null hypothesis is that there's no effect.

This is straightforward for situations where one has a causal hypothesis, but what about our case of describing the prevalence of a disease. In cases of description, the typical approach is to define the null hypothesis as a particular numerical value. For example, we could the null hypothesis that the prevalence is exactly 10%, which is the cutoff for triggering public health interventions.

What if we conclude the prevalence is *not* 10%? In such a case, we reject the null hypothesis and conclude the data support the **alternative hypothesis**, H_A. The alternative hypothesis is simply the opposite of the null hypothesis. There is an effect of the drug on blood pressure, there is an effect of greenspace on mental health, there is an effect of pollution on fish population size, and the prevalence of the viral illness is not 10%.

Because we are using data to inform our decision about hypotheses, we need to define null and alternative hypotheses as specific quantitative values of the parameter(s) of interest. For example, to test if the prevalence of the viral illness exceeds 10%, we could test the following statistical hypotheses:

$$H_0 : p_{infected} = 0.10 H_A : p_{infected} > 0.10$$

Here the null hypothesis is that 10% of the population is infected, and the alternative hypothesis is that greater than 10% are infected, reflecting values that would trigger interventions. The alternative hypothesis in this case is directional, specifying values only greater than 10%. This is called a **one-sided** hypothesis. Once our statistical hypotheses are specified numerically, we confront the hypotheses with data we've collected to see which hypothesis is more consistent with the data.

Defining statistical hypotheses numerically is straightforward for simple questions of description, but it can also be done when making causal hypotheses.

For example, if we are examining the effect of pollution on fish populations, we could frame our statistical hypotheses numerically in this way:

$$H_0 : \mu_{polluted} - \mu_{unpolluted} = 0 H_A : \mu_{polluted} - \mu_{unpolluted} \neq 0$$

Here we've identified the mean number of fish in polluted and unpolluted waters as the parameters of interest. The null hypothesis says the difference in the mean number of fish between polluted and unpolluted areas is 0, meaning there is no difference between the means, which is what we might expect if pollution has no effect on fish populations. In this case the hypothesis is **two-sided** because it does not specify directionality of a potential effect, allowing for the possibility that fish are more populated in polluted than unpolluted areas, or vice-versa. For reasons we will see later, it is generally more conservative (and often more appropriate for causal hypotheses) to use two-sided tests.

Ultimately the precise way in which null and alternative hypotheses are framed depends on the research question and the parameters that best represent the scientific hypothesis being tested. Sometimes those parameters are simple means and proportions, and other times they will be values from more complex quantitative models, such as linear models that we'll turn our attention to in a few chapters.

B.1.2 Assume the null hypothesis is true

So far we have seen that hypothesis testing involves flipping a scientific hypothesis on its head by forming a null hypothesis that assumes *no effect*, or that nothing interesting is going on. As if that wasn't peculiar enough, we now presume that the null hypothesis is true. Why do that?

To understand the focus on a null hypothesis, it's critical to remember that the foundation of frequentist inference is the idea that probability is defined in terms of long-run frequency. The concept of the sampling distribution illustrates this idea nicely. Consider that we're testing the hypothesis that pollution affects fish population size by comparing the mean number of fish between polluted and unpolluted areas. The frequentist approach assumes there is some true value for the difference in the mean population size between these areas. We can estimate the difference in the means between areas in each sample, but the challenge we face is that we are likely to find *some* difference in the means between areas even if only by chance. Indeed, the sampling distribution shows us that the estimates of any quanity from random samples will form a probability distribution around the true value of the parameter. Estimates close to the true parameter are most likely, but deviations from the truth happen because of sampling error.

Now, in practice we don't know the true difference in the means, which is why we're doing the study in the first place! Without knowing the true difference in the means, we can't describe the actual sampling distribution. But what we can do is assume the parameter takes on a particular value. If we do that, we can describe the sampling distribution based on that assumed parameter value and the estimated standard error, and then we can ask how likely it is that we'd see the particular estimate in our sample data under that assumption.

