Statistical Inference

July, 2023

Table of Contents

# About this Course

We are delighted that you’ve chosen to enroll in Statistical Inference, a part of the Data Science Specialization offered by Johns Hopkins Biostatistics! This course will introduce you to the essential concepts of statistical inference, which will serve as a solid foundation for the rest of your journey in the Data Science track. We strongly believe that the core essence of Data Science lies in its scientific approach.

The primary focus of this Specialization is to provide you with three key elements: Firstly, it will introduce you to the fundamental principles of working with data in a scientific manner, enabling you to derive new and reproducible insights. Secondly, it will equip you with the necessary tools to execute a data analytic strategy, starting from raw data stored in a database and culminating in a comprehensive report with interactive visualizations. Lastly, we emphasize hands-on practice, allowing you to gain practical experience and develop your skills through firsthand application of the techniques taught. This course serves as the fundamental and foundational component of the entire series. While minimizing the reliance on complex mathematics, we aim to provide students with the essential knowledge of utilizing statistics to draw inferences about populations.

We are thrilled about the prospect of expanding Data Science education at scale. Our intention is to create self-contained, fast-paced, and interactive courses that foster an engaging learning experience.

# 1 Probability & Expected Values

In this module, we’ll go over some information and resources to help you get started and succeed in the course. During this week, we’ll focus on the fundamentals including probability, random variables, expectations.

## 1.1 Introduction

Greetings and a warm welcome to the Probability class, which is a part of the Statistical Inference course within the Coursera Data Science series. I’m Brian Caffo, and I will be one of your instructors for this class. Alongside me, we have Jeff Leek and Roger Peng, who will also be co-teaching the course. We all belong to the Department of Biostatistics at the Bloomberg School of Public Health. ### Syllabus The primary instructor of this class is Brian Caffo. Brian is a professor at Johns Hopkins Biostatistics and co-directs the SMART working group.

This class is co-taught by Roger Peng and Jeff Leek. In addition, Sean Kross and Nick Carchedi have been helping greatly.

**Course Content**

In this course we will cover the following topics: 1. Probability 2. Conditional Probability 3. Expectations 4. Variance 5. Common Distributions 6. Asymptotics 7. T confidence intervals 8. Hypothesis testing 9. P-values 10. Power 11. Multiple Testing 12. Resampling

If you’d prefer to watch the videos on YouTube, you can do so through this [link](https://www.youtube.com/playlist?list=PLpl-gQkQivXiBmGyzLrUjzsblmQsLtkzJ).

### 1.1.1 Course Book:

Statistical Inference for Data Science book is now available. It’s a different sort of book published on LeanPub. If you purchase the book on LeanPub, you’ll get all editions in the future for free. You pick the price on the site. You can get it [here](https://leanpub.com/LittleInferenceBook).

Following our style for the specialization, the book is creative commons licensed to offer you maximum flexibility in how you use the materials.

You can also just read a web page rendering of the book [here](https://leanpub.com/LittleInferenceBook/read)

In addition, the book is available on [GitHub](https://github.com/bcaffo/LittleInferenceBook) if you wish to render it yourself using pandoc.

### 1.1.2 Github repository

The most up to date information on the course lecture notes will always be on the [Github repository](https://github.com/DataScienceSpecialization/courses/tree/master/06_StatisticalInference).

Please issue pull requests so that we may improve the materials. Notice that Brian’s forked github repo is sometimes out of sync with the Data Science Specialization repo managed by the other instructors. Make sure to check in Brian’s master repo for the most up to date material.

If you would just like the full set of lecture pdfs, grab them [here](https://github.com/bcaffo/courses/blob/master/06_StatisticalInference/lectures.zip?raw=true).

If you would just like the full set of Rmd files for the lecture code, get those [here](https://github.com/bcaffo/courses/blob/master/06_StatisticalInference/rmd.zip?raw=true).

### 1.1.3 Homework Problems

In the book, there are homework problems fairly similar to the quiz questions.

If you can do them, you should be in very good shape for the quizzes. The homework assignments in this course are optional. They won’t count toward your final grade, but they are a good opportunity to practice the skills covered in the course. There are worked out solutions on youtube linked to the book. These are ordered in an odd way, as the class has been restructured. So, it’s probably best to just do them through the book.

[Homework 1](http://bcaffo.github.io/courses/06_StatisticalInference/homework/hw1.html#1) [Homework 2](http://bcaffo.github.io/courses/06_StatisticalInference/homework/hw2.html#1) [Homework 3](http://bcaffo.github.io/courses/06_StatisticalInference/homework/hw3.html#1) [Homework 4](http://bcaffo.github.io/courses/06_StatisticalInference/homework/hw4.html#1)

### 1.1.4 Differences of opinion

Keep in mind that currently data analysis is as much art as it is science - so we may have a difference of opinion - and that is ok! Please refrain from angry, sarcastic, or abusive comments on the message boards. Our goal is to create a supportive community that helps the learning of all students, from the most advanced to those who are just seeing this material for the first time.

### 1.1.5 Data Science Specialization Community Site

Since the beginning of the Data Science Specialization, we’ve noticed the unbelievable passion students have about our courses and the generosity they show toward each other on the course forums. A couple students have created quality content around the subjects we discuss, and many of these materials are so good we feel that they should be shared with all of our students.

We’re excited to announce that we’ve created a site using [GitHub Pages](http://datasciencespecialization.github.io/) to serve as a directory for content that the community has created. If you’ve created materials relating to any of the courses in the Data Science Specialization, please send us a pull request so we can add a link to your content on our site. You can find out more about contributing [here](https://github.com/DataScienceSpecialization/DataScienceSpecialization.github.io#contributing)

We can’t wait to see what you’ve created and where the community can take this site!

## 1.2 Probability

In today’s lecture, we will cover the fundamentals of probability at a beginner’s level, providing you with the necessary knowledge for your journey in the data science specialization. If you’re interested in delving deeper into this topic, I highly recommend checking out my comprehensive mathematical biostatistics boot camp series. In addition,the course notes are on GitHub.

In this module we discuss probability, the foundation of statistical analysis. Probability assigns a number between 0 and 1 to events to give a sense of the “chance” of the event. Probability has become our default model for apparently random phenomena. Our eventual goal is to use probability models, our formal mechanism for connecting our data to a population. However, before we get to probability models, we need to understand the basics of probability calculus. The next few lectures cover these basics.

**Probability** = the study of quantifying the likelihood of particular events occurring - given a random experiment, probability = population quantity that summarizes the randomness This summary is not just about the data at hand, but a conceptual quantity that exist in the population that we want to estimate.

Let’s delve into the concept of probability. In the context of a random experiment, such as rolling a die, probability quantifies the inherent randomness of the outcomes. It’s important to highlight the term “population” here. When considering a die roll, probability is seen as an intrinsic characteristic of the die itself, rather than being dependent on a specific sequence of fixed rolls. Therefore, when we discuss probability, we’re referring to a conceptual property that exists within the population we aim to estimate, rather than being directly observable in the data we have. Now, let’s define the principles that govern probability, known as probability calculus. Firstly, probability operates on the potential outcomes of an experiment. For instance, when rolling a die, the possible outcomes could be 1, the set {1, 2}, the set of even numbers {2, 4, 6}, or the set of odd numbers {1, 3, 5}, and so on. Probability is a function that assigns a number between 0 and 1 to each of these sets of possible outcomes. We must adhere to the rule that the probability of an event occurring, such as rolling the die and obtaining a particular number, must be equal to one. Additionally, the probability of the union of two mutually exclusive sets of outcomes must be equal to the sum of their individual probabilities. For example, consider the scenario where one possible outcome is obtaining either a one or a two, while another possible outcome is obtaining either a three or a four. These two sets, {1, 2} and {3, 4}, cannot occur simultaneously. The probability of the union, i.e., obtaining a one, two, three, or four, is the sum of the probabilities of obtaining a one or two, plus the sum of the probabilities of obtaining a three or four.

Interestingly, these simple rules encompass all the necessary principles to establish the general rules that govern probability. This significant discovery was made by the Russian mathematician Kolmogorov. Let’s explore some of the essential rules that probability must abide by. While I have already mentioned a few, others naturally follow from the previously stated rules.

1. The probability of an event not occurring, or “nothing” happening, is zero. In the case of rolling a die, something is bound to occur, and you will obtain a number.
2. Conversely, the probability of an event occurring, such as rolling a specific number on the die, is equal to one.
3. It is intuitive to understand that the probability of an event happening is equal to one minus the probability of the opposite event occurring. For example, the probability of rolling an even number on a die is equal to one minus the probability of rolling an odd number. This is because the set of odd numbers is considered the opposite of obtaining an even number in the context of rolling a die.
4. The probability of at least one of two or more mutually exclusive events, which cannot occur simultaneously, is the sum of their individual probabilities. This aligns with the definition we discussed earlier.
5. Another consequence of probability calculus is that if event A implies the occurrence of event B, then the probability of event A is less than or equal to the probability of event B. Although this may sound complex when explained verbally, it becomes clearer when visualized using a Venn diagram.



Figure : Figure 1.1: Event A being sub section of B event

In the diagram, event A is represented by a circle contained within event B. When we consider the probability of A, we assign a number to the area within circle A. Similarly, when discussing event B, we refer to the probability assigned to the entire circle, which includes the area of A. Therefore, it logically follows that the probability of B is larger than or equal to the probability of A. This concept is often intuitive and easily understood once visualized. For instance, the probability of rolling a 1 (set A) is less than the probability of rolling a 1 or a 2 (set B).

1. For any two events, the probability of at least one occurring is equal to the sum of their probabilities minus the probability of their intersection.



Figure : Figure 1.2: Events A and B with intersection

Again, visualizing this with a Venn diagram helps in understanding it better. Consider set A and set B. When we add their individual probabilities, we are effectively adding the intersection region twice, once when considering A and once when considering B. Since we have counted the intersection twice, to obtain the probability of their union, we need to subtract the intersection once. This rule highlights that we cannot simply add probabilities if there exists a non-trivial intersection between the events.

Now, let’s illustrate an example to demonstrate why we cannot simply add probabilities when the events are not mutually exclusive. According to the National Sleep Foundation, approximately 3% of the American population has sleep apnea, while around 10% of the North American and European population has restless leg syndrome. Let’s assume, for the sake of argument, that these probabilities are derived from the same population. The question is, can we add these probabilities together to conclude that about 13% of people in this population have at least one of these sleep problems? The answer is no. The reason is that these events, sleep apnea and restless leg syndrome, can occur simultaneously and are not mutually exclusive. There is a non-trivial portion of the population that experiences both conditions concurrently.

To elaborate further, let’s define event A as the occurrence of sleep apnea in a person drawn from this population, and event B as the occurrence of restless leg syndrome. In this case, we believe that the intersection of these two events (the occurrence of both conditions) is non-trivial. If we were to naively add the probabilities of A and B, we would essentially count the intersection twice, which would result in an overestimate. To determine the probability of the union (at least one of the conditions), we need to subtract the intersection once, recognizing that it was mistakenly included twice in the initial addition.

### 1.2.1 Probability mass functions and probability density functions

Probability calculus provides a valuable framework for understanding the fundamental rules that govern probability and serves as the basis for all probabilistic thinking. However, when it comes to numeric outcomes of experiments, we require a more practical approach. This is where densities and mass functions for random variables come into play, serving as a convenient starting point. These concepts will be sufficient for our purposes, which is collecting data that will be utilized to estimate properties of the population.

One of the most well-known examples of a density function is the bell curve, also known as the normal distribution. In this class, you will gain a deeper understanding of what it truly means for data to follow a bell curve. You will learn about the significance and interpretation of the bell curve. Importantly, you will also realize that when discussing probabilities associated with the bell curve or the normal distribution, we are referring to population quantities, not statements solely based on the observed data.

Before delving into data analysis, it is crucial to develop our intuition for understanding population quantities. A random variable represents the numerical outcome of an experiment. In our study, we will encounter two types of random variables: discrete and continuous. Discrete random variables are those that can be counted, such as the number of web hits or the possible outcomes of rolling a die. They can even include non-numeric attributes like hair color, which can be assigned numeric values (e.g., 1 for blonde, 2 for brown, 3 for black, etc.). For discrete random variables, we assign probabilities to each possible value they can take. On the other hand, continuous random variables can assume any value within a range or continuum. When working with continuous random variables, we assign probabilities to ranges of values they can take.

Let’s consider some simple examples that can be viewed as random variables, as these examples will aid in building our intuition throughout the course. One prominent example is the flip of a coin, where we can assign values of “heads” or “tails” (or 0 and 1) to represent the outcomes. This is a discrete random variable since it can only take two distinct levels. Another example of a discrete random variable is the outcome of rolling a die. It can only take one of six possible values, making it a discrete random variable with simple probability mechanics. A more complex random variable would be, the amount of website traffic or the number of web hits on a given day. While we’ll likely treat it as discrete, it’s interesting because it doesn’t have an upper bound. In such cases, we might employ the Poisson distribution to model it. The hypertension status of a randomly selected subject from a population can also be a random variable. We may assign a value of 1 to indicate the presence of hypertension or a diagnosis, and 0 otherwise. This random variable would typically be modeled as discrete.

An example of continuous random variable would be measuring a subject’s body mass index (BMI). In this case, BMI would be considered a continuous random variable, as it can assume any value within a range. Intelligence quotients (IQ) are often modeled as continuous random variables.

When working with discrete random variables, we assign a probability to each possible value they can take. We represent this assignment using a function called the probability mass function (PMF). The PMF takes any value of the discrete random variable and assigns the probability of it taking that specific value.

For example, in the case of a die roll, the PMF would assign a probability of one-sixth to the value one, one-sixth to the value two, one-sixth to the value three, and so on.

To ensure that the PMF satisfies the basic rules of probability, we have two requirements. First, the PMF must always be greater than or equal to zero since probabilities range from zero to one, inclusive. Second, the sum of the probabilities assigned to all possible values of the random variable must add up to one. In the case of a die roll, if we add the probabilities of getting one, two, three, four, five, and six, the sum should equal one. This ensures that the probability of any possible outcome occurring is accounted for.

Therefore, the PMF of a discrete random variable must adhere to these two rules to accurately represent probabilities.

We will primarily focus on using probability mass functions (PMFs) that are particularly useful in our context. Two examples of such PMFs are the binomial distribution, commonly used for coin flips, and the Poisson distribution, commonly used for counting events. However, let’s discuss one of the most well-known PMFs, the Bernoulli distribution, which is often used to model the outcome of a coin flip.

Let’s denote the random variable representing the coin flip outcome as capital X, where X = 0 represents tails and X = 1 represents heads. In this notation, an uppercase letter represents a potential value of the random variable that may or may not occur. On the other hand, a lowercase x serves as a placeholder for a specific value that we will substitute.

The PMF for the Bernoulli distribution is represented as . When we substitute x = 0 into this PMF, we obtain a probability of one-half. Similarly, when we substitute x = 1, we also get a probability of one-half. This means that the probability of the random variable X taking the value 0 is one-half, and the probability of it taking the value 1 is also one-half.

When we introduce an unfair coin, we can adjust our approach by considering a parameter, theta, representing the probability of getting a head. The probability of getting a tail would then be 1 minus theta, where theta is a number between 0 and 1. In this case, the probability mass function can be written as follows:

. By substituting x = 1 into this PMF, we obtain the probability . Similarly, when we substitute x = 0, we get the probability . This implies that for this population distribution, the probability of the random variable X taking the value 0 is , and the probability of it taking the value 1 is .

This approach is particularly useful for modeling the prevalence of a certain condition or event. For instance, if we want to model the prevalence of hypertension, we can assume that the population or sample we are studying can be likened to the outcomes of biased coin flips with a success probability represented by . However, the challenge lies in not knowing the exact value of . Therefore, we will utilize our data to estimate this proportion within the population.

In contrast to the probability mass function, which assigns probabilities to specific values for discrete random variables, the probability density function (PDF) is associated with continuous random variables. Similar to the rules that the probability mass function follows, a valid probability density function must satisfy two specific rules: it must be greater than or equal to zero everywhere, and the total area under the function must be equal to one. The key concept of a probability density function is that areas under the curve correspond to probabilities for the random variable. For instance, if we state that intelligence quotients (IQ) are normally distributed with a mean of 100 and a standard deviation of 15, we are implying that the population follows a bell-shaped curve. In this case, the probability that a randomly selected individual from that population has an IQ between 100 and 115 is represented by the area under the curve within that range. It is important to note that the probability density function represents a statement about the population of IQs and not the data itself. The data will be used to assess and evaluate the assumptions made about the population’s probability distribution. It is worth emphasizing that whenever the term “probability” is used, it refers to a population quantity.



Figure : Figure 1.3: Area between 100-115 IQ under normal distribution

It is interesting to note that when we model continuous probabilities using probability density functions (PDFs) for continuous random variables, the probability of the variable taking any specific value is actually *zero*. This is due to the fact that the area under a line, which represents a single point, is zero. However, this does not pose a problem and is simply a quirk arising from modeling random variables with infinite precision. It does not affect the functioning of probability calculations.

The bell-shaped curve, which represents a normal distribution, can be quite challenging to work with until you learn the appropriate techniques, which will be covered in a separate lecture. For now, let’s consider a simpler density function that resembles a right triangle. We’ll use the function for x between 0 and 1, and 0 otherwise, as an example. Let’s provide some context for this function: imagine it represents the proportion of help calls that are addressed in a random day by a helpline.



Figure : Figure 1.4: Shape of the density function for f(x)=2x

What does this density function imply? It means that the probability of the number of calls being addressed falling between 20% and 60% of the total calls for that day is given by the area under the curve in that range. Now, let’s evaluate whether this function is a mathematically valid probability density function.

Looking at the plot of the PDF, which resembles a right triangle, we can see that it is always greater than or equal to zero. Next, let’s calculate the area under the curve. Since it is a right triangle, the area is equal to half the base (which is 1) multiplied by the height (which is 2). Thus, the area is 1. Therefore, this function satisfies the requirements of a valid probability density function, as it is always non-negative and the total area under the curve is equal to 1.

Example: we want to find the probability that 75% or fewer calls get addressed in a randomly sampled day from this population. At the point (0.75, 1.5) on the density function, the height is 1.5 because the function is defined as 2 times x. The base value is 0.75. To calculate the probability, we divide the area, which is half the base times the height, by 2. So the probability turns out to be 56%, as shown in the example.



Figure : Figure 1.5: Shape of the density function for f(x)=2x

Interestingly, this density function is a special case of a well-known distribution called the distribution. I have provided the R code here for obtaining the probability directly from the distribution. Although in this simple case we don’t need it because we are working with triangles, in more complex scenarios, we will require these functions. It’s worth mentioning that in R language we can right this as pbeta(0.75,2,1) the p prefix before a function denotes the calculation of probabilities, 1 define the specific triangle we are using in this example, and you can test and see that it yields the same result of 56%. Certain areas of the density are so commonly used that they are given specific names. For instance, the cumulative distribution function (CDF) of a random variable X gives the probability that X is less than or equal to a given value x.

This definition holds for both discrete and continuous random variables. In the case of the beta distribution we just examined, the pbeta function in R always returns the probability of being less than or equal to the first argument provided.

Alternatively, the survival function is another useful concept. It is defined as 1 minus the cumulative distribution function and represents the probability of a random variable being greater than a given value.

Suppose we wanted to determine the cumulative distribution function for the previously mentioned density. For instance, we might want to find the probability that 40% or fewer, 50% or fewer, or 60% or fewer of the calls get answered in a given day based on this specific right triangle population density function. In each case, the calculation will resemble what we did earlier for 0.75. Since the density function is a right triangle, the probability is half the area of the base times the height. This simplifies to one-half times x times 2x, which equals . Therefore, the function provides the probability of that percentage or fewer calls being answered on a randomly sampled day.