Fundamentally, frequentist hypothesis testing is basically about asking how likely the observed data are from a sample under some assumed parameter value. But what value should we assume for the parameter? Should we assume a value that reflects our scientific hypothesis? Perhaps the scientific hypothesis is that pollution will reduce fish populations because of toxicity effects on survival and reproduction. Makes sense. But what parameter value would best reflect that hypothesis? Should we assume the mean population size is 200 in polluted waters and 500 in unpolluted waters, such that the difference of the means is 300? Or what about a smaller effect, such as 467 in unpolluted waters and 425 in polluted waters, for a difference of 42?

Hopefully you see the problem. There's an infinite number of possible values that the parameter could take on to be consistent with our scientific hypothesis. Frequentists solve this problem by isolating the *one* parameter value that would *not* be consistent with teh scientific model, namely that there's no difference in fish populations between polluted and unpolluted waters. In other words, the null hypothesis! The only numerical value for the difference in the means consistent with the null hypothesis is 0. It's much more tractable to form the sampling distribution around the null hypothesis, and then quantify how likely the data are in light of that null hypothesis. The idea is that if the data aren't all that likely under the null hypothesis, then maybe the null hypothesis is wrong!

So ultimately the frequentist approach focuses on the null hypothesis because we can isolate a single parameter value for the null hypothesis to form a sampling distribution and quantify the likelihood of our data from that sampling distribution. The sampling distribution for the null hypothesis is called the **null distribution**. If the sample data are relatively likely under the null distribution, then we can conclude the data are consistent with the null hypothesis. If the sample data are very unlikely under the null distribution, then we will reject the null hypothesis and conclude the data support the alternative hypothesis, which is just the inverse of the null.

Assuming the null hypothesis is true is not an intuitive approach given that our scientific hypothesis is that there *is* an effect of pollution, but under the frequentist approach, it's the most practical option. If your head is spinning and you're thinking there are some problems with this approach, stay tuned! We'll address some of those challenges, but first let's finish working through the process.

B.1.3 Quantify the likelihood of the data under the null hypothesis: P-values

Once we have defined the null hypothesis and have the data in hand, we can go ahead and quantify how likely the data are assuming the null hypothesis is true. Remember: the general idea here is that when we randomly sample from populations, we always expect some deviation between a sample estimate and the true parameter value because of sampling error. We can minimize that sampling error by maximizing sample sizes, but it never completely goes away. With a defined null hypothesis, we can quantify the probability of observing the data from our particular sample occurring simply by chance with a **P-value**.

Let's quantify a P-value for our question about the prevalence of the viral infection. To keep things simple for now, we're going to test the null hypothesis that the proportion infected is exactly 10%, with a two-sided alternative hypothesis that the proportion infected is not 10%:

$$H_0 : p_{infected} = 0.10 H_A : p_{infected} \neq 0.10$$

We observed 8 out of 100 infected, for an estimated prevalence of 8%. The question now before us is how likely that observation is under the null hypothesis that the prevalence is 10%, which we will quantify as a P-value. Actually, this isn't *quite* a P-value. Sorry, but it gets just a tad more complicated (and confusing). Rather than simply quantifying the probability of the observed data (the estimated prevalence of 8%) under the null distribution, we actually quantify the probability of the observed data *or data that are more extreme than the observed data relative to the null hypothesis.* What values are more extreme than 8% relative to the null hypothesized value of 10% when we take a sample of N = 100? Well, 7%, 6%, 5%, 4%, and so on. And because we are assuming a two-sided alternative hypothesis, values of 12%, 13%, 14%, etc. are equally or more extreme than 8% in relationship to the null hypothesized value.