To examine the results when we use the pbeta function, which corresponds to the cumulative distribution function in R, for the three values mentioned earlier, we can write the followings.

pbeta(c(0.4,0.5,0.6),2,1) where parameters 2 and 1 are utilized to evaluate the specific density, yielding probabilities of 16%, 25%, and 36%. Therefore, the probability that 40% or fewer of the calls get answered on a given day is 16%, the probability that 50% or fewer get answered is 25%, and the probability that 60% or fewer get answered is 36%. In terms of the survival function, it is simply 1 minus the cumulative distribution function, which can be expressed as 1 minus .

As we progress, we will encounter more complex density functions. However, the process will be simpler since we can rely on existing functions such as pnorm and pbeta instead of calculating them directly.

### 1.2.2 Quantiles

You’re already familiar with sample quantiles, such as the 95th percentile, which represents the 0.95 quantile of a dataset. If you score at the 95th percentile on an exam, it means that 95% of the students scored worse than you while 5% scored better. Now, let’s introduce the concept of population analogs for quantiles. In the case of the 95th percentile or the 0.95 quantile, you would order the observations from least to greatest and locate the point or exam score below which 95% of the observations lie. This point is denoted as , where alpha corresponds to the quantile. In other words, it satisfies the condition , where F is the distribution function. To better understand this concept, let’s try to visualize it.

Let’s consider the distribution function , which represents the area below point x on a density plot. This area corresponds to the probability that a random variable from the population is less than or equal to x. As an example let’s imagine a population of test scores, an infinite population of students. The distribution function gives us the probability of obtaining a score equal to or lower than x for a randomly selected student from this population.

Now, let’s introduce the concept of the quantile. We move a line along the distribution until we find the point , where exactly proportion of the probability lies below it. This is similar to what we do with our data when finding an empirical quantile, where we locate the data point such that, for example, 95% of the test scores lie below it, which corresponds to the sample percentile. In the population distribution, we move the x point until we find the point where the probability of being below it is 95%. Percentiles are essentially quantiles with alpha expressed as a percentage rather than a proportion. The median, often the most well-known quantile, represents the 50th percentile.



Figure : Figure 1.6: 95 percentile of a distribution

Quantiles are frequently used, particularly with the normal distribution. However, we rarely need to directly work with densities to calculate quantiles, as the distributions we commonly encounter have well-defined quantiles. In R, we can easily find quantiles using the q prefix before the density function name. For example, for the density we discussed earlier, the function qbeta gives us the relevant quantile. We can input 0.5 (qbeta(0.5,2,1))to find the median, considering that R expects the quantile argument as a proportion rather than a percentage, i.e. 0.5 is acceptable and 50 for 50% is not acceptable. The parameters 2 and 1 are specific to the density we’re working with, which you’ll have to trust me on for now. When we calculate the quantile using qbeta with 0.5 as the argument, we obtain the same result as before, 0.7 or 0.71.

At this point, you might be wondering why the concept of the median seemed simpler before, when you would order observations and select the middle value or averaging the two middle values for an even number of observations.

The answer is there you had an estimator. However, in this class, we aim to go beyond just estimators and focus on the targets of estimation, known as estimands. In the case of the sample median, it estimates the population median.

To understand this, let’s consider an example where we sample a few days and calculate the percentage of calls answered on those days. If we line up these percentages in ascending order, the middle value represents the *sample median*. We can think of this sample median as an estimator for the true median percentage of calls answered in the population. However, to establish a connection between the sample and the population, we need to make certain assumptions, which we will thoroughly explore and formalize in this class.

In essence, for every estimator, there exists an estimand in this class. The sample mean estimates the population mean, the sample median estimates the population median, and the sample standard deviation estimates the population standard deviation, and so on. This process is known as *statistical inference*, where we link our sample data to the underlying population.

## 1.3 Conditional probability

Conditional probability is a very intuitive idea, “What is the probability given partial information about what has occurred?”. To illustrate the concept of conditioning, let’s consider [XKCD comic](https://xkcd.com/795/). The comic portrays two individuals standing in a field during a lightning storm near a tree. One person suggests going inside, but the other dismisses the idea, citing the low chance of getting struck by lightning, approximately one in seven million. However, the comic humorously points out that the death rate among people who know this statistic is one in six. The underlying message is that the second person has failed to consider the additional information available to them, leading to an incorrect assessment of risk.

Consider another example to better understand conditional probabilities. Suppose we have a standard die, and the probability of rolling a one is assumed to be . However, if we are given the extra information that the roll resulted in an odd number (one, three, or five), our perspective changes. Now, conditioned on this new information, we would no longer say that the probability of rolling a one is . Instead, we would consider the one, three, and five to be equally likely outcomes, so the probability of rolling a one becomes . This demonstrates how conditional probabilities adjust our understanding based on additional information.

To define conditional probability, suppose we have an event B with a nonzero probability. Then, the conditional probability of event A given that B has occurred is denoted as and is defined as:

In the case where A and B are statistically independent events (*we will define this later*), the conditional probability simplifies to the probability of A. This means that if the occurrence of event B provides no new information about event A, the probability of A remains unchanged.

Let’s verify that the concept of conditional probability aligns with our intuition in the example of rolling a die. Event B represents the occurrence of an odd number (one, three, or five), and event A represents rolling a one. We want to find the . In other words, we are interested in the probability of rolling a one when we know that the outcome is an odd number. Using the definition of conditional probability,

. Since A is entirely contained within B, the probability of A ∩ B is simply the probability of A, which is . The probability of B, in this case, is that is for each of the three mutually exclusive possibilities. Thus, the conditional probability equals divided by , which simplifies to , confirming our previous understanding.

Conditional probability allows us to update our probabilities based on new information, and it plays a **crucial** role in statistical inference. ### Bayes’ rule One of the well-known applications of conditional probability is Bayes’ rule, named after Thomas Bayes, a Presbyterian minister whose work was published posthumously. Bayes’ rule allows us to reverse the conditioning set and the set we are interested in finding the probability of. Suppose we want to calculate the probability of event B given event A, and we already know or can easily calculate the probability of event A given event B. Bayes’ rule enables us to evaluate the probability of B given A in terms of the probability of A given B.

However, to apply Bayes’ rule, we also need the marginal probability of event B, which is valuable in various contexts such as diagnostic tests.

Conditional probability in the context of a diagnostic test, exemplifies one of the significant applications of conditional probability and Bayes’ rule. Consider a test for a disease, where we define plus(+) and minus(-) as events representing a positive or negative test result, respectively. D and represent the events of having or not having the disease, respectively. The sensitivity of the test is the probability that the test is positive given that the subject actually has the disease. A high sensitivity indicates a good test. The specificity, on the other hand, is the probability that the test is negative given that the subject does not have the disease. A high specificity is desirable for a good test. While obtaining accurate estimates of sensitivity and specificity can be challenging, in certain cases, like an HIV blood test, it is possible to test individuals known to have or not have the disease to estimate these probabilities.

When a diagnostic test is positive, the probability of having the disease given the positive test result (positive predictive value) is of particular interest. Similarly, when the test is negative, the probability of not having the disease given the negative test result (negative predictive value) becomes relevant. In the absence of a test, the probability of having the disease is known as the prevalence of the disease.

Here is an example to illustrate the calculation of positive predictive value using Bayes’ rule. Suppose a study comparing the efficacy of an HIV test reports sensitivity as 99.7% and specificity as 98.5%. These numbers are for illustrative purposes and do not reflect actual HIV test statistics. Now, consider a subject from a population with a 0.1% prevalence of HIV who receives a positive test result. We want to calculate the associated positive predictive value.

Applying Bayes’ rule, we have the probability of disease given a positive test result equal to the probability of a positive test result given disease multiplied by the probability of disease , divided by the denominator. To simplify, we express the probability of a positive test result given no disease as and the probability of no disease as . Substituting known values, we find the positive predictive value to be 6% for this test in the given population. The low positive predictive value is primarily due to the low prevalence of the disease. However, in a counseling scenario, if the counselor discovers that the subject is an intravenous drug user who regularly has intercourse with an HIV-infected partner, the counselor would consider a much higher prevalence for this particular individual, leading to a higher positive predictive value.

Bayes’ rule provides a powerful framework for incorporating new information and adjusting probabilities based on conditional events, making it valuable in various fields, including diagnostics and decision-making. We want to distinguish between two components: the component that is dependent on the prevalence and the component that is objective evidence of the positive test result. This is where diagnostic likelihood ratios(DLR) come into play, and we’ll explore them further. First, let’s revisit the formula for positive predictive value in Bayes’ rule, which depends on sensitivity, specificity, and disease prevalence.

We can apply a similar approach to calculate the probability of not having the disease given a positive test result.

By dividing these two equations, we arrive at the odds of disease given a positive test result divided by the odds of not having the disease given a positive test result.

post test odds of = \* pre test odds of D Dividing a probability by 1 minus that probability gives us the **odds**. Therefore, on the left side, we have the odds of disease given a positive test result, while on the right side, we have the odds of disease without the test result. The factor in the middle represents the diagnostic likelihood ratio for a positive test result. The equation can be expressed as follows: the pretest odds of disease multiplied by the diagnostic likelihood ratio equals the post-test odds of disease. In other words, the diagnostic likelihood ratio of a positive test result indicates how much the odds change when multiplied by it, transitioning from pretest to post-test odds.

Returning to our example, assume a subject has a positive HIV test. Using the sensitivity and specificity values mentioned earlier, the diagnostic likelihood ratio is calculated as divided by , resulting in . Regardless of the pretest odds, multiplying them by 66 gives the post-test odds. Thus, the hypothesis of disease is 66 times more supported by the data compared to the hypothesis of no disease. Even if the pretest odds are initially small, multiplying them by 66 will still yield a larger but still small number.

Now, consider the scenario when a subject receives a negative test result using the . In this case, the , derived from the sensitivity and specificity values mentioned earlier, is 0.003.

Consequently, the post-test odds of disease in light of a negative test result become 0.3% of the pretest odds of disease. Stated differently, the hypothesis of disease is supported 0.003 times the hypothesis of no disease given the negative test result. By incorporating diagnostic likelihood ratios, we can assess the impact of a test result on the odds of disease and gain insights into the strength of evidence provided by the test.

### 1.3.1 Independence

As mentioned earlier, event A is considered independent of event B if the probability of A given B is equal to the probability of A, given that event B has a positive probability. Another definition of independence states that events A and B are independent if the probability of their intersection equals the product of their individual probabilities. This leads us to an important lesson: we cannot simply multiply probabilities without considering the independence of the events involved. Multiplication of probabilities is valid only for independent events.

Example: What is the probability of getting two consecutive heads when flipping a fair coin? We define event A as the probability of getting a head on the first flip and event B as the probability of getting a head on the second flip. Both probabilities are 0.5 since we assume a fair coin. In this case, because the events are independent, the probability of (getting heads on both flips) is the product of their probabilities, which is 0.25. This calculation is straightforward and correct. However, problems arise when people multiply probabilities in situations where they shouldn’t. A notable example of incorrectly multiplying probabilities was reported in volume 309 of Science. It involved a physician who gave expert testimony in a criminal trial. The trial concerned a mother whose two children had died from sudden infant death syndrome (SIDS). The expert testimony multiplied the prevalence of SIDS (1 out of 8,500) by itself to calculate the probability of two children from the same mother having SIDS. Based on this evidence, among other factors, the mother was convicted of murder. The fundamental mistake in this case was multiplying probabilities for events that were not necessarily independent. It is reasonable to assume that events within families, such as the occurrence of SIDS, are dependent due to genetic or familial environmental factors.

In our class, we will primarily use the concept of independence by assuming that a collection of random variables are independent and identically distributed (IID). This means that the random variables are independent from each other and follow the same probability distribution. For example, several coin flips can be considered IID because each flip is independent of the others, and they all follow the same distribution with a 0.5 probability for heads and 0.5 for tails. IID sampling serves as our default model for a random sample. Even if we do not have an actual random sample, we often use the conceptual model of random sampling or IID to analyze our data. It will be the principal mode of analysis in this class.

## 1.4 Expected values

The empirical average is a very intuitive idea; it’s the middle of our data in a sense. But, what is it estimating? We can formally define the middle of a population distribution. This is the expected value. Expected values are very useful for characterizing populations and usually represent the first thing that we’re interested in estimating.

Now, we will discuss the process of drawing conclusions about populations based on noisy data obtained from them. We will assume that the populations and the randomness governing our samples are described by probability density functions and probability mass functions. Instead of focusing on the entire function, we will examine characteristics of these distributions that are reflected in the random variables drawn from them. The most valuable such characteristics are expected values, particularly the mean. The mean represents the center of a distribution. As the mean shifts, the distribution moves either to the left or right.



Figure : Figure 1.7: Mean of a distribution

Another important characteristic is variance, which measures the spread of a distribution.  Similar to how sample quantiles estimate population quantiles, sample expected values estimate population expected values. Therefore, the sample mean serves as an estimate of the population mean, the sample variance estimates the population variance, and the sample standard deviation approximates the population standard deviation.

The expected value, or mean, of a random variable represents the center of its distribution. For a discrete random variable x with a probability mass function , the expected value is calculated by summing the possible values that x can take multiplied by their respective probabilities.

Conceptually, the expected value draws inspiration from the idea of the physical center of mass, where the probabilities act as weights and x represents the location along an axis. To illustrate this notion of center of mass, consider the sample mean. Even though we are focusing on the population mean in this discussion, it is interesting to note that the sample mean can be seen as the center of mass if we treat each data point as equally likely. In other words, each data point is assigned a probability of , where N is the sample size. Intuitively, we employ this center of mass idea when using the sample mean.

To demonstrate this concept, I have provided some code that calculates the sample mean of a dataset and depicts it as the center of mass by generating a histogram. The example employs a dataset from R called “Galton,” which consists of paired data representing the heights of parents and their children.

library(reshape2)  
library(UsingR)  
library(ggplot2)  
data(galton)  
longGalton <- melt(galton, measure.vars = c("child", "parent"))  
g <- ggplot(longGalton, aes(x = value)) + geom\_histogram(aes(y = ..density.., fill = variable), binwidth=1, colour = "black") + geom\_density(size = 2)  
g <- g + facet\_grid(. ~ variable)  
g

The histogram displays the child’s height distribution, and a continuous density estimate is superimposed.



Figure : Figure 1.9: Height distribution for Childran and Parents

To further explore this concept, we can use the “manipulate” function available in RStudio. By manipulating the mean value, we can observe how it balances out the histogram.

library(manipulate)  
library(UsingR)  
library(ggplot2)  
data(galton)  
  
myHist <- function(mu){  
 g <- ggplot(galton, aes(x = child))  
 g <- g + geom\_histogram(fill = "salmon",   
 binwidth=1, aes(y = ..density..), colour = "black")  
 g <- g + geom\_density(size = 2)  
 g <- g + geom\_vline(xintercept = mu, size = 2)  
 mse <- round(mean((galton$child - mu)^2), 3)   
 g <- g + labs(title = paste('mu = ', mu, ' MSE = ', mse))  
 g  
}  
manipulate(myHist(mu), mu = slider(62, 74, step = 0.5))

You can use the slider to move the mean value and observe how it affects the mean squared error. 

The mean squared error is a measure of imbalance, indicating how stable or unsteady the histogram appears. As we move the mean closer to the center of the distribution, the mean value increases, while the mean squared error decreases, signifying a better balance. However, if we move the mean too far from the center, the mean squared error increases again, indicating increased imbalance. This demonstration illustrates that the empirical mean serves as the balancing point for the empirical distribution, and we will utilize this concept when discussing the population mean, which serves as the balancing point for the population distribution.

Example: Suppose we flip a fair coin, and we assign the value 0 to tails and the value 1 to heads. What is the expected value of X? Again, the expected value represents a property of the population. By plugging the values into our formula, we calculate the expected value of X as follows:

When we compute this expression, we find that the expected value of X is 0.5. It’s interesting to note that the expected value is a value that the coin itself cannot actually take. However, from a geometric perspective, the answer becomes quite obvious. If we visualize the coin’s values as two bars of equal height, one at 0 and the other at 1, we can easily determine the balancing point by placing our finger exactly at 0.5.



Figure : Figure 1.11: Expected value of a coin flip

Example: A random variable X represents the outcome of a biased coin flip. The probability of obtaining heads is denoted as , while the probability of obtaining tails is . What is the expected value of X in this case? By directly applying the formula, we multiply the value 0 by the probability and add it to the value 1 multiplied by the probability . The result simplifies to . Therefore, the expected value of a coin flip, even when the coin is biased, corresponds to the true long-run proportion of obtaining heads in an infinite number of coin flips.

Example: Suppose we roll a fair six-sided die, and X represents the number that appears face up. What is the expected value of X? Here, we take the values 1, 2, 3, 4, 5, and 6 and multiply each by the corresponding probability of the random variable X taking those values (each value has a probability of ). When we perform this calculation, we find that the expected value of X is 3.5. Once again, this is a value that the die itself cannot actually show.



Figure : Figure 1.12: Expected value of a die roll

Similar to the coin example, the geometric argument makes it evident. We have six bars, each with a height of , representing the possible outcomes of the die. If we were to balance them, it becomes clear that the balancing point would be at 3.5. ### Expected values for PDFs When dealing with continuous random variables, it can be helpful to imagine cutting out the shape of probability density on a piece of wood and determining where you would place your finger to balance it out. This concept aligns with the notion of the center of mass of a continuous body. In the case of probability mass functions, as the bars representing the probabilities become narrower and smaller, we can visualize their balancing point.

Example: Suppose we have a density that ranges from zero to one, and the question arises: Is this a valid density? The answer is yes; it corresponds to a well-known density called the **Uniform density**. Now, what is its expected value? If we were to cut this density out of a piece of wood and balance it, the position where we would place our finger to achieve balance is precisely at 0.5. This aligns perfectly with the expected value of the uniform density.



Figure : Figure 1.13: Uniform density

It’s crucial to understand that expected values represent properties of the distribution. They serve as the center of mass of a distribution. Additionally, it’s important to note that the average of random variables is, in itself, a random variable. For example, if we roll six dice and calculate their average, the resulting value is a random variable. By repeatedly sampling from this average through multiple dice rolls, we generate a distribution that also possesses an expected value. The center of mass of this distribution coincides with the center of mass of the original distribution.

This topic becomes highly relevant to the field of inference, so let’s explore some simulation examples to gain a better understanding.



Figure : Figure 1.14: Simulation example

In the first example, the blue density represents the outcome of numerous simulations based on a standard normal distribution. Due to the large number of simulations, this density provides a reliable approximation of the true distribution. It shows that collecting ample data from a population allows us to approximate its originating distribution effectively. The center of mass of this distribution, which would achieve balance, is located at zero.

Now, let’s shift our focus to simulating the average of ten standard normals. By repeatedly performing this process and plotting the resulting histogram or density estimate, we obtain a different distribution. It no longer represents the distribution of standard normals; rather, it illustrates the distribution of averages of ten standard normals. This new distribution, represented by the salmon-colored plot, exhibits interesting properties. Notably, it is concentrated around zero, and this aligns with our previous point. The distribution of averages from a population tends to be centered at the same location as the distribution of the original population itself.

Although calculations and simulations can help us grasp these concepts conceptually, we can observe this phenomenon without explicitly performing them. Imagine rolling a die thousands of times and plotting a histogram of the results. In this case, approximately of the rolls would occur for each number from one to six. As we increase the number of rolls, these bars would eventually balance out. The center of mass for this distribution, which would achieve balance, is 3.5 (not exactly, given the finite number of rolls, but in theory, it would converge to 3.5 with an infinite number of rolls).



Figure : Figure 1.15: Die roll simulation

Consider the scenario where we roll the die twice and calculate the average of the numbers obtained. If we repeat this process multiple times and create a distribution of these averages, we see a different pattern in the second panel. It appears more Gaussian in shape (we’ll discuss this further later), and importantly, it is centered at the same location as before.