Why do this? Why not just quantify the probability of the data point we observed under the null hypothesis that $p_{infected} = 0.1$ and call it good? We know how to do that with the `dbinom` function:

```r
#exact probability of 8 out of 100 positives when p = 0.1
dbinom(x = 8, size = 100, prob = 0.1)
```

```
## [1] 0.114823
```

Indeed, the probability of 8 positives out of 100 tests when the prevalence is 10$ is exactly 0.115. But we don't stop there for a couple reasons. First, the goal of null hypothesis testing is to understand how unusual the data are relative to the null hypothesized value, and we can't really tell how unusual the observation is without considering how likely more extreme observations would have been. Think of those observations as a sort of reference point. Second, consider a scenario where we had sampled more than 100 people. For example, assume we conducted N = 1000 tests. With 1000 tests there is much more resolution on the parameter space, where we can estimate prevalence down to the thousandths in comparison to the hundredths with a sample size of 100. That means the probability of any given outcome is lower with N = 1000 tests than N = 100 tests, simply because there are more possible outcomes. Indeed, the exact probability of 80 positives out of 1000 tests with p = 0.1 is only 0.004. That observation alone is unusual in large part because there are so many possible outcomes. This problem gets even worse for continuous distributions, where we can't quantify the probability of any single outcome at all because there are an infinite number of possible outcomes!

The solution to this issue is the quantify the probability of a range of possible values. Because the goal of null hypothesis testing is to try to identify if the data are unusual under the null hypothesis, we focus specifically on the probability of all outcomes *at least* as extreme as the data we observed. Thus, the P-value is defined as the probability of sample estimates at least as extreme as the one we observed relative to the null hypothesis, all conditional on the null hypothesis being true. And when we use a two-sided alternative hypothesis, we consider observations that are at least as extreme as the one we observed both above and below the null hypothesized value.

```r
#all possible values of positive tests out of N = 100
x <- seq(from = 0, to = 100, by = 1)

#probability of each outcome assuming prevalence is 11%
p.hat <- dbinom(x = x, size = 100, prob = 0.10)

#combine into a data frame
d <- cbind.data.frame(x, p.hat)
d$pval <- ifelse(d$x <= 8 | d$x >=12, TRUE, FALSE)

#plot the sampling distribution
ggplot(d, aes(x = x, y = p.hat, fill = pval)) +
  geom_col(width = 1, color = "black", show.legend=FALSE) +
 scale_fill_manual(values = c("TRUE" = "darkorange", "FALSE" = "steelblue")) +
  labs(x = "Number of positives out of N = 100", y = "Probability") +
  xlim(0,30) +
  theme_classic()
```

Figure B.1 illustrates the null distribution assuming the true prevalence is 10%, and it highlights the values used to quantify the P-value. This includes the actual observation of 8 positive tests, plus all lower values, as well as observations of 12 and above. We can quantify the P-value in this way:

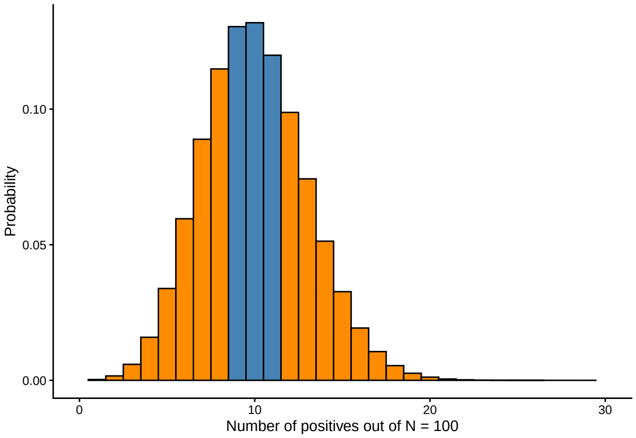


Figure B.1: Null distribution of the prevalence of infection based on a sample of N = 100 assuming the true prevalence is 10%. The values highlighted in orange make up the P-value for a two-sided hypothesis test when there are X = 8 positives.

```
sum(dbinom(x = c(0:8, 12:100), size = 100, prob=0.1))
```

```
## [1] 0.6178408
```

Here we see the p-value is 0.62. In other words, there was a 62% chance of seeing 8 or fewer positive tests, or 12 or more positive tests, if the true prevalence was exactly 10%.

Now in our particular circumstance, it is probably more appropriate to use a one-sided test because we are explicitly interested in the the possibility that the prevalence of the infection is *above* 10%. In such a case, we specify our statistical hypotheses in this way:

$$H_0 : p_{infected} = 0.10 H_A : p_{infected} > 0.10$$

A one-sided tests requires a different approach to quantify the P-value than a two-sided test. Whereas we include observations in both tails of the null distribution to quantify the P-value for a two-sided test, we include only values into the tail specified by the alternative hypothesis in a one-sided test. For our case of 8 out of 100 tests, that means we need to include the probability of getting 8, 9, 10, 11, 12, and so on out of 100 positives, which is illustrated in Figure B.2.

```
#all possible values of positive tests out of N = 100
x <- seq(from = 0, to = 100, by = 1)

#probability of each outcome assuming prevalence is 11%
p.hat <- dbinom(x = x, size = 100, prob = 0.10)

#combine into a data frame
d <- cbind.data.frame(x, p.hat)
d$pval <- ifelse(d$x >= 8, TRUE, FALSE)

#plot the sampling distribution
ggplot(d, aes(x = x, y = p.hat, fill = pval)) +
  geom_col(width = 1, color = "black", show.legend=FALSE) +
 scale_fill_manual(values = c("TRUE" = "darkorange", "FALSE" = "steelblue")) +
  labs(x = "Number of positives out of N = 100", y = "Probability") +
  xlim(0,30) +
  theme_classic()
```

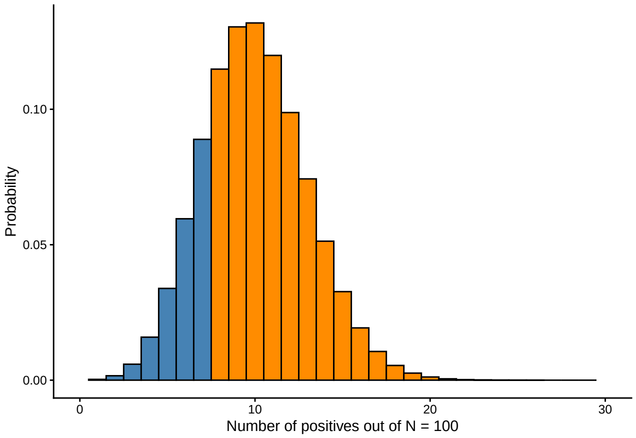


Figure B.2: Null distribution of the prevalence of infection based on a sample of N = 100 assuming the true prevalence is 10%. The values highlighted in orange make up the P-value for a one-sided hypothesis test when there are X = 8 positives.

The P-value for the one-sided test is:

```
sum(dbinom(x = 8:100, size = 100, prob=0.1))
```

```
## [1] 0.7939491
```

We see that the P-value for a two-sided hypothesis test is 0.79. In other words, there was a 79% chance of seeing observations of 8 or more positive tests when the true prevalence is 10%.

B.1.4 Making a decision based on the P-value and significance value

Hypothesis testing involves making a decision between two competing statistical hypotheses: the null and alternative. Indeed, these hypotheses are defined in a way to be mutually exclusive. If the prevalence of the disease is 10% (null hypothesis), then it must be some value other than 10% (alternative hypothesis). The goal is to determine if the data are more consistent with one hypothesis than another.

To accomplish this goal, the P-value is used to examine the likelihood of the data under the null hypothesis. The idea is that if the P-value is high, it means that the data are very consistent with the null hypothesis, and so perhaps the null hypothesis is true. Indeed, we saw the P-value was 0.79 when considering the null hypothesis that the prevalence of the infection is 10% against the alternative that the prevalence is greater than 10%. In other words, there was a 79% of seeing our data, or more extreme observations, if the prevalence is really 10%. Because that set of observations is quite likely under the null hypothesis, we conclude that the data support the null hypothesis and are inconsistent with the alternative hypothesis that the prevalence is greater than 10%.