Figure : Figure 1.16: Coin toss average

The population mean of averages of two die rolls is identical to the population mean of individual die rolls. This concept applies to other scenarios as well. For instance, if we were to flip a coin numerous times, we would expect approximately 50% of the outcomes to be zero (tails) and 50% to be one (heads). These proportions would converge to balance at around 0.5. When flipping the coin only a few times, the observed sample proportion may deviate from 0.5. However, as we increase the number of flips, the simulation variability becomes insignificant, and the proportion approaches 0.5.

If we flip the coin ten times, calculate the average, and repeat this process multiple times. This simulation provides insights into the distribution of averages of ten coin flips. We can extend this analysis to averages of 20 coin flips and averages of 30 coin flips. In each case, we observe that as the average incorporates more coin flips, the distribution becomes more concentrated around the mean. Nevertheless, regardless of the number of coin flips involved, the distribution of averages is consistently centered at 0.5.

To summarize the key points covered thus far:

* Expected values are inherent properties of distributions. The population mean represents the center of mass of that population, and any movement in the mean would correspondingly shift the distribution.
* The sample mean represents the center of mass of the observed data. It serves as an estimate of the population mean and is considered unbiased.
* The population mean of the distribution of sample means precisely matches the population mean it aims to estimate. This understanding is vital as it allows us to estimate the population distribution accurately when collecting substantial amounts of data.
* We must recognize that while we obtain only one sample mean from our data, knowing the properties associated with sample means is immensely valuable.
* As more data contributes to the sample mean, the density mass function becomes more concentrated around the population mean. We also observe that, even in cases such as coin flipping and dice rolling, the distribution tends to exhibit Gaussian-like characteristics. We’ll explore these concepts further in subsequent lectures.

## 1.5 Practical R Exercises in swirl

During this course we’ll be using the [swirl](http://swirlstats.com/) software package for R in order to illustrate some key concepts. The swirl package turns the R console into an interactive learning environment. Using swirl will also give you the opportunity to construct and explore your own regression models. In this programming assignment, you’ll have the opportunity to practice some key concepts from this course.

Since swirl is an R package, you can easily install it by entering a single command from the R console:

install.packages("swirl")

If you’ve installed swirl in the past make sure you have version 2.2.21 or later. You can check this with:

packageVersion("swirl")

Every time you want to use swirl, you need to first load the package. From the R console:

library(swirl)

Install the Statistical Inference course swirl offers a variety of interactive courses, but for our purposes, you want the one called Statistical Inference. Type the following from the R prompt to install this course:

install\_from\_swirl("Statistical Inference")

Start swirl and complete the lessons Type the following from the R console to start swirl:

swirl()

Then, follow the menus and select the Statistical Inference course when given the option. For the first part of this course you should complete the following lessons:

* Introduction
* Probability1
* Probability2
* ConditionalProbability
* Expectations

If you need help…

* Visit the [Frequently Asked Questions](https://github.com/swirldev/swirl/wiki/Coursera-FAQ) (FAQ) page to see if you can answer your own question immediately.
* Search the Discussion Forums this course.
* If you still can’t find an answer to your question, then create a new thread under the swirl Programming Assignment sub-forum and provide the following information:
  + A descriptive title
  + Any input/output from the console (copy & paste) or a screenshot
  + The output from sessionInfo()

Good luck and have fun!

For more information on swirl, visit Swirlstats(<https://swirlstats.com>).

# 2 Variability, Distribution, & Asymptotics

## 2.1 Variability

An important characterization of a population is how spread out it is. One of the key measures of spread is variability. We measure population variability with the sample variance, or more often we consider the square root of it, called the standard deviation. The reason for taking the standard deviation is because that measure has the same units as the population. So if our population is a length measurement in meters, the standard deviation is in meters (whereas the variance is in meters squared). Variability has many important uses in statistics. First, the population variance is itself an intrinsically interesting quantity that we want to estimate. Secondly, variability in our estimates is what makes them not imprecise. An important aspect of statistics is quantifying the variability in our estimates.

In the previous lecture, we discussed the population mean as a measure of the center of a distribution. Now, let’s explore another important property called variance, which describes the spread or concentration of the density around the mean. If we imagine a bell curve, the probability density function will shift to the left or right as the mean changes. Variance quantifies how widely or narrowly the density is distributed around the mean. 

library(ggplot2)  
x <- seq(-4, 4, length.out = 1000)  
mean <- 0  
  
# Standard deviations  
sds <- c(0.5, 1, 1.5, 2)  
  
# Create the plot  
plot(x, dnorm(x, mean = mean, sd = sds[1]), type = "l", lwd = 3, xlab = "x", ylab = "",yaxt="n")  
  
# Add lines for the other distributions with different colors  
colors <- c("blue", "green", "orange", "purple")  
for (i in 1:length(sds)) {  
 lines(x, dnorm(x, mean = mean, sd = sds[i]), col = colors[i], lwd = 3)  
}  
  
# Add legend  
legend("topright", legend = paste("sd=", sds), col = colors, lwd = 2,box.lwd = 0, box.col = "white")

The blue-colored density represents a standard normal distribution with a standard deviation of 0.5. As the standard deviation increases, the density becomes flatter and spreads more into the tails. Consequently, if a variable is from a normal distribution with a standard deviation of 1.5, they are more likely to have a value beyond 2 compared to a variable from a normal distribution with a standard deviation of 1.

For a random variable X with a mean μ, the variance is precisely the expected squared distance between the random variable and the mean.

There’s also a useful shortcut:

Densities with higher variance are more spread out compared to those with lower variances. The square root of variance is known as the standard deviation, which is expressed in the same units as X.

Example: In the previous lecture, we found that the expected value of X, when rolling a die, is 3.5.

To calculate the expected value of X squared, we square each number (1, 2, 3, 4, 5) and multiply them by their associated probabilities. Summing these values gives us 15.17.

Example: Consider tossing a coin with a probability of heads, . From the previous lecture, we know that the expected value of a coin toss is . When calculating the expected value of X squared, 0 squared is 0, and 1 squared is 1. Thus, the expected value of X squared is . Plugging these values into our formula, we get , which simplifies to . This formula is widely recognized and we suggest you memorize it.

Similar to the relationship between population mean and sample mean, the population variance and sample variance are directly analogous. The population mean represents the center of mass of the population, while the sample mean represents the center of mass of the observed data. Similarly, the population variance quantifies the expected squared distance of a random variable from the population mean, while the sample variance measures the average squared distance of the observed data points from the sample mean.

Note that in the denominator of the sample variance formula, we divide by instead of .

Recall that the sample variance is a function of the data, making it a random variable with its own population distribution. The expected value of this distribution corresponds to the population variance being estimated by the sample variance. As we gather more data, the distribution of the sample variance becomes increasingly concentrated around the population variance it seeks to estimate.

### 2.1.1 Variance simulation examples

Suppose we simulate ten standard normal random variables and calculate their sample variance. If we repeat this process many times, we will obtain a distribution of sample variances. This distribution, represented by the salmon-colored density, emerges from repeating the process thousands of times. If we sample enough data points, the center of mass of this distribution will precisely match the variance of the original population we were sampling from—the standard normal distribution with a variance of one.



Figure : Figure 1.3: Variance of a distribution of die roll

library(ggplot2)  
nosim <- 10000;   
dat <- data.frame(  
 x = c(apply(matrix(rnorm(nosim \* 10), nosim), 1, var),  
 apply(matrix(rnorm(nosim \* 20), nosim), 1, var),  
 apply(matrix(rnorm(nosim \* 30), nosim), 1, var)),  
 n = factor(rep(c("10", "20", "30"), c(nosim, nosim, nosim)))   
 )  
ggplot(dat, aes(x = x, fill = n)) + geom\_density(size = 2, alpha = .2) + geom\_vline(xintercept = 1, size = 2)

The same holds true when we consider sample variances based on 20 observations from the standard normal distribution. we repeat the process of sampling 20 standard normals, calculating the sample variance, and obtaining a distribution of sample variances. This distribution, depicted in a more aqua color, is also centered at one. The pattern continues when we examine sample variances based on 30 observations. However, what’s interesting to note is that as the sample size increases, the variance of the population distribution of the sample variances becomes more concentrated. In simpler terms, collecting more data leads to a better and more tightly focused estimate of what the sample variance is trying to estimate. In this case, all the sample variances are estimating a population variance of one because they are sampled from a population with a variance of one.

Before we found that the variance of a die roll was . Now, imagine if we were to roll ten dice and calculate the sample variance of the numbers on the sides facing up. By repeating this process numerous times, we can obtain a reliable understanding of the population distribution of the variance of ten die rolls. Although it requires a large number of repetitions, with the help of a computer, we can simulate this process thousands of times, as demonstrated here. 

dat <- data.frame(  
 x = c(apply(matrix(sample(1 : 6, nosim \* 10, replace = TRUE),   
 nosim), 1, var),  
 apply(matrix(sample(1 : 6, nosim \* 20, replace = TRUE),   
 nosim), 1, var),  
 apply(matrix(sample(1 : 6, nosim \* 30, replace = TRUE),   
 nosim), 1, var)  
 ),  
 size = factor(rep(c(10, 20, 30), rep(nosim, 3))))  
g <- ggplot(dat, aes(x = x, fill = size)) + geom\_histogram(alpha = .20, binwidth=.3, colour = "black")   
g <- g + geom\_vline(xintercept = 2.92, size = 2)  
g + facet\_grid(. ~ size)

Notice that the distribution of the variance of ten die rolls is precisely centered around 2.92, which is the variance of a single die roll. As I increase the number of dice to 20 and 30, the center of the distribution remains the same, but it becomes more concentrated around the true population variance. This indicates that the sample variance provides a good estimate of the population variance. As we collect more data, the distribution of the sample variance becomes increasingly concentrated around the true value it aims to estimate, demonstrating its unbiasedness.

The reason we divide by instead of in the denominator of the sample variance formula is to ensure its unbiasedness. ### Standard error of the mean Now that we have extensively discussed variances and briefly touched upon the distribution of sample variances, let’s revisit the distribution of sample means. It is important to remember that the average of numbers sampled from a population is a random variable with its own population mean and population variance. The population mean remains the same as the original population, while the variance of the sample mean can be related to the variance of the original population. Specifically, the variance of the sample mean decreases to zero as more data is accumulated.

This means that the sample mean becomes more concentrated around the population mean it is trying to estimate, which is a valuable characteristic since we usually only have one sample mean in a given dataset.

Although we do not have multiple repeated sample means to investigate their variability like we do in simulation experiments, we can still estimate the population variance, denoted as , using the available data. With knowledge of and the sample size (denoted as n), we can gather valuable information about the distribution of the sample mean. The square root of the statistic, sigma over square root n, is referred to as the standard error of the mean, denoted as the standard deviation of the distribution of a statistic. The term “standard error” is used to represent the variability of means, while the standard error of a regression coefficient describes the variability in regression coefficients.

In summary, considering a population with a mean and variance , when we draw a random sample from that population and calculate the variance , it serves as an estimate of . Similarly, when we calculate the mean, it estimates (population mean). However, (sample variance) is also a random variable with its own distribution centered around , becoming more concentrated around it as more observations contribute to the squared value. Additionally, the distribution of sample means from that population is centered at and becomes more concentrated around as more observations are included. Moreover, we precisely know the variance of the distribution of sample means, which is divided by n.

Since we lack repeated sample means in a given dataset, we estimate the sample variance of the mean as divided by n and the logical estimate of the standard error as . The standard error of the mean (or the sample standard error of the mean) is defined as . The standard deviation (S) is an estimate of the variability of the population, while the standard error () represents the variability of averages of random samples of size n from the population.

Example: If we take standard normals (with a variance of one), the standard deviation of means of n standard normals is expected to be one over . By simulating multiple draws of ten standard normals and calculating their mean, followed by taking the standard deviation of these averages, we should obtain an approximate value of .

You can explore this using the following code snippet in R:

## [1] 0.3192829

## [1] 0.3162278

Similar simulations can be performed for standard uniforms (variance of ), Poisson(4) distributions (variance of 4), and coin flips (variance of , assuming p=0.5 then ). The results of these simulations should align with the theoretical values predicted by our rule.

Understanding the standard error of the mean is crucial in determining the variability of sample means. Simulation experiments can help illustrate these concepts, especially when investigating the distribution of sample means and estimating their standard error.

Example: Consider the father-son data from UsingR library. We will focus on the height of the sons, with “n” representing the number of observations as usual. If we plot a histogram of the son’s height and overlay it with a continuous density estimate, we observe a distribution that closely resembles a Gaussian curve. 

library(UsingR); data(father.son);   
x <- father.son$sheight  
n<-length(x)  
g <- ggplot(data = father.son, aes(x = sheight))   
g <- g + geom\_histogram(aes(y = ..density..), fill = "lightblue", binwidth=1, colour = "black")  
g <- g + geom\_density(size = 2, colour = "black")  
g

This density estimate provides an approximation of the population density, given the finite amount of data we have collected. The histogram’s variability, which the sample variance calculates, serves as an estimate of the variability in son’s height from the population this data was drawn from, assuming it was a random sample.

By calculating the variance of x, variance of x divided by n, standard deviation of x, and standard deviation of , and rounding them to two decimal places, we obtain 7.92 and 2.81 as the variance of x and the standard deviation of x, respectively.

library(UsingR); data(father.son);   
x <- father.son$sheight  
n<-length(x)  
round(c(var(x), var(x) / n, sd(x), sd(x) / sqrt(n)),2)

These numbers represent the variability in son’s heights from the dataset and act as estimates of the population variability of son’s heights if we assume these sons are a random sample from a meaningful population. In this case, we prefer the value 2.81 over 7.92 since 7.92 is expressed in inches squared, while 2.81 is expressed in inches. Working with the actual units is more intuitive. Moving on to 0.01 and 0.09, these values no longer reflect the variability in children’s heights. Instead, they represent the variability in averages of ten children’s heights. The value 0.09 is particularly meaningful as it represents the standard error or the standard deviation in the distribution of averages of n children’s heights. While it’s an estimate based on the available data, it’s the best estimate we can derive from the dataset.

In this section we covered several complex topics, but at its core, understanding variability is the key to understanding statistics. In fact, grasping the concept of variability might be the most crucial aspect of statistics. Here’s a summary of our findings: the sample variance provides an estimate of the population variance, and the distribution of the sample variance is centered around the value it is estimating, indicating an unbiased estimation. Moreover, as more data is collected, the distribution becomes more concentrated around the estimated value, leading to a better estimate. We have also gained insights into the distribution of sample means. In addition to knowing its center, as discussed in the previous lecture, we now understand that the variance of the sample mean is the population variance divided by n, and its square root, is known as the standard error. These quantities capture the variability of averages drawn from the population, and surprisingly, even though we only have access to one sample mean in a given dataset, we can make substantial inferences about the distribution of averages from random samples. This knowledge provides us with a solid foundation for various statistical analyses and methodologies. ## Distributions Some probability distributions are so important that we need to internalize their characteristics. Here we will cover the most important probability distributions. ### Binomial distribution Perhaps the simplest distribution is known as the Bernoulli distribution, named after Jacob Bernoulli, a renowned mathematician from a distinguished family of mathematicians. If you’re interested, you can explore the Bernoulli family further through their Wikipedia pages. The Bernoulli distribution originates from a coin flip, where a “0” represents tails and a “1” represents heads. We can consider a potentially biased coin with a probability “p” for heads and “1 - p” for tails. The Bernoulli probability mass function is typically denoted as:

As we have seen before, the mean of a Bernoulli random variable is , and the variance is . In the context of a Bernoulli random variable, we often refer to “x = 1” as a success, irrespective of the specific definition of success in a given scenario, and “x = 0” as a failure.

A binomial random variable is obtained by summing up a series of independent and identically distributed (iid) Bernoulli random variables. Essentially, a binomial random variable represents the total number of heads obtained in a series of coin flips with a potentially biased coin. Mathematically, if we let to be iid Bernoulli variables with parameter , then the sum of these variables, denoted as , is a binomial random variable.

The binomial probability mass function closely resembles the Bernoulli mass function, but with the inclusion of “n choose x” in front. The notation “n choose x” represents the binomial coefficient, calculated as . It is worth noting that “n choose 0” and “n choose n” both equal 1. This coefficient helps solve a common combinatorial problem, counting the number of ways to select “x” items out of “n” without replacement while disregarding the ordering of the items.

Example: Suppose your friend has eight children, with seven of them being girls (and no twins). Assuming each gender has an independent 50% probability for each birth, what is the probability of having seven or more girls out of eight births? We can apply the binomial formula to calculate this probability:

In the provided R code, you can find the implementation of this calculation. Furthermore, for most common distributions, including the binomial distribution, there are built-in functions in R. For example, the pbinom function can be used to obtain these probabilities conveniently.

choose(8, 7) \* .5 ^ 8 + choose(8, 8) \* .5 ^ 8   
pbinom(6, size = 8, prob = .5, lower.tail = FALSE)

### 2.1.2 Normal distribution

Probabilities play a crucial role in statistics, and among all the distributions, the normal distribution stands out as the most important one. In the upcoming lecture, we will explore why it holds such significance. In fact, if all distributions were to gather and elect a leader, the normal distribution would undoubtedly take the crown.

A random variable that follows a normal (Gaussian) distribution with a mean of and a variance of . This distribution is characterized by a density function that resembles a bell curve. If we have a random variable X with this density, its expected value is μ, and its variance is . We can express this concisely as , denoting a normal distribution with mean μ and variance . When μ=0 and σ=1, the resulting distribution is known as the *standard normal distribution*. Standard normal random variables are often denoted by the letter . Here, we depict the standard normal density function, which represents the famous bell curve you have likely encountered before.

x <- seq(-3, 3, length = 1000)  
library(ggplot2)  
g <- ggplot(data.frame(x = x, y = dnorm(x)),   
 aes(x = x, y = y)) + geom\_line(size = 2)  
g <- g + geom\_vline(xintercept = -3 : 3, size = 2)  
g



Figure : Figure 1.12: Standard normal distribution

It is important to note that for the standard normal distribution, the mean is 0, and the standard deviation (and variance) is 1. In the diagram, we illustrate one standard deviation above and below the mean, two standard deviations above and below the mean, and three standard deviations above and below the mean. The units on the standard normal distribution can be interpreted as standard deviation units. Additionally, it is worth mentioning that statisticians often find it convenient to revert to the standard normal distribution when discussing normal probabilities, even when dealing with non-standard normal distributions. Therefore, if you want to calculate the probability that a non-standard normal lies between and (where μ and σ are specific to its distribution), the probability area is equivalent to that between -1 and +1 on the standard normal distribution. In essence, all normal distributions have the same underlying shape, with the only difference being the units along the axis. By reverting to standard deviations from the mean, all probabilities and calculations can be transformed back to those associated with the standard normal distribution.

Some fundamental reference probabilities related to the standard normal distribution can be easily explained using the graph above as visual aids. First, consider one standard deviation from the mean in the standard normal distribution (or any normal distribution). Approximately 34% of the distribution lies on each side, resulting in a total area of 68% within one standard deviation. Moving on to two standard deviations, denoted by the magenta area in the diagram, around 95% of the distribution falls within this range for any normal distribution. This leaves 2.5% in each tail, and we often utilize this information when calculating confidence intervals. Lastly, when considering three standard deviations from the mean, the area encompasses approximately 99% of the distribution’s mass, although it may be difficult to discern from the diagram. *These reference probabilities are essential to commit to memory.*

Probabilities are a fundamental concept, and the normal distribution holds a special place in statistics. Understanding its properties and the relationship to the standard normal distribution allows us to solve problems effectively. All normal distributions share the same essential shape, differing only in their units along the axis. By leveraging the standard normal distribution and converting the non standard normals to standard normals, we can simplify calculations and derive consistent results. The primary difference between different normal distributions lies in the units along the axis. When discussing normal probabilities and converting to standard deviations from the mean, all probabilities and calculations revert back to those associated with the standard normal distribution.

Rules for converting between standard and non-standard normal distributions. If we have a random variable X that follows a normal distribution with a mean of μ and variance of σ squared, we can convert the units of X to standard deviations from the mean by subtracting the mean μ and dividing by the standard deviation σ. If then:

The resulting random variable Z will follow a standard normal distribution. Conversely, if we start with a standard normal random variable Z and want to convert back to the units of the original data, we multiply Z by σ and add μ. If is standard normal

The resulting random variable X will then follow a non-standard normal distribution with a mean of μ and variance of .

Standard normal quantiles that are important to remember -1.28 is a quantile such that 10% of the density lies below it, and 90% lies above it. By symmetry, 1.28 on the standard normal distribution represents the quantile at which 10% lies above it. For a potentially non-standard normal distribution, this point would be . Another crucial quantile is 1.96 (often approximated as 2), where -1.96 represents the point below which 2.5% of the mass of the normal distribution lies, and +1.96 represents the point above which 2.5% of the mass lies. This implies that 95% of the distribution lies between these two points. For a potentially non-standard normal distribution, these points would be and , respectively. It is worth noting that when μ equals 0 and σ equals 1 for the standard normal distribution, the calculation of 1.96 directly yields the correct value.

Example: Determine the percentile of a normal distribution with mean μ and variance σ squared. In other words, we seek the value such that 95% of the distribution lies below it. This value represents the threshold if we were to draw samples from this population.

We can find the point , which represents the percentile of a normal distribution, by utilizing the q qualifier for the density in R. In this case, we can use the function qnorm with the desired quantile 0.95.

qnorm(0.95, mean = mu, sd = sd)

It’s crucial to input the mean μ and the standard deviation σ (not the variance) into the function. By using qnorm with the specified parameters, we can directly obtain the desired value. Another approach to solving this is by leveraging our memorized standard normal quartiles. Since we know that 1.645 standard deviations from the mean corresponds to a quantile with 95% lying below it and 5% lying above it for the standard normal distribution (centered at 0 with standard deviation units from the mean), we can apply this concept to a non-standard normal distribution as well. To calculate the desired point, we can simply compute .

Example: What is the probability that a non-standard normal distribution is larger than ? To answer this question in R, we can use the pnorm function with the specified values of , , and standard deviation . It’s important to remember to input the sigma value rather than the value to avoid incorrect results. Additionally, we set the argument lower.tail = FALSE to indicate that we are interested in the upper tail of the distribution. Alternatively, we can omit this argument and calculate to achieve the same result.

A conceptually easy way to estimate this probability, which allows us to quickly assess probabilities mentally, is to convert the value x into the number of standard deviations it is from the mean. To achieve this, we compute . The resulting number represents x expressed in terms of how many standard deviations it is from the mean. For example, if the calculated value is approximately two standard deviations from the mean, we can estimate that the probability associated with it is around 2.5%.

Example: The number of daily ad clicks for companies follows an approximately normal distribution with a mean of 1020 clicks per day and a standard deviation of 50 clicks per day. We want to determine the probability of getting more than 1160 clicks on a given day.

Since which means 2.8 standard deviation away from the mean, we can infer that this probability will be relatively low. This is because it is nearly 3 standard deviations away from the mean, and we know that such values are located in the tail of the normal distribution. To calculate this probability, we can use the pnorm function with the input values of 1,160 for the clicks, a mean of 1,020, and a standard deviation of 50.

pnorm(1160, mean = 1020, sd = 50, lower.tail = FALSE)  
pnorm(2.8, lower.tail = FALSE)

By setting the argument lower.tail = FALSE, we ensure that we obtain the probability of the value being larger than 1,060. The result we obtain is approximately 0.003.

Alternatively, we can directly calculate this probability using the standard normal distribution. By expressing 1,160 as the number of standard deviations it is away from the mean, which is 2.8, we can plug this value into the pnorm function with lower.tail = FALSE and obtain the same result.

pnorm(2.8, lower.tail = FALSE)

Example: Assuming the number of daily ad clicks for the company follows an approximately normal distribution with a mean of 1020 and a standard deviation of 50, we want to find the number of daily ad clicks that represents the point where 75% of the days have fewer clicks.

Since 1020 is both the mean and the median of the specific normal distribution, we know that about 50% of the days lie below this point. Therefore, the desired number of clicks should be greater than 1020. Additionally, one standard deviation above the mean, which corresponds to 1,070. Within this range, we know that 68% of the days lie, leaving 32% outside of it, and 16% in each tail due to the symmetry of the normal distribution. Hence, the desired number of clicks should be around 84% of the distribution, lying between 1,020 and 1,070.

To calculate this quantile, we can use the qnorm function with the input value of 0.75, representing the 75th percentile. The mean is set to 1020, and the standard deviation is 50. When we execute this command, qnorm(0.75, mean = 1020, sd = 50), we obtain a number between the previously mentioned range, approximately 1054.

### 2.1.3 Poisson distribution

If there were a competition to determine the most useful distribution, the normal distribution would unquestionably win by a wide margin. However, selecting the second most useful distribution would spark a lively debate, with the Poisson distribution being a strong contender. The Poisson distribution is commonly employed to model counts, and its probability mass function is given by

where x represents non-negative integers (0, 1, 2, and so on). The mean of a Poisson random variable is equal to , and the variance also equals . When modeling with Poisson data, the mean and variance must be equal a condition that can be verified if one has repeated Poisson data.

The Poisson distribution finds utility in various instances. Whenever count data needs to be modeled, especially when the counts are unbounded, the Poisson distribution is a suitable choice. Another prevalent application arises in the field of biostatistics, where event time or survival data is common. For example, in cancer trials, the time until the recurrence of symptoms is modeled using statistical techniques that account for censoring, and these techniques have a strong association with the Poisson distribution. Additionally, when classifying a sample of people based on certain characteristics, creating a contingency table—such as tabulating hair color by race—the Poisson distribution is the default choice for modeling such data. The Poisson distribution is deeply connected to other models, including multinomials and binomials, which might be considered as alternatives.

Another prominent application of the Poisson distribution, though often overlooked due to its commonplace usage, is in cases where a binomial distribution is approximated by the Poisson distribution. This occurs when the sample size (n) is large, and the probability of success (p) is small. Epidemiology, for instance, frequently employs this approximation when dealing with situations where n is large (representing a population) and p is small (indicating the occurrence of rare events). By assuming a Poisson distribution, researchers can effectively model the occurrence rates of events, such as the number of new cases of respiratory diseases in a city as air pollution levels fluctuate. This practice is so prevalent that it is commonly understood within the field without explicit mention.

Example: The number of people showing up at a bus stop follows a Poisson distribution with a mean of 2.5 people per hour. If we observe the bus stop for four hours, we can calculate the probability of three or fewer people showing up during that entire duration. To do this, we apply the Poisson probability formula to the values of three, two, one, and zero, using a rate of 2.5 events per hour multiplied by four hours. The resulting probability is approximately 1%.

ppois(3, lambda = 2.5 \* 4)

Furthermore, we can discuss the Poisson approximation to the binomial distribution, specifically when the sample size (n) is large, and the probability of success (p) is small. In this scenario, the Poisson distribution can serve as a reasonably accurate approximation for the binomial distribution. To establish notation, let x represent a binomial distribution with parameters n and p, and define . When n is large and p is small, it is proposed that the probability distribution governing x can be well approximated using Poisson probabilities, where the rate parameter λ is determined as n times p.

Example: In flipping a coin with a success probability of 0.01 for a total of 500 times, we want to calculate the probability of obtaining two or fewer successes. Using the binomial distribution with size 500 and probability 0.01, we obtain approximately 12%. By employing the Poisson approximation with a rate of λ = 500 \* 0.01, the result is around 12.5%, which is reasonably close to the binomial calculation.

pbinom(2, size = 500, prob = .01)  
ppois(2, lambda=500 \* .01)

## 2.2 Asymptotics

Asymptotics are an important topics in statistics. Asymptotics refers to the behavior of estimators as the sample size goes to infinity. Our very notion of probability depends on the idea of asymptotics. For example, many people define probability as the proportion of times an event would occur in infinite repetitions. That is, the probability of a head on a coin is 50% because we believe that if we were to flip it infinitely many times, we would get exactly 50% heads.

We can use asymptotics to help is figure out things about distributions without knowing much about them to begin with. A profound idea along these lines is the *Central Limit Theorem*. It states that the distribution of averages is often normal, even if the distribution that the data is being sampled from is very non-normal. This helps us create robust strategies for creating statistical inferences when we’re not willing to assume much about the generating mechanism of our data.

### 2.2.1 Asymptotics and LLN

Here we will explore the behavior of statistics as the sample size or some other relevant quantity approaches infinity, which is known as asymptotics. Specifically, we will discuss the case where the sample size tends to infinity.

In the land of asymptopia, everything works out well because there is an infinite amount of data available. Asymptotics play a crucial role in simple statistical inference and approximations. They serve as a versatile tool, akin to a Swiss army knife, allowing us to investigate the statistical properties of various statistics without requiring extensive computations. Asymptotics form the foundation for the frequency interpretation of probabilities. For instance, intuitively, we know that if we flip a coin and calculate the proportion of heads, it should approach 0.5 for a fair coin.

Fortunately, we don’t have to delve into the mathematical intricacies of the limits of random variables. Instead, we can rely on a set of powerful tools that enable us to discuss the behavior of sample means from a collection of independently and identically distributed (iid) observations in large samples. One of these tools is the law of large numbers, which states that the average of the observations converges to the population mean it is estimating. For example, if we repeatedly flip a fair coin, the sample proportion of heads will eventually converge to the true probability of a head.

Example: We’ll generate a large number of random normal variables and calculate their cumulative means. Initially, there is considerable variability in the means, but as the number of simulations increases, the cumulative means converge towards the true population mean of zero.



Figure : Figure 2.2: Cumulative means of random normal variables

n <- 10000; means <- cumsum(rnorm(n)) / (1 : n); library(ggplot2)  
g <- ggplot(data.frame(x = 1 : n, y = means), aes(x = x, y = y))   
g <- g + geom\_hline(yintercept = 0) + geom\_line(size = 2)   
g <- g + labs(x = "Number of obs", y = "Cumulative mean")  
g

Similarly, we can apply the law of large numbers to the case of coin flipping. By repeatedly flipping a coin and calculating the cumulative means, we observe that the sample proportion of heads converges to the true value of 0.5 as the number of coin flips increases.



Figure : Figure 2.3: Cumulative means of coin flips

means <- cumsum(sample(0 : 1, n , replace = TRUE)) / (1 : n)  
g <- ggplot(data.frame(x = 1 : n, y = means), aes(x = x, y = y))   
g <- g + geom\_hline(yintercept = 0.5) + geom\_line(size = 2)   
g <- g + labs(x = "Number of obs", y = "Cumulative mean")  
g

Note: An estimator is considered *consistent* if it converges to the parameter it aims to estimate. For instance, the sample proportion from iid coin flips is consistent for estimating the true success probability of a coin. As we collect more and more coin flip data, the sample proportion of heads approaches the actual probability of obtaining a head. Moreover, not only are sample means consistent estimators, but the sample variance and sample standard deviation of iid random variables are also consistent estimators.

The law of large numbers guarantees the consistency of sample means, but it also applies to sample variances and standard deviations of iid random variables. In other words, these estimators also converge to their respective population counterparts as the sample size increases.

### 2.2.2 Asymptotics and the CLT

The Central Limit Theorem (CLT) is perhaps the most important theorem in statistics. It states that the distribution of averages of iid random variables becomes approximately standard normal as the sample size grows. The Central Limit Theorem is remarkably versatile, applying to a wide range of populations. Its loose requirements make it applicable in numerous settings.

To understand the Central Limit Theorem, let’s consider an estimate like the sample average . If we subtract its population mean and divide by its standard error the resulting random variable approaches a standard normal distribution as the sample size increases.

Importantly, replacing the unknown population standard deviation with the known sample standard deviation does not affect the Central Limit Theorem.

The most useful interpretation of the Central Limit Theorem is that the sample average is approximately normally distributed, with a mean equal to the population mean and a variance given by the standard error of the mean.

Example: Using standard die with the mean of 3.5, and variance of 2.92. We simulate the die roll n times, calculate the sample mean, subtract the population mean, and dividing by the standard error.

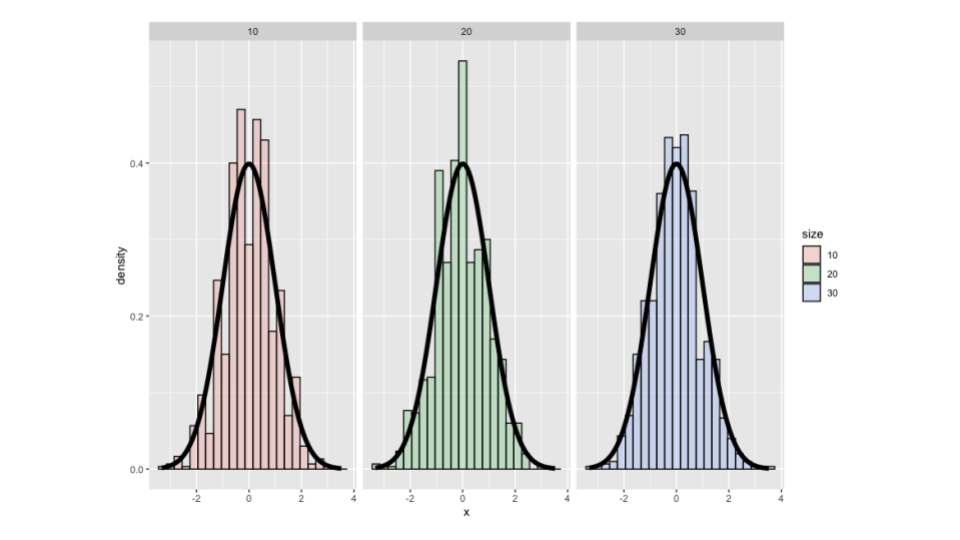


Figure : Figure 2.4: Distribution of averages of iid random variables in die roll

nosim <- 1000  
cfunc <- function(x, n) sqrt(n) \* (mean(x) - 3.5) / 1.71  
dat <- data.frame(  
 x = c(apply(matrix(sample(1 : 6, nosim \* 10, replace = TRUE),   
 nosim), 1, cfunc, 10),  
 apply(matrix(sample(1 : 6, nosim \* 20, replace = TRUE),   
 nosim), 1, cfunc, 20),  
 apply(matrix(sample(1 : 6, nosim \* 30, replace = TRUE),   
 nosim), 1, cfunc, 30)  
 ),  
 size = factor(rep(c(10, 20, 30), rep(nosim, 3))))  
g <- ggplot(dat, aes(x = x, fill = size)) + geom\_histogram(alpha = .20, binwidth=.3, colour = "black", aes(y = ..density..))   
g <- g + stat\_function(fun = dnorm, size = 2)  
g + facet\_grid(. ~ size)

The distribution approximates a bell curve. As we increase the number of rolls, the approximation improves.

Example: Let be the or result of the flip of a possibly unfair coin. The sample proportion, say , is the average of the coin flips. and Standard error of the mean is Then

will be approximately normally distributed

Flipping a fair coin times, taking the sample proportion of heads, subtracting off 0.5 and multiply the result by divide by is displayed below.

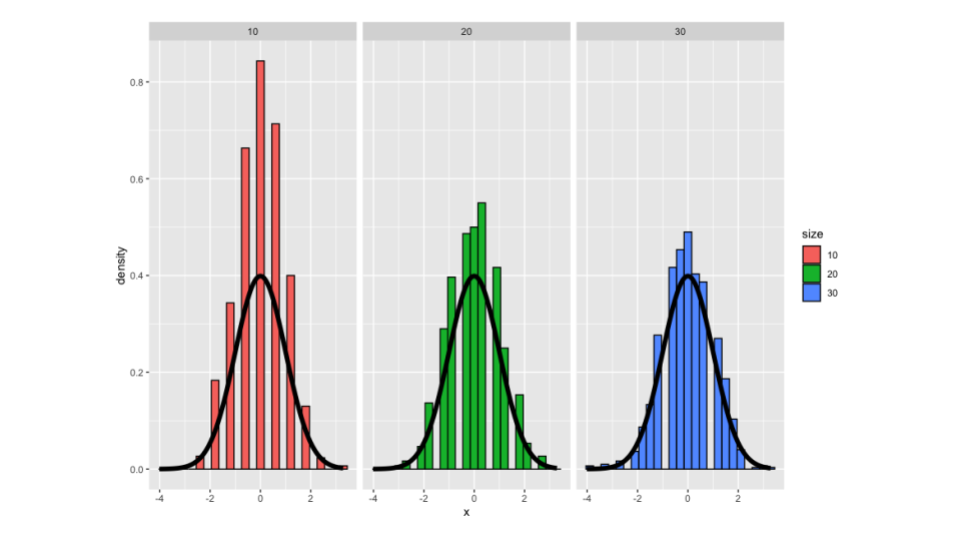


Figure : Figure 2.5: Distribution of averages of iid random variables in coin flip

nosim <- 1000  
cfunc <- function(x, n) 2 \* sqrt(n) \* (mean(x) - 0.5)   
dat <- data.frame(  
 x = c(apply(matrix(sample(0:1, nosim \* 10, replace = TRUE),   
 nosim), 1, cfunc, 10),  
 apply(matrix(sample(0:1, nosim \* 20, replace = TRUE),   
 nosim), 1, cfunc, 20),  
 apply(matrix(sample(0:1, nosim \* 30, replace = TRUE),   
 nosim), 1, cfunc, 30)  
 ),  
 size = factor(rep(c(10, 20, 30), rep(nosim, 3))))  
g <- ggplot(dat, aes(x = x, fill = size)) + geom\_histogram(binwidth=.3, colour = "black", aes(y = ..density..))   
g <- g + stat\_function(fun = dnorm, size = 2)  
g + facet\_grid(. ~ size)

Taking the result of each flip (0 or 1) as an iid random variable, we calculate the sample proportion of heads . We again obtain a distribution that approximates a bell curve. Similar to the previous example, the approximation improves as the number of coin flips increases.

It’s important to note that the speed at which the normalized coin flips converge to normality depends on the bias of the coin. If the coin is heavily biased, the approximation may not be perfect even with a large sample size. However, as the number of coin flips approaches infinity, the Central Limit Theorem guarantees an excellent approximation.

As a fun example, let’s discuss Galton’s quincunx. This machine, often found in science museums, visually demonstrates the Central Limit Theorem using a game resembling Pachinko. [An image from Wikipedia showing Galton’s quincunx](https://upload.wikimedia.org/wikipedia/commons/c/c1/Galton_box.jpg). In Galton’s quincunx, a ball falls through a series of pegs, bouncing left or right at each peg. Each bounce can be thought of as a coin flip or binomial experiment. The total number of successes (heads) follows an approximately normal distribution, as predicted by the Central Limit Theorem. At the museum, the balls collect in bins, forming a histogram that aligns with the expected normal distribution.

In summary, the Central Limit Theorem is a powerful tool that allows us to approximate the distribution of averages of iid random variables. It applies to various settings and provides valuable insights into statistical inference. The examples we explored, from dice rolls to coin flips to Galton’s quincunx, illustrate the practical applications of the Central Limit Theorem and the convergence to a standard normal distribution as the sample size increases.

### 2.2.3 Asymptotics and confidence intervals

The central limit theorem tells us that the sample mean follows an approximately normal distribution with a population mean of μ and a standard deviation of . This distribution allows us to make inferences about the population mean based on sample data. When considering the distribution, we observe that standard errors is quite far out in the tail, with only a 2.5% chance of a normal value being larger than two standard deviations in the tail. Similarly, standard errors is far in the left tail, with only a 2.5% chance of a normal value being smaller than two standard deviations in the left tail. Therefore, the probability that the sample mean is greater than standard errors or smaller than standard errors is 5%. Equivalently, the probability that μ is between these limits is 95%. By reversing the roles of and μ, we can conclude that the interval contains μ with a probability of 95%.