But when should we conclude the opposite, that the data *don't* support the null hypothesis. Remember the **significance value** from the last chapter? The significance value, α, represents a specific probability of observations being in the tails of the null distribution. In null hypothesis testing, the significance value represents the threshold for rejecting the null hypothesis. The most typical value of α is 0.05, such that if the P-value is below 0.05, one should conclude that the data do not support the null hypothesis. The idea is that if there's a less than 5% chance of seeing the observed data (or more extreme observations) under the null hypothesis, then maybe the null hypothesis isn't true. Those observations are so rare that we typically reject the null hypothesis and conclude that the data supports the alternative hypothesis.

For example, suppose that instead of finding X = 8 positives out of N = 100 tests, we find X = 17 positives instead. Let's compute the P-value under the null that the prevalence is 10% and the alternative that the prevalence is greater than 10%:

```r
sum(dbinom(x = 17:100, size = 100, prob=0.1))
```

```
## [1] 0.02059881
```

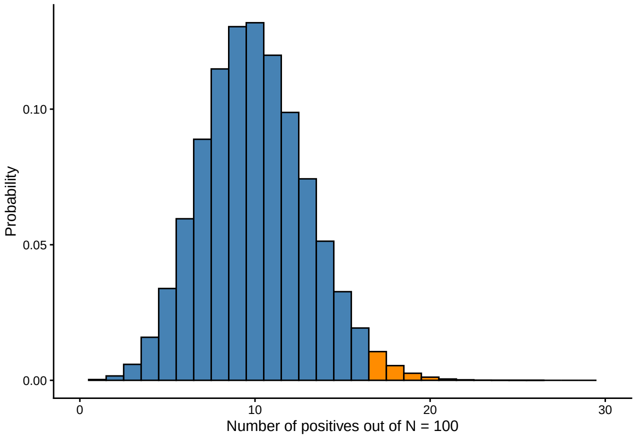


Figure B.3: Null distribution of the prevalence of infection based on a sample of N = 100 individuals assuming the true prevalence is 10%. The values highlighted in orange make up the P-value for a one-sided hypothesis test when there are X = 17 positives.

We see the P-value in this case with X = 17 positives is only 2%. Figure B.3 illustrates the null distribution and highlights the values used to compute the P-value in this case. The interpretation here is that there was only a 2% chance of getting 17 or greater positives if the true prevalence was 10%. Because the P-value is below the significance value of 0.05, we would reject the null hypothesis and conclude the data support the alternative hypothesis that the prevalence is greater than 10%.

B.1.5 Decision errors happen

If you think about it, the sampling distribution tells us that we *should* see the observations outer 5% of the null distribution exactly 5% of the time *when the null hypothesis is true*. Indeed, extreme observations happen just by chance! We're simply concluding that our one sample estimate in those tails is not a chance event, but we could be wrong. How often will we be wrong in this case? Well, exactly 5% of the time, or whatever level we set for the significance value.

This kind of error, where we reject a null hypothesis that is actually true, is (boringly) called a **Type I error**. The idea is simply that if we can imagine the null hypothesis is actually true and we repeat our sampling process thousands of times, we would see extreme observations in the outer 5% of the null distribution exactly 5% of the time.

We can also make the opposite error, namely *failing* to reject a null hypothesis that is false. This is called a **Type II error**, and unlike a Type I error, the probability of a Type II error is not straightforward. In general, the probability of a Type II error is most common under the following circumstances:

- The quantities being estimated are estimated with low precision (i.e., high standard error). Remembering that the standard error is a ratio of the standard deviation (variability among observations) to teh sample size, we know that precision is lowest when there is high variability int he observations and low sample size. In other words, low sample size and high variability in the observations both contribute to a greater likelihood of a Type II error.
- The **effect size** is small. Imagine you are comparing the difference in mean population size of those fish between polluted and unpolluted areas. The magnitude of the difference in the mean population between polluted and unpolluted areas is the effect size. When effect sizes are small, in this case only a small difference in the population sizes between areas, then the probability of a Type II error is greater.
- Low significance value. There's nothign special about a significane value of 0.05. Indeed, one could set it higher or lower, but there's a trade-off. Lowering the significance value will decrese the probability of a Type I error, but at the same time it will increase the probability of a Type II error!