It’s important to note that in this interpretation, we treat the interval as random, while μ is fixed. This allows us to discuss the probability that the interval contains μ. In practice, if we repeatedly obtain samples of size n from the population and construct a confidence interval in each case, about 95% of the intervals will contain μ, the parameter we are trying to estimate. If we want a 90% confidence interval, we need 5% in each tail, so we would use a different multiplier instead of 2 (e.g., 1.645).

Example: Using the father-son data from the “Using R” package we want to estimate the average height of sons . We can calculate the mean of the sample plus or minus the 0.975th normal quantile times the standard error of the mean.

library(UsingR)  
data(father.son)  
x <- father.son$sheight  
(mean(x) + c(-1, 1) \* qnorm(.975) \* sd(x) / sqrt(length(x))) / 12

Dividing by 12 ensures that our confidence interval is in feet rather than inches. If we obtain a confidence interval of 5.710 to 5.738, we can say that if the sons’ height in this data are a random sample from the population of interest, the confidence interval for the average height of the sons would be 5.71 to 5.74.

Another application is when dealing with coin flips and estimating the success probability of the coin. Each observation in this case is either 0 or 1, with a common success probability . The variance of a coin flip is , where p is the true success probability of the coin. The standard error of the mean is then:

Since we don’t know the true value of , we replace it with the estimated value . This type of confidence interval is known as the *Wald confidence interval*, named after the statistician Wald. When p equals 0.5, the variance is maximized, resulting in a standard error of 0.5. Multiplying it by 2 in the 95% interval cancels out, leaving the following expression for a 95% confidence interval, which is a quick estimate for p:

Example: Imaging you are running for political office, and in a random sample of 100 likely voters, 56 intend to vote for you. To determine if you can relax or if you need to campaign more, you can use a quick calculation.

With the information from the sample you can with with probability of 0.56 taking means the approximate 95% interval is 0.46 to 0.66. The confidence interval suggests that we cannot rule out possibilities below 0.5 with 95% confidence. Therefore, you shouldn’t relax and should continue campaigning.

As a general guideline, you typically need at least 100 observations for one decimal place in a binomial experiment, 10,000 for two decimal places, and a million for three decimal places. These numbers reflect the approximate sample sizes needed for accurate estimation.

In summary, the central limit theorem provides us with a practical tool for constructing confidence intervals and making inferences about population parameters. It allows us to estimate the population mean using the sample mean and provides a measure of uncertainty through confidence intervals. The Wald confidence interval is a useful approximation for estimating the success probability in binomial experiments. Additionally, considering the sample size helps determine the level of precision and confidence in our estimates.

Consider a simulation where we repeatedly flip a coin with a known success probability. The goal is to calculate the percentage of times that the confidence interval covers the true probability. In each simulation, we flip the coin 20 times and vary the true success probability between 0.1 and 0.9 in steps of 0.05. We conduct 1,000 simulations for each true success probability.

n <- 20; pvals <- seq(.1, .9, by = .05); nosim <- 1000  
coverage <- sapply(pvals, function(p){  
 phats <- rbinom(nosim, prob = p, size = n) / n  
 ll <- phats - qnorm(.975) \* sqrt(phats \* (1 - phats) / n)  
 ul <- phats + qnorm(.975) \* sqrt(phats \* (1 - phats) / n)  
 mean(ll < p & ul > p)  
})

For each true success probability, we generate 1,000 sets of 20 coin flips and calculate the sample proportion. Then, we compute the lower and upper limits of the confidence interval for each set of coin flips. Finally, we determine the proportion of times that the confidence interval covers the true value of the success probability. we store these proportions in a variable called “coverage.”

To visualize the results, we can plot the coverage as a function of the true success probability used in the simulation. For example, if the true value of p is 0.5, we perform 1,000 simulations and calculate the coverage based on whether the confidence interval covers 0.5 or not. In this case, the coverage is over 95%, indicating that the confidence interval provides better than 95% coverage for a true success probability of 0.5.

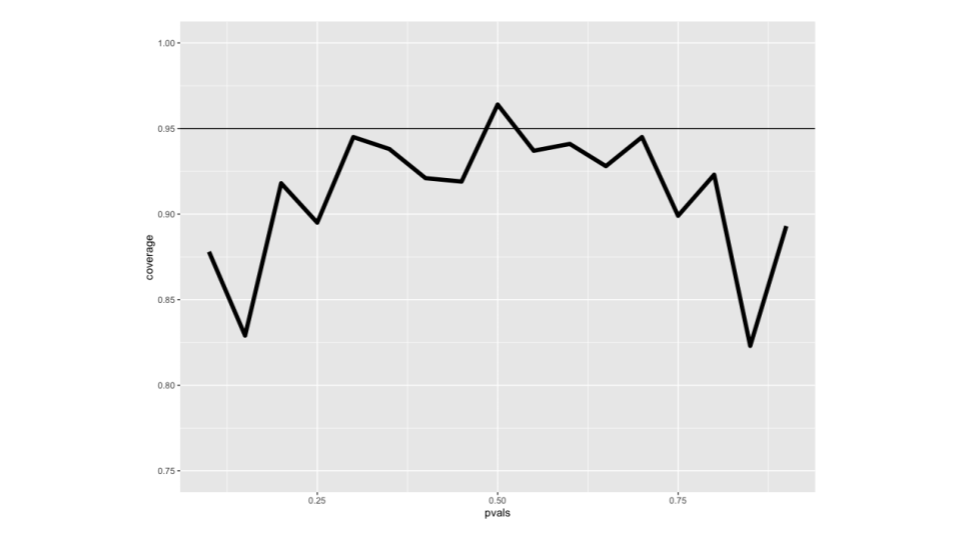


Figure : Figure 2.6: Coverage of confidence intervals for coin flips, n=20

Although there is some Monte Carlo error due to the finite number of simulations, 1,000 simulations generally yield good accuracy. For a true success probability around 12%, the coverage falls well below the expected 95%. The reason behind this discrepancy is that the central limit theorem is not accurate enough for this specific value of n (the number of coin flips) and the true probability. To address this issue for smaller values of n, a quick fix is to add 2 to the number of successes and 2 to the number of failures. This adjustment modifies the sample proportion, making it . After applying this adjustment, the confidence interval procedure can be performed as usual. This modified interval is known as the *Agresti/Coull interval* and tends to perform better than the standard Wald interval. Before demonstrating the results for the adjusted intervals, it is important to note that larger values of n yield better performance. In a simulation where n is increased to 100, the coverage probability improves and remains close to the expected 95% across different values of p.

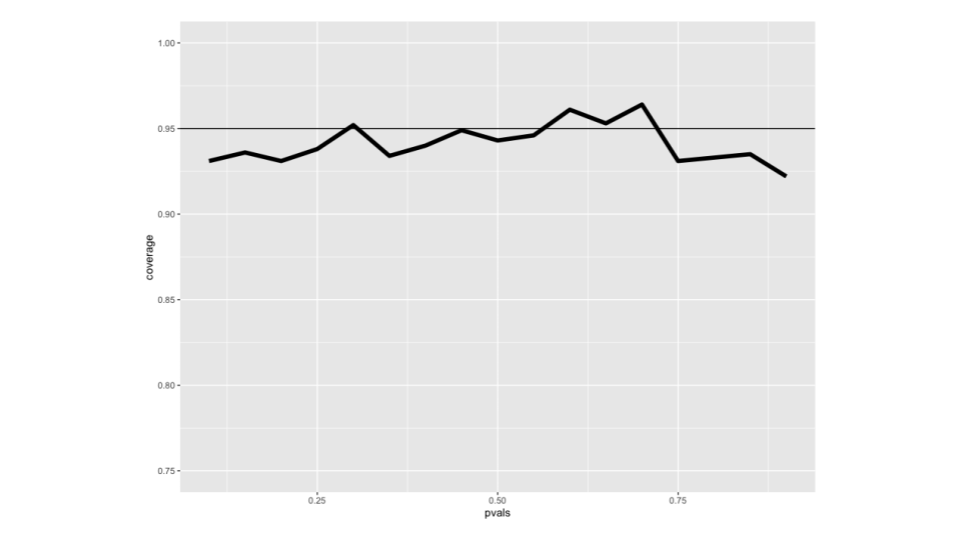


Figure : Figure 2.7: Coverage of confidence intervals for coin flips, n=100

Returning to the simulation with n=20, when using the add 2 successes and 2 failures interval, the coverage probability is higher than 95%, indicating an improvement compared to the poor coverage of the Wald interval for certain true probability values. However, it’s important to balance coverage and interval width, as being too conservative can lead to overly wide intervals. Based on these observations, we strongly recommend using the add 2 successes and 2 failures interval instead of the Wald interval in this specific scenario.

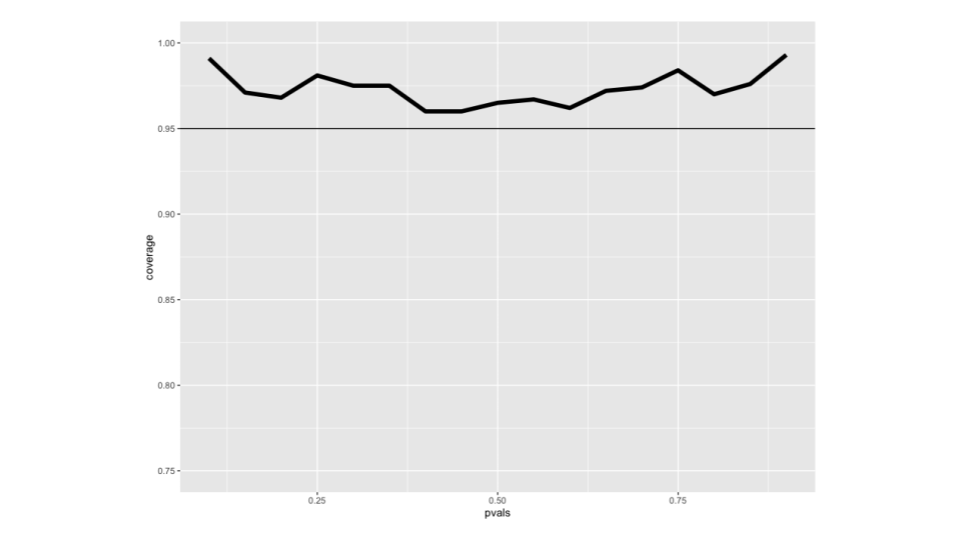


Figure : Figure 2.8: Coverage of confidence intervals for coin flips, n=20, add 2 successes and 2 failures

Example: Create a Poisson interval using the formula that involves the estimate plus or minus the normal quantile standard error. Although the application of the central limit theorem in this case may be less clear, we will discuss it shortly.Consider a nuclear pump that failed 5 times out of 94.32 days over a monitoring period. We want to calculate a 95% confidence interval for the failure rate per day. Assuming the number of failures follows a Poisson distribution with a failure rate of lambda and the monitoring period is denoted as t, the estimate of the failure rate is the number of failures divided by the total monitoring time. The variance of this estimate is .

x <- 5; t <- 94.32; lambda <- x / t  
round(lambda + c(-1, 1) \* qnorm(.975) \* sqrt(lambda / t), 3)  
poisson.test(x, T = 94.32)$conf

In the calculations performed in R, the number of events is set to 5, and the monitoring time is 94.32. The rate estimate is computed as , and the confidence interval estimate is obtained by adding or subtracting the relevant standard normal quantile multiplied by the standard error. The resulting interval is rounded to three decimal places. In addition to the large sample interval, we can also calculate an exact Poisson interval using the poisson.test function in R. This exact interval guarantees the specified coverage (e.g., 95%), but it may be conservative and result in wider intervals than necessary.

To examine how confidence intervals perform in repeated samplings, let’s conduct a simulation similar to the one for the coin example, but for the Poisson coverage rate. We select a range of values around those from our previous example and perform 1,000 simulations. The monitoring time is set to 100 for simplicity. We define coverage as the percentage of times the simulated interval contains the true value used in the simulation. The simulation is repeated for various values, and the resulting plot shows the values on the x-axis and the estimated coverage on the y-axis.

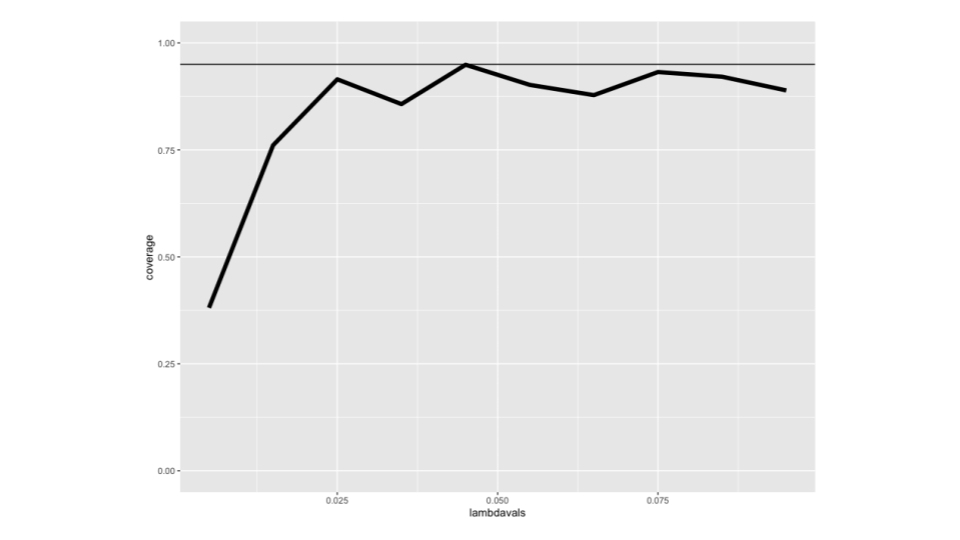


Figure : Figure 2.9: Coverage of confidence intervals for Poisson data

The plot reveals that as values increase, the coverage approaches 95%. However, there is some Monte Carlo error due to the finite number of simulations. On the other hand, as the true value becomes smaller, the coverage deteriorates significantly. For very small values, the purported 95% interval may only provide 50% actual coverage. To address this issue, it is recommended not to rely on the asymptotic interval for small values, especially when there are relatively few events during a large monitoring time. In such cases, the asymptotic interval does not align well with the Poisson distribution. Instead, an exact Poisson interval can be used as an alternative.

Although the central limit theorem’s application in the Poisson case may not be immediately clear, a simulation with a larger monitoring time (e.g., changing t from 100 to 1,000) demonstrates that as the monitoring time increases, the coverage improves and converges to 95% for most values. However, some poor coverage may still occur for small values, which we know the interval has trouble handling. In such cases, the exact Poisson interval remains a viable option.

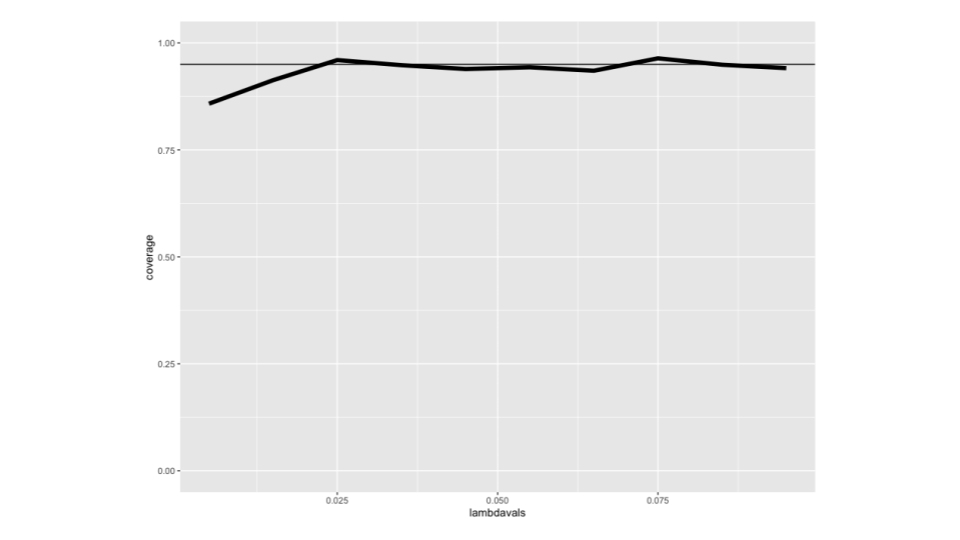


Figure : Figure 2.10: Coverage of confidence intervals for Poisson data, t=1000

To summarize briefly, we covered the Law of Large Numbers, which states that averages of independent and identically distributed (iid) random variables converge to the quantities they are estimating. This applies to Poisson rates as well, although the convergence process may be less clear. As the monitoring time tends to infinity, for example, Poisson rates converge to their estimated values. We also discussed the Central Limit Theorem, which states that averages are approximately normally distributed. These distributions are centered at the population mean, a concept we already knew without the theorem, with standard deviations equal to the standard error of the mean. However, the Central Limit Theorem does not guarantee that the sample size is large enough for this approximation to be accurate. We have observed instances where confidence intervals are very accurate and others where they are less accurate. Speaking of confidence intervals, our default approach for constructing them is to take the mean estimate and add or subtract the relevant normal quantile times the standard error. This method, known as “walled intervals,” is used not only in this context but also in regression analysis, general linear models, and other complex subjects. For a 95% confidence interval, the quantile value can be taken as 2 or, for more accuracy, 1.96. Confidence intervals become wider as the desired coverage increases within a specific technique. This is because wider intervals provide more certainty that the parameter lies within them. To illustrate, imagine an extreme scenario where your life depends on the confidence interval containing the true parameter. In this case, you would want to make the interval as wide as possible to ensure your safety. The mathematics behind confidence intervals follows the same principle. In the cases of Poisson and binomial distributions, which are discrete, the Central Limit Theorem may not accurately approximate their distributions. However, exact procedures exist for these cases. We also learned a simple fix for constructing confidence intervals in the binomial case by adding two successes and two failures, which provides a better interval without requiring complex computations. This method can be easily done by hand or mentally, even without access to a computer.

* The LLN states that averages of iid samples converge to the population means that they are estimating
* The CLT states that averages are approximately normal, with distributions
  + centered at the population mean
  + with standard deviation equal to the standard error of the mean
  + CLT gives no guarantee that is large enough
* Taking the mean and adding and subtracting the relevant normal quantile times the SE yields a confidence interval for the mean
  + Adding and subtracting 2 SEs works for 95% intervals
* Confidence intervals get wider as the coverage increases (why?)
* Confidence intervals get narrower with less variability or larger sample sizes
* The Poisson and binomial case have exact intervals that don’t require the CLT
  + But a quick fix for small sample size binomial calculations is to add 2 successes and failures

## 2.3 Practical R Exercises in swirl

During this week of the course you should complete the following lessons in the Statistical Inference swirl course:

* Variance
* CommonDistros
* Asymptotics

# 3 Intervals, Testing, & Pvalues

When we estimate something using statistics, usually that estimate comes with uncertainty. Take, for example, election polling. When we get a polled percentage of voters that favor a candidate, we were only able to sample a small subset of voters. Therefore, our estimate has uncertainty associated with it. Confidence intervals are a convenient way to communicate that uncertainty in estimates. In this lecture, we will discuss confidence intervals. We will also discuss the concept of hypothesis testing and p-values.

## 3.1 Confidence intervals

In the previous section, we explored the creation of confidence intervals using the central limit theorem. The intervals we discussed followed:

where the estimates are plus or minus a quantile from the standard normal distribution multiplied by the estimated standard error. Here, we will focus on methods suitable for small sample sizes. Specifically, we will discuss the Student’s or Gosset’s T distribution and T confidence intervals.

where the estimates are plus or minus a quantile from the T quantile multiplied by the estimated standard error. The only difference is that we have replaced the Z quantile with a T quantile.

The T distribution has heavier tails compared to the normal distribution, resulting in slightly wider intervals. These intervals are extremely useful in statistics, and when you have the option to choose between a T interval and a Z interval for cases where both are available, it is advisable to select the T interval. As you gather more data, the T interval gradually becomes more similar to the Z interval. We will cover the single and two-group versions of the T interval. Additional T intervals that are valuable will be discussed in our regression class. The T distribution was developed by William Gosset, who published his work under the pseudonym “Student” in 1908. Since Gosset worked for the Guinness Brewery, they did not allow him to publish under his real name.

Unlike the normal distribution, which is characterized by two parameters (mean,variance), the T distribution is primarily centered around **zero** with a standardized formula for the scale. It is indexed by a single parameter known as degrees of freedom. As the degrees of freedom increase, the T distribution becomes more similar to the standard normal distribution. The reason for the T distribution is that when we divide the difference between the sample mean () and the population mean by the estimated standard error for independent and identically distributed (iid) Gaussian data, the resulting distribution is not Gaussian.

If we replaced the estimated standard error () with the true standard deviation (), the distribution would be exactly standard normal. However, when using the estimated standard error, the distribution follows a T distribution with degrees of freedom. This distinction becomes less significant as the sample size () increases, but for small sample sizes, the difference can be substantial. Using the standard normal distribution for small sample sizes can lead to narrow confidence intervals. The formula for the T interval:

The following graph shows t distribution overlaid the normal distribution for various degrees of freedom using R Studios manipulate function.



Figure : Figure 1.1: T distribution overlaid on normal distribution

k <- 1000  
xvals <- seq(-5, 5, length = k)  
myplot <- function(df){  
 d <- data.frame(y = c(dnorm(xvals), dt(xvals, df)),  
 x = xvals,  
 dist = factor(rep(c("Normal", "T"), c(k,k))))  
 g <- ggplot(d, aes(x = x, y = y))   
 g <- g + geom\_line(size = 2, aes(colour = dist))  
 g  
}  
manipulate(myplot(mu), mu = slider(1, 20, step = 1))

As the degrees of freedom decrease, the T distribution exhibits heavier tails compared to the normal distribution. However, in the plot, it may be challenging to observe the impact clearly as the focus is on the peak region where the distributions are similar. To illustrate the quantiles, we also plotted the t distribution quantiles against the normal distribution quantiles, starting at the 50th percentile.



Figure : Figure 1.3: T distribution quantiles overlaid on normal distribution quantiles

pvals <- seq(.5, .99, by = .01)  
myplot2 <- function(df){  
 d <- data.frame(n= qnorm(pvals),t=qt(pvals, df),  
 p = pvals)  
 g <- ggplot(d, aes(x= n, y = t))  
 g <- g + geom\_abline(size = 2, col = "lightblue")  
 g <- g + geom\_line(size = 2, col = "black")  
 g <- g + geom\_vline(xintercept = qnorm(0.975))  
 g <- g + geom\_hline(yintercept = qt(0.975, df))  
 g  
}  
manipulate(myplot2(df), df = slider(1, 20, step = 1))

The plot includes reference lines for the 97.5th quantile, which is typically around 1.96 for the standard normal distribution but can be considerably larger for the T distribution. For instance, with two degrees of freedom, the T quantile exceeds four. However, having only three data points to estimate the variance (n-1 degrees of freedom) is not ideal. When we increase the degrees of freedom to 20, the T quantiles become much closer to the normal quantiles. The light blue reference line represents the identity line, and deviations from this line demonstrate the distinction between the two intervals. The T interval will yield a quantile slightly larger than two, while the Z interval will yield a quantile slightly smaller than two. This discrepancy can have a notable impact on the intervals and even determine whether the interval includes zero or not. Hence, we opt for the T distribution in such cases.

In summary, the T interval is wider than the normal interval due to the additional parameter we estimate, the standard deviation. The T interval assumes the data are independent and identically distributed (iid) and approximately symmetric and mound-shaped. It is also applicable for paired observations, such as measurements taken at different times on the same units, by considering the differences or differences on the logarithmic scale. As the degrees of freedom increase, the T quantiles approach those of the standard normal distribution. Therefore, it is advisable to use the T interval rather than selecting between the T and normal intervals. For skewed distributions, the assumptions of the T interval are violated, and alternative procedures like working on the log scale or using bootstrap confidence intervals may be more appropriate. Lastly, for highly discrete data, such as binary or Poisson data, other intervals are available and preferable to the T interval.

In R, if you type data(sleep), it will load the sleep data set, which was originally analyzed in Gosset’s Biometrika paper. The data set shows the increase in hours slept for patients on sleep medications. R treats the data as two groups, but we will treat it as paired data. To load the data, you can use the command head(sleep) to view the first few rows of the data frame. The variable extra represents the extra hours slept, group is the group ID, and ID is the subject ID. The subjects are numbered from 1 to 10, and then the numbering repeats.



Figure : Figure 1.5: Sleep data

library(ggplot2)  
g <- ggplot(sleep, aes(x = group, y = extra, group = factor(ID)))  
g <- g + geom\_line(size = 1, aes(colour = ID)) + geom\_point(size =10, pch = 21, fill = "salmon", alpha = .5)  
g

This visualization clearly demonstrates the benefit of acknowledging that these are repeat measurements on the same subjects. If you fail to acknowledge this, you would be comparing the variation within group 1 to the variation within group 2. However, if you acknowledge the pairing, you compare the subject-specific differences across groups, where the variation in these differences is lower due to the correlation within subjects.

g1 <- sleep$extra[1 : 10]; g2 <- sleep$extra[11 : 20]  
difference <- g2 - g1  
mn <- mean(difference); s <- sd(difference); n <- 10

To calculate the differences between group 2 and group 1, we extract the first ten measurements (subjects 1-10) and the latter ten measurements (subjects 1-10 on the second medication). The vector y\_subdra represents the subtraction of group 2 minus group 1, and we calculate the mean and standard deviation of the difference. To obtain a t confidence interval, we use the formula and define n as 10. Alternatively, you can use the function t.test by passing it the two vectors and setting the argument paired to true. Another option is to use a model statement, such as outcome ~ group, paired = TRUE, data = sleep.

mn + c(-1, 1) \* qt(.975, n-1) \* s / sqrt(n)  
t.test(difference)  
t.test(g2, g1, paired = TRUE)  
t.test(extra ~ I(relevel(group, 2)), paired = TRUE, data = sleep)

rbind(  
mn + c(-1, 1) \* qt(.975, n-1) \* s / sqrt(n),  
as.vector(t.test(difference)$conf.int),  
as.vector(t.test(g2, g1, paired = TRUE)$conf.int),  
as.vector(t.test(extra ~ I(relevel(group, 2)), paired = TRUE, data = sleep)$conf.int)  
)

We formatted the results into a matrix for better readability. The output provides similar results from these commands, indicating that the difference in means between the groups is between 0.7 and 2.46. Since this is a confidence interval, we can interpret it as follows: if we were to repeatedly perform this procedure on independent samples, about 95% of the intervals obtained would contain the true mean difference we are estimating. This assumes that the subjects are a relevant sample from the population of interest.

### 3.1.1 Independent group T intervals

Suppose we want to compare the mean blood pressure between two groups in a randomized trial: the treatment group and the placebo group. This scenario is similar to A/B testing, commonly used in data science. In both A/B testing and randomized trials, randomization is performed to balance unobserved covariates that may affect the results. Since randomization has been conducted, it is reasonable to compare the two groups using a t confidence interval or a t test. However, we cannot use a paired t test in this case because there is no matching of subjects between the two groups. Therefore, we will discuss methods for comparing independent groups.

The standard confidence interval for comparing independent groups is calculated as follows:

The degrees of freedom (df) are determined by , where is the number of observations in group and is the number of observations in group . The standard error of the difference is given by , where is the pooled standard deviation.

The pooled standard deviation is the square root of the pooled variance. It is an estimate of the common variance if we assume that the variances in the two groups are equal due to randomization. The pooled variance is a weighted average of the variances from each group, with the weights determined by the sample sizes. If the sample sizes are equal, the pooled variance is the simple average of the variances.

Note: This interval assumes a constant variance across the two groups. If this assumption is violated, the interval may not provide accurate coverage. In such cases, alternative approaches accounting for different variances per group should be considered.

Here is an example from Rosner’s “Fundamentals of Biostatistics” book, we compare 8 oral contraceptive users to 21 controls regarding blood pressure. The average systolic blood pressure for contraceptive users is 133 mmHg with a standard deviation of 15, while the control group has an average blood pressure of 127 mmHg with a standard deviation of 18.

To manually construct the independent group interval, we calculate the pooled standard deviation by taking the square root of the weighted average of the variances. The weights are determined by the sample sizes and adjusted for degrees of freedom. We then compute the interval. In this specific example, the interval ranges from negative 10 to 20. Since the interval contains zero, we cannot rule out the possibility of the population difference being zero.

sp <- sqrt((7 \* 15.34^2 + 20 \* 18.23^2) / (8 + 21 - 2))  
132.86 - 127.44 + c(-1, 1) \* qt(.975, 27) \* sp \* (1 / 8 + 1 / 21)^.5

Aa another example we’ll revisit the sleep patients example, but this time let’s assume that the subjects were not matched. In this case, we have and as the sample sizes for group 1 and group 2, respectively. Both of these sample sizes will be 10 in this example.

We begin by constructing the pooled standard deviation estimate, calculating the mean difference, and determining the standard error of the mean difference. Then, we manually construct the confidence interval. Next, we use the t.test function to perform the t-test, specifying paired equals FALSE to indicate that the samples are not paired, and var.equal equals TRUE to assume equal variances in the two groups. We extract the confidence interval from the t.test results.

n1 <- length(g1); n2 <- length(g2)  
sp <- sqrt( ((n1 - 1) \* sd(g1)^2 + (n2-1) \* sd(g2)^2) / (n1 + n2-2))  
md <- mean(g2) - mean(g1)  
semd <- sp \* sqrt(1 / n1 + 1/n2)  
rbind(  
md + c(-1, 1) \* qt(.975, n1 + n2 - 2) \* semd,   
t.test(g2, g1, paired = FALSE, var.equal = TRUE)$conf,  
t.test(g2, g1, paired = TRUE)$conf  
)

Comparing the results of the manual calculation and the t.test, we find that they agree perfectly. However, the interval obtained when considering the pairing of subjects is entirely above 0, whereas the interval obtained without considering pairing contains 0. The plot of the data clearly illustrates why this is the case. When comparing the variability between the two groups, there is significant variability. However, when accounting for pairing and considering the variability in the differences within each subject, a substantial portion of the variability is explained by inter-subject differences.

Example: We will use the ChickWeight dataset in R, which contains weight measurements of chicks from birth to a few weeks later. To access the dataset, you can load the datasets package and use the command data(ChickWeight). To work with the data, the reshape2 package is recommended.

The ChickWeight data is initially in a long format, where the chicks are arranged in a long vector. If you want to convert it to a wide format, where each time point has its own column, you can use the dcast function from the reshape2 package. By applying dcast to the ChickWeight data frame, with Diet and Chick as the variables that remain the same, and Time as the variable to be converted from long to wide format, you can reshape the data. If you prefer different column names, you can rename them accordingly.

library(datasets); data(ChickWeight); library(reshape2)  
##define weight gain or loss  
wideCW <- dcast(ChickWeight, Diet + Chick ~ Time, value.var = "weight")  
names(wideCW)[-(1 : 2)] <- paste("time", names(wideCW)[-(1 : 2)], sep = "")  
library(dplyr)  
wideCW <- mutate(wideCW,  
 gain = time21 - time0  
)

Furthermore, you may want to create a specific variable that represents the total weight gain from time zero in the dataset. We utilized the dplyr package for data manipulation. First, we used the mutate command to create a new variable in the data frame. This variable represents the change in weight, calculated as the final time point weight minus the baseline weight. From this point onward, we will analyze the change in weight variable. Before conducting the test, let’s examine the data visually.



Figure : Figure 1.11: ChickWeight data

g <- ggplot(ChickWeight, aes(x = Time, y = weight,   
 colour = Diet, group = Chick))  
g <- g + geom\_line()  
g <- g + stat\_summary(aes(group = 1), geom = "line", fun.y = mean, size = 1, col = "black")  
g <- g + facet\_grid(. ~ Diet)  
g

We created a spaghetti plot using the ggplot2 package, which displays the weight measurements for each of the four diets over time. Each line represents a different diet, starting from the baseline and ending at the final time point. It appears that there are some noticeable differences, particularly regarding the variability between the diets. However, due to varying sample sizes, it can be challenging to make definitive conclusions. We included a reference line representing the mean for each diet. Without conducting a formal statistical test, it seems that the average weight gain for the first diet is slightly slower than that of the fourth diet. To confirm this observation, proceed with a formal confidence interval analysis. Instead of plotting individual measurements, we created a violin plot to compare the end weight minus the baseline weight for diets one and four. We will be comparing these two violin plots. The assumption of equal variances seems questionable in this case. To perform a t-test, we need the explanatory variable to have only two levels. To address this, we used the subset command to filter the data, including only records where the diet variable is one or four, excluding diets two and three. Please note that if you conduct this analysis on your own, you may want to compare all possible combinations (e.g., one to two, one to three, one to four, etc.), and adjust for multiplicity if necessary.



Figure : Figure 1.13: ChickWeight data

g <- ggplot(wideCW, aes(x = factor(Diet), y = gain, fill = factor(Diet)))  
g <- g + geom\_violin(col = "black", size = 2)  
g

Next, we used the t.test function to calculate the confidence interval. Since the vectors for diet one and diet four have different lengths, the paired equals TRUE option is not available. We compared the assumption of equal variances versus the assumption of unequal variances. The resulting intervals differ, but both indicate that weight gain on diet one is lower than on diet four. The first interval is -108 to -14, and the second is -104 to -18, with both intervals entirely below zero. However, whether the specific interval change is substantial or not depends on the dataset, which we don’t have enough information about.

wideCW14 <- subset(wideCW, Diet %in% c(1, 4))  
rbind(  
t.test(gain ~ Diet, paired = FALSE, var.equal = TRUE, data = wideCW14)$conf,  
t.test(gain ~ Diet, paired = FALSE, var.equal = FALSE, data = wideCW14)$conf  
)

### 3.1.2 A note on unequal variance

Hopefully the formula for the case of unequal variances seems familiar to you.

It involves calculating the difference in means and adding or subtracting a t quantile times the standard error. The standard error is calculated assuming different variances in each of the two groups. However, it’s important to note that if the and observations are independent and identically distributed (iid) normal, potentially with different means and variances, the relevant normalized statistic does not follow a t distribution exactly. Instead, it can be approximated by a t distribution with a specific formula for degrees of freedom.

While the degrees of freedom calculation may seem unusual because it doesn’t involve sample sizes, it relies on estimated standard deviations and variances from the two groups. Despite this complexity, using this approach yields a t calculation that closely approximates the true distribution, even though it’s not strictly a t distribution. In practice, it’s often recommended to use the unequal variance interval when in doubt.

We demonstrate the calculations for the oral contraceptive example mentioned earlier. Going through this calculation can help you understand how to plug in the values, particularly noting that the degrees of freedom in this case are 15.04. However, typically, when we want to perform unequal variance t tests, we can simply use the t.test function in R with var.equal set to FALSE. This will conduct the relevant t test with unequal variances and provide the corresponding t quantile.

In R, t.test(..., var.equal = FALSE)

In summary we explored creating intervals using the t distribution, which are highly useful in statistics. When dealing with single or paired observations, where the differences are taken into account, the t interval provides robust intervals that are not heavily dependent on assumptions about the data distribution. However, there are cases where alternatives to the t distribution and t intervals may be preferable. For example, if the data is highly skewed, it may be beneficial to consider taking a logarithmic transformation or explore different procedures. In addition, for binary data, odds ratios can be more suitable, which we will cover in the regression class’s generalized linear model component. Similar considerations apply to count data, where we will discuss Poisson models and generalized linear models for rates in the regression class. For other special cases involving two groups, you can find further coverage in the course “Mathematical Biostatistics Boot Camp 2” on Coursera.

## 3.2 Hypothesis testing

Deciding between two hypotheses is a core activity in scientific discovery. Statistical hypothesis testing is the formal inferential framework around choosing between hypotheses. After covering confidence intervals, developing an understanding of hypothesis testing should be relatively straightforward.

Hypothesis testing involves making decisions based on data. It typically begins with a null hypothesis, denoted as , which represents the status quo or a default assumption. The null hypothesis is assumed to be true initially, and we need statistical evidence to reject it in favor of an alternative hypothesis , often referred to as the research hypothesis.

Suppose we are interested in studying sleep disordered breathing, and a respiratory disturbance index (RDI) of more than 30 events per hour indicates severe sleep disordered breathing. In a sample of 100 overweight subjects with other risk factors for sleep disordered breathing in a sleep clinic, we find that the mean RDI is 32 events per hour, with a standard deviation of 10 events per hour. We want to test whether the population mean RDI for this population is equal to 30 (our benchmark for severe sleep disordered breathing) or if it is greater than 30. We can specify the null hypothesis as and the alternative hypothesis as . In this example, we are interested in determining if the respiratory disturbance index for this population is greater than 30.

Please note that the truth can only be one of the following: either is true, or is true. As a result, there are only four possible outcomes. If the truth is , and we decide to accept , we have correctly accepted the null hypothesis. If the truth is , but we decide to reject and accept , we have made a **Type I error**. In the hypothesis testing framework we will present, we aim to control the probability of Type I error to be small. On the other hand, if the truth is , and we correctly reject and accept , we have correctly rejected the null hypothesis. However, if the truth is , but we mistakenly accept and reject , we have made a **Type II error**. *The rates of Type I and Type II errors are related, meaning that as the Type I error rate decreases, the Type II error rate tends to increase, and vice versa.*

| Truth | Decide | Result |
| --- | --- | --- |
|  |  | Correctly accept null |
|  |  | Type I error |
|  |  | Correctly reject null |
|  |  | Type II error |

An analogy that can help illustrate this is a court of law. In most courts, the null hypothesis is that the defendant is innocent until proven guilty. Rejecting the null hypothesis in this case would mean convicting the defendant. We require evidence and set a standard for that evidence to reject the null hypothesis and convict someone. If we set a very low standard, meaning we don’t require much evidence to convict, we may increase the percentage of innocent people wrongly convicted (Type I errors). However, we would also increase the percentage of guilty people correctly convicted. On the other hand, if we set a very high standard, such as requiring irrefutable evidence to convict, we may increase the percentage of innocent people correctly acquitted (a good thing), but we would also increase the percentage of guilty people wrongly acquitted (Type II errors). This example demonstrates the relationship between Type I and Type II error rates.

Ideally, we aim to gather better evidence for a given standard, such as increasing the sample size.

Let’s revisit the respiratory disturbance index example and consider a reasonable testing strategy. We can reject the null hypothesis if our sample mean respiratory disturbance index is larger than a certain constant, denoted as . This constant takes into account the variability of the sample mean . Typically, is chosen such that the probability of a Type I error (rejecting the null when it is true) is low. In hypothesis testing, a common benchmark for the Type I error rate is 5%.

To determine the appropriate constant , we need to consider the standard error of the mean, which is 10 (assumed standard deviation of the population) divided by the square root of the sample size, which is 100 in this case, resulting in a value of 1.

Under the null hypothesis , the distribution of the sample mean follows a normal distribution with a mean of 30 and a variance of 1 (calculated from the standard error squared).