The probability of a Type II error is referred to as β, and the inverse if the probability of a Type II error $(1 - \beta)$ is called the *power* of a statistical test. The general goal when designing a study for a classical hypothesis test is to maximize power, or in other words, minimize the probability of making a Type II error. Researchers can only control some aspects of study design that affect power, namely the sample size (increase power by increasing sample size) and significance value (increase power by increasing the significance vqlue, but of course at greater risk of committing a Type I error).

B.2 Making decisions with confidence intervals

Statistical hypotheses can be evaluated without P-values, namely by using confidence intervals. Recall in the last chapter we estimated the 95% confidence interval for the true proportion infected was 2.7 - 13.3% based on our 8 observed positives. Note that our null hypothesized value of 10% is in the 95% confidence interval, so in other words it is considered a plausible value of the proportion infected based on our sample data. In this case, we would conclude with the confidence interval alone that the data are consistent with the null hypothesis that 10% of individuals are infected.

In some ways the 95% confidence interval gives you more information than a null hypothesis test of a single value. How so? Well, the range of values in the 95% confidence intervals defines the values of the paramter for which you would *not* reject the null hypothesis at a significance value of 5% (specifically when assuming a two-sided alternative hypothesis). In other words, suppose we defined the null hypothesis as $p_{infected} = 0.124$ and the alternative as $p_{infected} \neq 0.124$. Because 12.4% is in the 95% confidence interval, we would not reject the null hypothesis in that case. Any time the null hypothesis is between 2.4% and 13.3%, we wouldn't reject it at the 5% significance levels.

We can interpret confidence intervals in the same way at any degree of confidence. The corollary is that the interval is defining the range of parameter values for which we wouldn't reject the null hypothesis at the significance value being used.

B.3 Issues with the null hypothesis framework

Does your brain hurt after reading to this point? Don't feel bad if it does. Decision theory with frequentist inference is not the most intuitive set of ideas! In this section we take a brief look at some of the flaws and criticisms of classic hypothesis testings.

B.3.1 Significance testing reinforces binary thinking

In my decade plus of teaching null hypothesis testing and emphasizing that scientists and statisticians are not in the business of certainty, I have continued to see students take a box-checking type of approach to making decisions about null hypotheses. "If the P-value is less than 0.05, reject the null hypothesis and accept the alternative hypothesis. If the P-value is greater than or equal to 0.05, accept the null hypothesis." This is classic binary thinking - that the data we collect will produce a P-value that we can use to make a definitive either/or decision about our statistical hypotheses.

I don't blame students of statistics for thinking this way. The competition between two diametrically opposed hypotheses and the predominance of a standard threshold for the significance value (0.05) reinforces the misconception that the results of statistical analyses can be boiled down to "significant" (i.e., P < 0.05, reject the null) or "not significant" (P > 0.05, accept the null). I thought this way myself when I was learning statistics. It took years of practice (and teaching) to really understand these concepts and the nuances of interpretation.

Although it must be true that null hypotheses (or any hypothesis) are either right or wrong, statistical evidence can never get us to the point where we can conclude with certainty that a null hypothesis is right or wrong. We have to think about hypotheses probabilistically in light of the evidence in front of us. Null hypothesis testing gives students of statistics the illusion that we can conduct statistical tests and find "the answer" based on a P-value.

B.3.2 Statistical testing reinforces gamification in science

Why is the significance value typically 0.05? Why not 0.047? Or 0.09? We could clearly trace the history of the significance value set at 0.05 to early frequentist statisticians, but it's just not my goal. There's ultimately no good reason to set the value to 0.05. It's arbitrary.

One of the problems with using an arbitrary threshold - and this is related to the idea that significance testing reinforces binary thinking - is that scientists know what they have to do to find a significant result. The last thing scientists want to see when fitting a statistical model on their computer is the output "P = 0.72". Not significant, not publishable. Sadly that has been the state of science publishing, and it has reinforced bad behavior.

What kind of behaviors? Basically the culture of publishing has incentivized research practices to achieve significant results rather than to achieve trustworthy results. Any scientist who does their own statistical analysis knows that there are often many ways one could analyze the data to shed light on a hypothesis. Some scientists - again, encouraged by the incentives of publishing, which affects promotions, grant seeking, and more - have engaged in what is called **p-hacking**, essentially the practice of analyzing the data multiple ways in hopes that one of those ways produces something like "P = 0.003" on their computer screen. This is thought to explain why the **reproducibility** crisis in some fields, where the findings from many studies that have been published cannot be replicated.

B.3.3 Statistical significance is not the same thing as practical significance

I once reviewed a paper where the researchers were interested in whether the genetic distance between individuals in a wildlife population (basically the inverse of genetic relatedness) was related to the physical distance between them. The data looked like Figure **??**, which shows N = 3000 data points and a line that summarizes the relationship between genetic distance and physical distance. If you fit a model to test the null hypothesis that the slope of that line is exactly 0, you would compute a P-value of P = 0.02, leading one to reject the null hypothesis and conclude that the genetic distance is positively related to physical distance. And that's exactly what the authors did.

Hopefully you can see the problem. Although the finding is "statistically significant", the strength of the relationship between genetic and physical distance is rather unimpressive. This is an important lesson. Statistical significance is not equivalent to practical significance in one's field.

Why do we see cases like this? The problem is that P-values are affected by sample size in frequentist statistical tests. Increasing the sample size reduces the standard error, which increases the precision with which even a small effect size can be detected.

It's important to note that not finding statistical significance does not rule out practical importance. Because P-values are affected by sample size, one might find what appears to be a strong effect even with a P-value above 0.05. Consider our example of fish populations and pollution. Suppose you find a mean difference in fish populations of 100 between polluted and unpolluted water such as a mean difference of N = 40 individuals between polluted and unpolluted waters. Also assume the standard deviation of population size among locations is 50. If these estimates were made with a sample size of N = 50 polluted and unpolluted locations each, the P-value would be 0.0002. However, if the same estimates were made with a sample size of N = 10 polluted and unpolluted locations each, the P-value would be 0.08. Same effect size, but a borderline P-value when the sample size is low. Rather than outright accepting the null hypotheses, a better practice would be to consider the effect size in light of the low sample size.

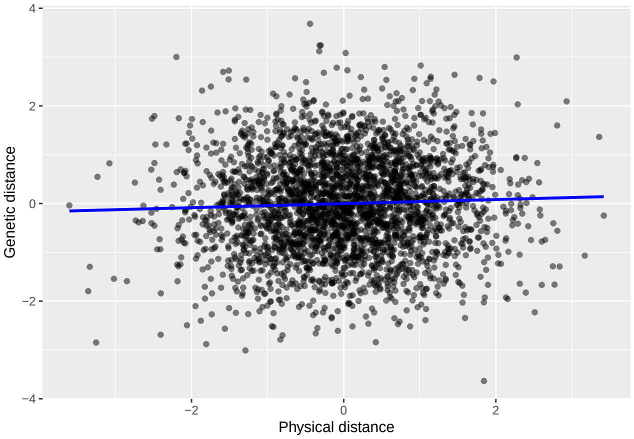


Figure B.4: Simulated relationship of genetic distance to physical distance between individuals in a wildlife population. There are 3000 data points and the P-value for the null hypothesis that the slope of the line relating genetic distance to physical distance is P = 0.02, which would lead to rejecting the null hypothesis.

B.3.4 Type I errors become more likely with multiple tests

Consider again the case of estimating the prevalence of a respiratory illness. Suppose that for each person brought in for a random test, you collect 10 measurements about each individual to investigate potential differences between infected and non-infected individuals. This might include individual's sex, weight, BMI, exposure history, and so on. After the test results come back, you conduct a hypothesis test comparing the infected and non-infected groups for each of those 10 measurements. You use the standard significance value of 0.05 for each test, and so the probability of making a Type I error (rejecting a true null hypothesis) is 0.05 for each test. What's the probability of making at least one Type I error?