We want to choose the constant such that is 5%. The 95th percentile of the standard normal distribution corresponds to 1.645 standard deviations from the mean. Therefore, setting as 1.645 standard deviations from the mean under the null hypothesis will achieve the desired probability of 5%. In this case:

To summarize, the rule is to reject the null hypothesis if the observed sample mean is larger than 31.645. This rule ensures that we will reject the null hypothesis 5% of the time when the null hypothesis is true, assuming a sample size of 100 and a population standard deviation of 10. Instead of calculating in the original units of the data, it is common to convert the sample mean into standard error units from the hypothesized mean. For example, if the observed sample mean is 32, the hypothesized mean is 30, and the standard error is 2, which is greater than 1.645, the chance of this occurring is less than 5%. Therefore, we would reject the null hypothesis in favor of the alternative hypothesis.

To summarize the rule, we reject the null hypothesis when (X bar - hypothesized mean) divided by the standard error of the mean is greater than the appropriate upper quantile that leaves Alpha percent in the upper tail.

### 3.2.1 T tests

Suppose the sample size is now 16 instead of 100. The test statistic remains the same, which is calculated as the sample mean minus the hypothesized mean divided by the standard error of the mean. However, now we have a square root of 16 instead of the square root of 100, and the standard deviation is still 10.

In this case, the test statistic follows a t-distribution with 15 degrees of freedom. Under the null hypothesis, the probability of the test statistic being larger than the percentile of the T-distribution is 5%. To calculate this percentile, we can use the function qt(0.95, 15), which gives a value of 1.7531. If we plug in the values of s = 10 and x bar = 32 into the test statistic formula, we get a test statistic value of 0.8. Since 0.8 is smaller than 1.7531, we fail to reject the null hypothesis. For two-sided test, we want to reject the null hypothesis if the mean is different from 30, regardless of whether it is too large or too small. To achieve a two-sided test, we need to split the probability equally in both tails of the distribution, resulting in a 2.5% probability in each tail.

We can calculate the critical values for the two-sided test by using the function qt(0.975, 15) and qt(0.025, 15). These values represent the 2.5th percentile and 97.5th percentile of the T-distribution with 15 degrees of freedom. If the test statistic is larger than the positive critical value or smaller than the negative critical value, we reject the null hypothesis. Alternatively, we can take the absolute value of the test statistic and reject it if it is larger than the positive critical value. In our example, since the test statistic is positive, we only need to consider the larger side. Since 0.8 is smaller than the positive critical value , we fail to reject the null hypothesis in the two-sided test. In practice, instead of manually calculating the rejection region and performing hypothesis tests, it is common to use statistical functions like t.test() in R. These functions provide all the relevant statistics, including the test statistic, degrees of freedom, and p-values. They also offer the option to perform paired tests when appropriate.

As an illustration, we can use the t.test() function in R with the father.son dataset to test whether the population mean of son’s heights is equivalent to the population mean of father’s heights.

library(UsingR); data(father.son)  
t.test(father.son$sheight - father.son$fheight)

Since the observations are paired, we can pass the difference between son’s and father’s heights directly to the function or specify paired = TRUE when passing the individual vectors. The t.test() function provides the t statistic, degrees of freedom, and automatically calculates the t confidence interval. By analyzing the output, we can determine the statistical significance and evaluate the practical significance of the results by examining the range of values in the confidence interval. ### Confidence intervals and hypothesis tests

The t.test() function conveniently provides the t confidence interval automatically. It is valuable to examine the confidence interval alongside the test output as it helps bridge the gap between statistical significance and practical significance. By assessing the range of values in the confidence interval, which is expressed in the units of the data of interest, we can determine whether they are practically meaningful or not. In the previous section in the book on confidence intervals, we explored whether a hypothesized mean was supported by checking if it fell within the confidence interval. Similarly, we performed a hypothesis test to determine if the mean was equal to a specific value or not. **It turns out that these two procedures do not disagree.** Checking whether the hypothesized mean falls within the interval is equivalent to conducting a two-sided hypothesis test, with the caveat that the significance level used for the interval should be equal to where is the significance level used for the hypothesis test. In other words, if we construct a 95% confidence interval and check whether is within that interval, we fail to reject the null hypothesis if it is inside the interval and reject it if it is outside. This procedure aligns with performing the hypothesis test at a significance level of . This relationship is stated in the slide, where the confidence interval can be seen as the set of all possible values for which we fail to reject the null hypothesis.

With the understanding of confidence intervals and hypothesis tests for one group in place, extending these concepts to two groups is a straightforward extension. The rejection rules remain the same, and now we want to test whether the means of two groups are equal or not. We have the same set of alternatives: , , or . The test statistic remains the same, calculated as the difference between the sample means minus the hypothesized mean difference , divided by the standard error of the mean.

Example: Take the ChickWeight dataset. We can load the dataset using library(datasets) and data(ChickWeight). To reshape the data into the desired format, we may need to use the reshape2 package. The ChickWeight dataset contains measurements for different chicks at various time points, represented in a long format. We desired a wide format for our data, so we used the dcast() function to achieve that. Additionally, we renamed the resulting dataset and defined a new variable called weight\_gain, which represents the difference between time21 and time0. In this particular dataset, most chicks had nearly identical weights at time0 relative to time21. Although this doesn’t significantly affect the results, we wanted to demonstrate the use of the mutate() function, which simplifies adding variables to a data frame.

To conduct the t-test, we selected a subset of the data where the diet was either 1 or 4. This was necessary because the tilde operator in the t.test() function requires the predictor variable Diet to have exactly two levels when comparing groups. By setting paired = FALSE, we indicate that the chicks receiving diet 1 and diet 4 are completely separate groups, and there is no pairing between them. Chick 1 from diet 1 has no connection to chick 1 from diet 4.

library(datasets); data(ChickWeight); library(reshape2)  
##define weight gain or loss  
wideCW <- dcast(ChickWeight, Diet + Chick ~ Time, value.var = "weight")  
names(wideCW)[-(1 : 2)] <- paste("time", names(wideCW)[-(1 : 2)], sep = "")  
library(dplyr)  
wideCW <- mutate(wideCW,  
 gain = time21 - time0  
)

We discussed how assuming equal variances may not be the best approach for this dataset. Therefore, we suggest trying the example with var.equal = FALSE to observe how the results change.

wideCW14 <- subset(wideCW, Diet %in% c(1, 4))  
t.test(gain ~ Diet, paired = FALSE,   
 var.equal = TRUE, data = wideCW14)

The resulting t statistic represents the estimate of the difference in average weight gain between the two diets, minus the hypothesized value of 0. When comparing two groups, unless a specific hypothesized difference in means is specified, the default assumption is that we are testing whether the means are equal under the null hypothesis or different under the alternative. The degrees of freedom are calculated as , which we covered in the confidence interval section. If unequal variances are used, fractional degrees of freedom may be obtained. The output also provides the confidence interval, which is always useful to examine when performing a hypothesis test. In the following section, we will discuss the concept of p-values and how they facilitate hypothesis testing. The calculated T statistic of -2.7 indicates how many estimated standard errors the difference in means is from the hypothesized mean. Since it is far into the tail of the t-distribution or normal distribution, it falls well below our cutoff value. Although we don’t explicitly determine a cutoff value in this case, we can immediately conclude, after learning about p-values, that this result would be rejected in a 5% level test without calculating the specific t quantiles.

Example: Suppose you have a friend who has eight children, with seven of them being girls. You want to evaluate whether this supports the belief that the genders are independent and equally likely (like a fair coin flip). You want to test the null hypothesis that the probability of having a girl is 0.5 against the alternative hypothesis that it is greater than 0.5 ( vs. ), as you are slightly skeptical.

To determine the number of girls the couple could have for the probability of having that many or more to be less than 5% under the null hypothesis of a fair coin, we can set up a rejection region. However, if we consider a rejection region from zero to eight girls, we would always reject the null hypothesis.

| Rejection region | Type I error rate |
| --- | --- |
| [0 : 8] | 1 |
| [1 : 8] | 0.9961 |
| [2 : 8] | 0.9648 |
| [3 : 8] | 0.8555 |
| [4 : 8] | 0.6367 |
| [5 : 8] | 0.3633 |
| [6 : 8] | 0.1445 |
| [7 : 8] | 0.0352 |
| [8 : 8] | 0.0039 |

If we set up a rejection region where we would reject the null hypothesis if the couple had one to eight girls, we still wouldn’t achieve a 5% significance level. It would be nearly one, indicating a very high rejection rate. However, if we consider having seven or eight girls as the rejection region, the probability of rejecting under the null hypothesis is just under 5%. It is worth noting that we cannot achieve an exact 5% level test in this case due to the discrete nature of the binomial distribution. For larger sample sizes, a normal approximation could have been used by treating the coin flip outcomes as averages and assuming a Gaussian distribution. However, you already know how to handle that. In this specific test, we observe that the closest rejection region consists of having seven or eight girls. Since your friend had seven girls, we would reject the null hypothesis based on this observation. However, it’s important to acknowledge that an exact 5% level test is not feasible in this case due to the discrete nature of the binomial distribution. For two-sided tests, the approach is not obvious. We will discuss a method for conducting two-sided tests in the next lecture, and we believe it will become clearer when we introduce the concept of p-values. If this example seems confusing, the lecture on p-values will help clarify it. The exact binomial or Poisson tests will become easier to comprehend.

If you can perform a two-sided test for a binomial or Poisson distribution, you can also invert those tests. By considering the values for which you would fail to reject the null hypothesis, you can generate exact confidence intervals for the binomial and Poisson parameters. This is precisely how R calculates exact binomial intervals, without relying on asymptotic or central limit theorem approximations. They invert a two-sided hypothesis test of this nature.

## 3.3 P values

P-values are a convenient way to communicate the results of a hypothesis test. When communicating a P-value, the reader can perform the test at whatever Type I error rate that they would like. Just compare the P-value to the desired Type I error rate and if the P-value is smaller, reject the null hypothesis. Formally, the P-value is the probability of getting data as or more extreme than the observed data in favor of the alternative. The probability calculation is done assuming that the null is true. In other words if we get a very large T statistic the P-value answers the question “How likely would it be to get a statistic this large or larger if the null was actually true?”. If the answer to that question is “very unlikely”, in other words the P-value is very small, then it sheds doubt on the null being true, since you actually observed a statistic that extreme. P-values are widely used as a measure of statistical significance. Almost every statistical software that performs hypothesis tests provides a p-value as an output. However, due to their popularity and frequent misinterpretation, p-values have become a subject of controversy among statisticians. Our main goal is to understand how to generate p-values correctly and interpret them appropriately.

The fundamental concept of a p-value is to start by assuming the null hypothesis, which assumes no effect or relationship, and then calculate the probability of obtaining evidence as extreme or more extreme than the evidence observed under this null hypothesis. In other words, we assess how unusual our observed result is if the null hypothesis were true. First we will follow a simple three-step approach, and then we will delve into the calculations.

Firstly, we establish the hypothetical distribution of a summary statistic, often referred to as a test statistic, such as the t-statistic from the t-test lecture, assuming no effect or relationship (null distribution). Secondly, we calculate the test statistic using the actual data we have. For example, in the case of a t-test, we substitute the empirical mean, subtract the hypothesized mean, and divide by the standard error. Finally, we determine the probability of obtaining a test statistic as extreme or more extreme than the one calculated. In other words, we compare our calculated test statistic to the hypothetical distribution and assess its level of extremity towards the alternative hypothesis. If the p-value is small, it indicates that the probability of observing a test statistic as extreme as the one we observed is low under the assumption that the null hypothesis is true.

Formally, the p-value is the probability, under the null hypothesis, of obtaining evidence as extreme or more extreme than what was actually observed. Typically, this evidence refers to the test statistic. Therefore, the p-value represents the probability of obtaining a test statistic as extreme or more extreme in favor of the alternative hypothesis than the observed test statistic. If the p-value is small, it suggests that either the null hypothesis is true, and we have observed something highly supportive of the alternative that is unlikely to occur, or the null hypothesis itself is false.

Example: Suppose we want to test the null hypothesis versus the alternative hypothesis . If our calculated t-statistic is 2.5 with 15 degrees of freedom, we can determine the probability of obtaining a t-statistic as large as 2.5 in this scenario. By calculating pt(2.5, 15, lower.tail = false), we find that the probability is approximately 1%. Therefore, the probability of observing evidence as extreme or more extreme than what was actually obtained under the null hypothesis is 1%. This suggests that either the null hypothesis is true and we have observed an unusually large test statistic, or the null hypothesis is false. Another way to interpret the p-value is as the attained significance level.

Example: Imagine our test statistic is 2 for the null hypothesis versus the alternative hypothesis . Test statistics larger than 2 provide stronger evidence in favor of the alternative hypothesis, where 2 represents two standard errors above the hypothesized mean of 30. Assuming our test statistic follows a standard normal distribution instead of a t-distribution for simplicity, if we set alpha to 0.05, we would reject the null hypothesis because the test statistic lies above the critical value of 1.645 corresponding to an alpha of 0.05.

Now, imagine if we set alpha to 0.04, resulting in a slightly closer critical value than 1.645. What if we found the exact error rate where the critical value aligns exactly with 2? That would be equivalent to calculating the probability of obtaining a test statistic as large or larger than 2 under the null hypothesis, which is nothing other than the p-value we calculated. In essence, the p-value represents the smallest alpha level for which we would still reject the null hypothesis. Hence, it is referred to as the attained significance level. The advantage of the p-value is that it provides a convenient test statistic that can be interpreted by others. When you report a p-value, the reader or recipient can perform the hypothesis test at any alpha level they choose. The simple rule is that if the p-value is less than the chosen alpha level, the null hypothesis is rejected. If the p-value is greater than the alpha level, the null hypothesis is not rejected. For one-sided hypothesis tests using t-tests or z-tests, the calculated p-value already accounts for evidence as extreme or more extreme in one direction. However, for two-sided hypothesis tests, where evidence in both tails is considered equally probable, the p-value needs to be doubled.

Note: Most statistical software automatically interprets the p-value as a two-sided test for most cases. If it’s not explicitly mentioned, the calculated p-value is for the two-sided test. Additionally, in more advanced statistics classes covering tests like the chi-squared test, the calculated p-values are inherently two-sided, and there’s no need to double them.

Previously, we discussed an example that illustrates the concept of p-values, and we would like to revisit it now that we have a better understanding. Suppose you have a friend who has had seven girls out of eight kids, and you want to determine the probability that the coin lands on a girl, denoted by p. We are interested in testing whether p is equal to 0.5 or greater than 0.5, with the null hypothesis and the alternative hypothesis .

Under the null hypothesis, we calculate the probability of obtaining evidence as extreme or more extreme. In this case, the most logical test statistic is the count of girls out of eight. The p-value calculation involves considering the binomial probability of observing seven or eight girls, assuming . This calculation yields a p-value of approximately 3.5%. Alternatively, you can use the pbinom function to directly calculate the p-value, resulting in the same value. If we were conducting a hypothesis test, we would reject the null hypothesis at a 5% level and also at a 4% level. However, we would not reject it at a type 1 error rate of 3%.

choose(8, 7) \* .5 ^ 8 + choose(8, 8) \* .5 ^ 8   
pbinom(6, size = 8, prob = .5, lower.tail = FALSE)

In this specific problem, the calculation of the two-sided p-value is not obvious. To address this, a simple trick is to calculate the two one-sided p-values. For instance, the probability of having seven or more girls represents one-sided p-value, and the probability of having seven or fewer girls represents the other one-sided p-value. Taking the smaller of these two p-values and doubling it gives us the two-sided p-value for binomial exact calculations.

Example(Poisson): Imagine a hospital that has an infection rate of 10 infections per 100 person-days at risk, equivalent to a rate of 0.1 infections per person-day at risk during the last monitoring period. The hospital considers a rate of 0.05 infections per person-day at risk as an important benchmark, and they would implement quality control procedures if the rate exceeds this threshold. However, they don’t want to trigger these procedures based on random fluctuations alone. To formally test this hypothesis, accounting for the data’s uncertainty, we assume that the count of infections follows a Poisson distribution.

The null hypothesis states that (the rate) is 0.05, while the alternative hypothesis suggests that is greater than 0.05. In this case, we want to determine the probability of observing 10 or more infections if the true infection rate for 100 person-days at risk is 5. Using the ppois function in R, we calculate the upper tail probability. Due to a quirk in R’s syntax, we need to input 9 instead of 10 as the value and set lower.tail = FALSE to obtain the probability of strictly greater than 9.

ppois(9, lambda = 5, lower.tail = FALSE)

The resulting p-value indicates the probability of obtaining 10 or more infections when the true rate is 5 for 100 person-days at risk. In this example, the probability is approximately 3%, suggesting a relatively low likelihood of observing as many as 10 infections for 100 person-days at risk. Thus, the hospital may consider implementing quality control procedures.

To summarize, the calculation of a p-value involves determining the probability of obtaining data as extreme or more extreme than what was observed in favor of the alternative hypothesis, with the probability calculation performed under the null hypothesis. This approach applies to all p-values, and we have explored the formal rules for executing z-tests and t-tests, as well as examples involving the binomial distribution and the Poisson distribution.

## 3.4 Knitr

In the following, we give you some information about knitr if you are interested you can checkout, Roger Peng’s course on Reproducible Research.

Open RStudio and navigate to “File” (Alt-F) and select “New File.” You’ll see various options, and you should choose “R Markdown.” This will generate a simple Knitr document for you. Feel free to edit the title; for example, you can rename it to “Test Knitr Document.” In the document, you’ll notice R commands and some formatting. To execute R code within the document, you need to use a set of three backticks or quotation marks (usually found below the Escape key on the keyboard). After the initial backticks, add an “r” to indicate that you’re using R code, followed by a comma to open up additional options.

Knitr offers numerous options, but we will highlight a few essential ones. The cache option determines whether R should store the code’s results. Another useful option is eval, which specifies whether the code should be evaluated or simply displayed in the document. You can choose to display code with results or hide code using the results option. Additionally, the echo option controls whether the code is displayed or not. For example, the document includes a code snippet demonstrating a plot using the command plot(cars). Feel free to modify and add your own code snippets.



Figure : Figure 2.4: Knitr rendered Rmd file

Once you’ve made your changes, save the document. Give it a name like “test.Rmd.” To knit the document into an HTML format, you can either use the Knit HTML button in the toolbar or go to Code and select “Knit Document.” This will create an HTML document displaying the code, results, and any additional content.

If you want to view the HTML document, you can find it in the working directory. In R, you can use the dir() function to see the files, and then use the browseURL() function to open the HTML document in a web browser. For example, browseURL("test.html") will open the HTML document named “test.html.”

That’s the basic process of using Knitr in a nutshell. Feel free to explore the additional options and customization features to create dynamic and interactive documents.

## 3.5 Practical R Exercises in swirl

During this week of the course you should complete the following lessons in the Statistical Inference swirl course:

1. T Confidence Intervals
2. Hypothesis Testing
3. P Values

# 4 Power, Bootstrapping, & Permutation Tests

We’ve talked about a Type I error, rejecting the null hypothesis when it’s true. We’ve structured our hypothesis test so that the probability of this happening is small. The other kind of error we could make is to fail to reject when the alternative is true (**Type II error**). Or we might think about the probability of rejecting the null when it is false. This is called **Power = 1 - Type II error**. We don’t have as much control over this probability, since we’ve spent all of our flexibility guaranteeing that the Type I error rate is small. One avenue for the control of power is at the design phase. There, assuming our finances let us, we can pick a large enough sample size so that we’d be likely to reject if the alternative is true. Thus the most frequent use of power is to help us design studies.

## 4.1 Power

Power refers to the probability of rejecting the null hypothesis when it is actually false. As the name suggests, having more power is desirable. Interestingly, power becomes more significant when we fail to reject the null hypothesis than when we do reject it! This is because when we reject the null hypothesis, we are confident that the alternative hypothesis is true. However, when we fail to reject the null hypothesis, we are unsure whether the null hypothesis is true or if we simply failed to detect the alternative hypothesis. Therefore, power is more relevant when we fail to reject the null hypothesis.