The most efficient solution to this question is to quantify the probability of making no Type I errors, and then applying the not rule to find the probability of at least one Type I error. When we conduct 10 tests each with a 0.05 probability of a Type I error, the probability of making no Type I errors is $(1-0.05)^{10}$, and so the probability of making at least one Type I error is $1-(1-0.05)^{10} = 0.40$. Wow - a 40% chance of making at least one Type I errors when we conduct 10 tests. This is the problem of making **multiple comparisons**. For each additional hypothesis test, the probability of making at least one Type I error increases.

The probability of making at least one Type I error across a *family* of tests is sometimes called the **family-wise Type I error rate**, and it can be quantified as $1-(1-\alpha)^N$, where α is the significance level for each test and N is the number of tests. When multiple tests are conducted in a single study, the probability of a spurious result increases with each additional test. One way to combat this is by applying a correction to maintain the family-wise error rate at 0.05. One popular correction is the **Bonferonni correction**, which is applied by using a corrected significance for each test that is based on the number of tests: $0.05/N$. Thus, if we were to conduct 10 tests, we could use a corrected alpha of $0.05/10 = 0.005$ for each test to maintain the family-wise Type I error rate at 0.05.

B.3.5 The null hypothesis is almost certainly wrong

Consider the null hypothesis that the prevalence of a disease is 10%. The claim is that the prevalence is *exactly* $10.\bar{0}\%$. That hypothesis is almost certainly wrong. If the prevalence was 9.9%, or 10.3%, we should technically reject teh null hypothesis. If you're evaluating the effect of some new drug on an outcome, the likelihood that the drug has *exactly* no impact is very low. At best all we can do with frequentist hypothesis tests is say that there's either enough information to reject the null hypothesis, or there is not. But why test a null hypothesis that is almost certainly not true to begin with?

B.3.6 The null hypothesis focuses on data you did not observe

Recall the P-value does not just measure the likelihood of the data you actually observed, assuming the null hypothesis is true. It measures the likelihood of the data you observed, *or data equally or more extreme*, assuming the null hypothesis is true. Why should we be asking about the probability of data we didn't actually observed?

Moreover, because the P-value is quantified based on tail regions of the sampling distribution more extreme than the observed data, scientists can game the system by changing the type of tail region to include. This could be done by conducting a one-tailed hypothesis tests, which will cut the P-value exactly in half, increasing the likelihoood of rejecting the null hypothesis, or by using a different kind of probability distribution for null distribution that has lower probability in the tail.

B.3.7 A single null and alternative hypothesis is too constraining

Recall that our epidemiological goal was to examine whether the prevalence of the disease was greater than 10%. We want to know this because public health measures will be implemented at that level. But how do test whether the prevalence of disease is greater than 10% in a null hypothesis framework? The null hypothesis requires a specific value for the parameter of interest. That's why we set the null hypothesis to exactly 10% rather than looking at a set of possible values. We *were* able to specify a set of possible values for the alternative hypothesis, but remember that in the end, null hypothesis significance testing is a test of the null hypothesis, not the alternative hypothesis.

B.3.8 The P-value is not the probability we want

I'll save the best for last. When we conduct a scientific study, what we really want to know is the probability that a scientific hypothesis is true. We represent those scientific hypotheses by quantitative parameters that we can estimate from data, but with null hypothesis significance testing, we never actually quantify the probability of a hypothesis. P-values are frequently interpreted as the probability that the null hypothesis is true, but that interpretation is wrong. A P-value is the probability of observing the data or more extreme observations *all conditional on the null hypothesis being true.* P-values are conditional probabilities: P(data | hypothesis). What we really want is the reverse conditional probability: P(hypothesis | data), but P-values don't give us that.