Imagine conducting a study to compare Treatment A and Treatment B, where you only randomized three individuals to each treatment. In this case, if you find no significant difference between the two treatments, it wouldn’t be surprising because the small sample size limits your power to detect meaningful differences. The null result is somewhat expected due to the limited data. On the other hand, if you had 300 individuals in each treatment group and still failed to reject the null hypothesis, it would be more meaningful because with such a large sample size, you would expect to observe a difference if it truly existed. Therefore, power comes into play more prominently with null results rather than non-null results. Power is commonly considered during the study design phase. You want to design your study in a way that provides a reasonable chance of detecting the alternative hypothesis if it is true. This involves carefully selecting factors like sample size, statistical tests, and effect sizes.

A type II error, also known as a **beta error**, is the failure to reject a false null hypothesis. It is undesirable because it means missing a true effect or relationship. Power is simply , representing the probability of correctly rejecting the null hypothesis. In hypothesis testing, the two crucial quantities are the type I error rate () and the type II error rate (). However, we often discuss power () rather than itself.

Suppose we are interested in testing the mean respiratory disturbance index (RDI) in a specific population of obese subjects. We want to compare whether the mean RDI () . We calculate a t-statistic, which follows a t-distribution under the null hypothesis assumption. By calculating the probability of obtaining a t-statistic larger than the upper quantile of the t-distribution at the significance level , we determine the probability of rejecting the null hypothesis under the assumption that . This probability corresponds to .

Power, on the other hand, is calculated similarly but with the alternative hypothesis in mind. Instead of plugging in , we use a value greater than 30. Power represents the probability of correctly rejecting the null hypothesis when the true mean (). If we consider a larger , such as 60, our power will increase because we’ll have a higher chance of detecting the difference. Conversely, if the true alternative mean is very close to the null value, say 30.00001, our power will be lower since it is more challenging to detect such a small difference. Thus, power is a function that depends on the mean under the null hypothesis. Values close to the null mean will resemble the type I error rate, while values far from it will yield higher power, potentially approaching 100%.

### 4.1.1 Calculating Power

Assume a sample with normal distribution. This assumption is true either because we have a large sample size and can apply the central limit theorem, or we can simply assume that the underlying population is normally distributed. For the sake of argument, let’s assume that the sample mean () is normally distributed, and the population standard deviation () is known. In this scenario, we would reject the null hypothesis if our Z-statistic:

Alternatively, we can state that we will reject the null hypothesis if is greater than 30 plus the product of the z quantile and the standard error of the mean.

$$Under\space H\_0: \bar X \sim N(\mu\_0, \sigma^2 / n)$$

$$Under\space H\_a: \bar X \sim N(\mu\_a, \sigma^2 / n)$$

We can easily calculate the power using the R programming language. By taking the pnorm (normal probability)

alpha = 0.05  
z = qnorm(1 - alpha)  
pnorm(mu0 + z \* sigma / sqrt(n), mean = mua, sd = sigma / sqrt(n), lower.tail = FALSE)

Example: Suppose someone wants to conduct a study to test whether the mean () for a specific population is 30 or greater. They are interested in detecting a difference as large as 32, with a sample size of 16, and a known standard deviation of 4.

We can plug in these values and calculate the power. When we set , the power is 5%. However, when we set , the power increases to 64%.

mu0 = 30; mua = 32; sigma = 4; n = 16  
z = qnorm(1 - alpha)  
pnorm(mu0 + z \* sigma / sqrt(n), mean = mu0, sd = sigma / sqrt(n),   
 lower.tail = FALSE)  
pnorm(mu0 + z \* sigma / sqrt(n), mean = mua, sd = sigma / sqrt(n),   
 lower.tail = FALSE)

This means there is a 64% probability of detecting a mean as large as 32 or larger if we conduct the experiment. By plotting the power curves, which represent the power as a function of , with different sample sizes shown in different colors, we can observe interesting patterns.



Figure : Figure 1.3: Plotting the power curve

library(ggplot2)  
nseq = c(8, 16, 32, 64, 128)  
mua = seq(30, 35, by = 0.1)  
power = sapply(nseq, function(n)  
 pnorm(mu0 + z \* sigma / sqrt(n), mean = mua, sd = sigma / sqrt(n),   
 lower.tail = FALSE)  
 )  
colnames(power) <- paste("n", nseq, sep = "")  
d <- data.frame(mua, power)  
library(reshape2)  
d2 <- melt(d, id.vars = "mua")  
names(d2) <- c("mua", "n", "power")   
g <- ggplot(d2,   
 aes(x = mua, y = power, col = n)) + geom\_line(size = 2)  
g

As gets larger, the power increases, which indicates a higher likelihood of detecting a larger difference. Additionally, as the sample size increases, the curves shift upwards and reach higher power levels earlier. This aligns with our expectation that collecting more data increases the likelihood of detecting a specific effect. We can use RStudio’s manipulate function to explore power in relation to the two normal distributions. We can visually evaluate the power and observe how it changes with different values.



Figure : Figure 1.5: Plotting the power curve

library(manipulate)  
mu0 = 30  
myplot <- function(sigma, mua, n, alpha){  
 g = ggplot(data.frame(mu = c(27, 36)), aes(x = mu))  
 g = g + stat\_function(fun=dnorm, geom = "line",   
 args = list(mean = mu0, sd = sigma / sqrt(n)),   
 size = 2, col = "red")  
 g = g + stat\_function(fun=dnorm, geom = "line",   
 args = list(mean = mua, sd = sigma / sqrt(n)),   
 size = 2, col = "blue")  
 xitc = mu0 + qnorm(1 - alpha) \* sigma / sqrt(n)  
 g = g + geom\_vline(xintercept=xitc, size = 3)  
 g  
}  
manipulate(  
 myplot(sigma, mua, n, alpha),  
 sigma = slider(1, 10, step = 1, initial = 4),  
 mua = slider(30, 35, step = 1, initial = 32),  
 n = slider(1, 50, step = 1, initial = 16),  
 alpha = slider(0.01, 0.1, step = 0.01, initial = 0.05)  
 )

In this graph the parameters are currently set to the values used in the previous calculations: , , , and . The first plot illustrates the distribution of the sample mean under the null hypothesis. It is centered at 30 with a variance of . The second plot represents the alternative hypothesis, where the sample mean is centered at 32 with the same variance. We have set a critical value (shown as a black line) such that if the sample mean exceeds this threshold, we reject the null hypothesis. The black line is calibrated so that the probability of obtaining statistics larger than it is 5% if the red density is true (i.e., the null hypothesis is true). Power refers to the probability of obtaining a sample mean greater than the black line, assuming the blue curve (alternative hypothesis) is true. In other words, power represents the probability of rejecting the null hypothesis correctly. Conversely, corresponds to the type II error rate.

If we decrease the significance level (), the black line moves to the right, making it harder to reject the null hypothesis and resulting in lower power. On the other hand, increasing leads to better power but also increases the type I error rate. When we decrease (the standard deviation), the black line moves downward. With less variability in the sample mean, the probability of rejecting the null hypothesis (power) approaches 100%. Conversely, increasing decreases power because there is more noise in the measurements. Adjusting the mean under the alternative hypothesis () shifts the black line accordingly. As moves closer to 30, power decreases, and as it moves further away, power increases. Manipulating the sample size (n) affects the variance of the sample mean. A larger sample size tightens the densities, resulting in better power. Conversely, a smaller sample size leads to lower power. It is recommended to review the code for the manipulate function to gain a better understanding of power in this context. By experimenting with the various parameters, you can observe how power changes in different ways.

To recap the scenario when testing the alternative hypothesis, specifically focusing on the case where is greater than the power is in terms of (where represents the type II error rate), we can use the equation:

Here, is the probability that the sample mean is larger than , calculated under the assumption that equals (the mean under the alternative hypothesis). It’s important to note that follows a normal distribution with mean and variance . The key point to highlight is that in this equation, , , , and are the unknowns, while and usually (the specified type I error rate) are known. By specifying any three of the unknowns, you can solve for the fourth. For example, if you know the desired alternative mean (), the assumed standard deviation (), and the desired sample size (n), you can solve for power. This approach is commonly used in power calculations during trial planning. Usually, the main concerns revolve around n (sample size) or (desired power). You may want to determine the necessary sample size to achieve a specific power level, or given a fixed sample size, assess the resulting power. However, you can also solve for or depending on your requirements.

Here are some basic rules regarding power:

1. As alpha (significance level) increases, power also increases. The power of a one-sided test is greater than the power of the corresponding two-sided test.
2. The further (mean under the alternative hypothesis) is from , the higher the power.
3. As the sample size (n) increases, the sample mean has less variability, resulting in higher power.
4. When the standard deviation () decreases, the sample mean has less variability, leading to higher power.

An interesting fact about power is that it often depends on a function of these parameters rather than each parameter individually. In this case, the function is one-dimensional, so you only need to know one number to calculate power. That number is the effect size, which is the difference between the null and alternative means divided by the standard error. The effect size is unit-free, making it useful and interpretable across different problems. ### T test power We do not actually calculate power in the exact manner described in the previous section. Those were meant to help understand the concepts, assuming we knew and the data followed a Gaussian distribution or could be approximated as such due to the central limit theorem. We often use the power.t.test function in R to calculate power. We will provide some insights around t-test power before explaining how to use power.t.test.

The argument for t-test power is similar to what we did for the normal distribution case. We reject the null hypothesis if our test statistic, , exceeds a quantile instead of a z quantile since we are dealing with a t-test. However, when calculating power, this is done under the assumption that (the value under the alternative hypothesis), not . It is important to note that the statistic () does not follow a t-distribution if the true mean is not . Instead, it follows something called the non-central t-distribution, which we won’t cover here. Thus, evaluating the non-central t-distribution is necessary to calculate power.

The power.t.test function allows you to calculate power by evaluating the non-central t-distribution. Similar to before, you have some known parameters like and , and some unknown parameters like , , and . By omitting one of the unknown parameters and specifying the others, power.t.test will solve for the omitted parameter.

In the following examples, we specify n, delta (difference in means), and the standard deviation to calculate power.

power.t.test(n = 16, delta = 2 / 4, sd=1, type = "one.sample", alt = "one.sided")$power  
power.t.test(n = 16, delta = 2, sd=4, type = "one.sample", alt = "one.sided")$power  
power.t.test(n = 16, delta = 100, sd=200, type = "one.sample", alt = "one.sided")$power

The resulting power is dependent on the effect size, which is the difference between and divided by the standard deviation. The effect size is the key factor driving the power calculation.

We can also calculate sample size by providing the desired power to power.t.test.

power.t.test(power = .8, delta = 2 / 4, sd=1, type = "one.sample", alt = "one.sided")$n  
power.t.test(power = .8, delta = 2, sd=4, type = "one.sample", alt = "one.sided")$n  
power.t.test(power = .8, delta = 100, sd=200, type = "one.sample", alt = "one.sided")$n

For a given effect size, we can determine the required sample size to achieve a specific power level. It is advisable to round up the fractional sample size to the nearest integer when dealing with real data.

Throughout the examples, we observe that the calculations remain the same when specifying equivalent effect sizes, regardless of the specific values used for delta and the standard deviation.

Exercise: you can omit the delta parameter and input a specific sample size (n) into the power.t.test function. This will allow you to determine the minimum detectable delta, the smallest difference between mu0 and mu a that can be detected with 80% power using that sample size. With the code provided earlier, this extension should not be too challenging.

We recommend using power.t.test as your initial approach for power calculations. One of the main reasons behind this preference is that power calculations involve various factors and settings that can easily lead to confusion. It is common to overestimate the power or underestimate the required sample size. Therefore, when in doubt, it is advisable to simplify the power calculation as much as possible. You can try to reframe your question as a t-test or a binomial test to perform a straightforward power calculation. Although this approach may yield a slightly conservative power or sample size estimate, it provides a clear understanding of the calculation. Once you have a solid foundation, you can proceed to more complex power calculations if necessary.

## 4.2 Multiple Comparisons

This topic is being taught by Jeff Leek. Jeff is known for his contributions to the field.

This part focuses on the topic of multiple testing. We discussed hypothesis testing earlier in the course, and mentioned the importance of correcting for multiple tests to avoid deceiving ourselves. In this section, we’ll explore how to perform these corrections. The underlying idea is that hypothesis testing and significance analysis are frequently used techniques. However, they are often misused. One common practice is calculating multiple p-values when analyzing the same dataset and then only reporting the smallest p-value or considering all p-values below 0.05 as significant. This approach leads to issues as we demonstrate shortly.

Our goal is to correct for multiple testing in order to prevent false positives or false discoveries when conducting analyses involving numerous variables. There are two essential components to multiple testing corrections. First, we need to define an error measure that we want to control. Second, we require a statistical method or correction that helps control that error measure. These concepts are related to the three eras of statistics, as discussed in Brad Efron’s book. The first era focused on large-scale population description using extensive datasets. The second era developed optimal inference techniques for extracting information from small sample sizes when data collection was challenging and costly. The third and current era is characterized by scientific mass production, where data is readily available and inexpensive. However, this also means that we perform an increasing number of analyses. If we fail to correct for the fact that each analysis carries a small potential for error, these errors can accumulate.

The need for multiple testing corrections arises due to advancements in technology, which have led to a proliferation of data. These technologies span various fields, from next-generation sequencing in molecular biology to patient imaging in clinical studies and the use of electronic medical records. Additionally, personalized or individualized quantitative self-measurements, such as those obtained with devices like the Nike Fuel Band or FitBit, contribute to the increase in available data. So, why should we correct for multiple tests? Allow me to illustrate the key problem using this cartoon example from [xkcd page](https://xkcd.com/882/), where we want to investigate whether jelly beans cause acne..



Figure : <https://imgs.xkcd.com/comics/significant.png>

So what happens is you send a team of scientists to investigate the relationship between jelly beans and acne. Initially, they examine all types of jelly beans and find that the p-value is greater than 0.05, indicating no significant association. They then decide to test each color of jelly beans individually to see if any specific color is linked to acne. However, for each color tested, the p-value is still greater than 0.05, so they do not report any significant results. Finally, after testing over 20 different types of jelly beans, they discover a significant association between green jelly beans and acne. They claim that there is only a 5% chance of this association occurring by chance alone. However, considering the number of hypotheses tested (20 in this case), it becomes highly likely that at least one of them would result in a coincidental finding. If we allow for a 5% chance of error in each hypothesis test and perform at least 20 tests, we can expect to encounter at least one error, as 20 multiplied by 5% is approximately 100%. We have been using p-values and hypothesis testing as interchangeable terms, but they are not exactly the same.

To illustrate this, let’s consider a hypothesis test for a parameter, , where we want to determine if it equals zero or not. In a linear regression model, if the coefficient for a certain variable is equal to zero, it implies no association between the variables. Conversely, if it is not equal to zero, there is some association. To perform a hypothesis test, we fit the linear regression model and calculate a p-value. Then, we compare the p-value to a predetermined threshold. If the p-value is below the threshold, we reject the null hypothesis that equals zero and conclude that is not equal to zero. If the p-value is above the threshold, we fail to reject the null hypothesis and state that equals zero. This is the essence of a hypothesis test.

In the context of hypothesis testing, we have a table of possible outcomes. Each row represents a specific claim about (equal to zero or not equal to zero), while each column represents the true state of the world ( equals zero or not equal to zero). When conducting multiple hypothesis tests, situations where we claim is equal to zero but it is actually not fall into one cell, and situations where we claim is not equal to zero but it is actually equal to zero fall into another cell. Therefore, there are two types of errors that can occur.

|  |  |  |  |
| --- | --- | --- | --- |
| Claim |  |  |  |
| Claim |  |  |  |
| Claims |  |  |  |

Type I errors, or false positives, occur when we claim that beta is not equal to zero (there is a relationship), but in reality, there is no relationship. We denote the number of these errors as v. On the other hand, Type II errors, or false negatives, occur when we claim that beta is equal to zero (no relationship), but in reality, there is a relationship.

Generally, when conducting scientific investigations, people tend to be more concerned about Type I errors, or false positives. We want to minimize the number of times we are led astray or encounter false positives. However, the emphasis on the two error rates may vary depending on the nature of the problem and the relative costs associated with each type of error. In multiple testing, there are several error rates to consider, which form the first component of a multiple testing procedure.

The first error rate to consider is the false positive rate. This rate represents the frequency at which false results are deemed significant. In other words, it measures the average fraction of times we classify results as significant when they are not. This rate is calculated by dividing the number of false significant variables by the total number of non-significant variables.

Another error measure is the family-wise error rate (FWER). It quantifies the probability of having at least one false positive result. The variable V represents the count of instances where there is no relationship between variables, yet we claim that there is. By controlling the family-wise error rate, we aim to limit the probability of making one or more false claims.

The false discovery rate (FDR) is a distinct error measure that differs from the false positive rate. It captures the rate at which claims of significance are false. It calculates the ratio of expected false discoveries (E) to the number of claims of significance (R). In other words, it measures the rate at which our assertions of a relationship between variables are incorrect.

The false positive rate is closely related to the Type I error rate, but there is a subtle distinction between the two. If you want to explore this topic further, you can refer to the Wikipedia page [linked here](https://en.wikipedia.org/wiki/False_positive_rate).

The next step in multiple testing is defining a procedure that can control the specified error measure. In other words, we need a method to constrain the error rate in a particular manner. To control the false positive rate, you can utilize the calculated p-values directly. By setting a threshold alpha (between 0 and 1) and considering all p-values below this threshold as significant, you can effectively control the false positive rate at the chosen level on average.

In other words, when controlling the false positive rate, the expected rate of false positives is kept below the threshold alpha. However, there’s a problem with this approach. Let’s consider a scenario where you perform a large number of hypothesis tests, say 10,000 tests. Although this might seem extreme, it is common in high-dimensional settings or signal processing contexts. If you designate all p-values below 0.05 as significant (with alpha set to 0.05), the expected number of false positives would be 500 (10,000 tests multiplied by the false positive rate). Consequently, if you obtain 500 significant results from these tests, it is highly likely that most of them are false positives. This raises the question of how to control a different error rate to avoid an excessive number of false positives.

One option is to control the family-wise error rate, as discussed earlier. The [Bonferroni correction](https://en.wikipedia.org/wiki/Bonferroni_correction), the oldest multiple testing correction method, can be used for this purpose. The basic idea is to calculate the p-values normally and then adjust the level. By dividing the original by the number of hypothesis tests performed (e.g., divided by 10 if there are 10 tests), a new level is obtained. Any p-values below this new level are deemed significant, effectively controlling the family-wise error rate on average. This method is easy to calculate and minimizes errors, as the probability of even one false positive is kept low. However, a downside is that it can be overly conservative, especially when dealing with a large number of tests. Allowing for a few false positives might be preferred if it increases the chances of discovering more genuine signals.

**Basic idea**: \* Suppose you do tests \* You want to control FWER at level so \* Calculate P-values normally \* Set \* Call all -values less than significant

**Pros**: Easy to calculate, conservative **Cons**: May be very conservative

This is where the false discovery rate (FDR) correction comes into play. It is the most commonly used error rate correction method for multiple testing, particularly when conducting numerous hypothesis tests in genomics, imaging, astronomy, and other signal processing fields. The FDR aims to control the expected number of false discoveries divided by the total number of discoveries, which can be interpreted as the level of noise in the results. If the FDR is set to , it means that you expect percent of the claimed discoveries to be false. To apply the FDR correction, calculate the p-values normally and arrange them in ascending order. Each p-value is annotated with its rank (e.g., the smallest p-value is labeled as (1)).

**Basic idea**: \* Suppose you do tests \* You want to control FDR at level so \* Calculate P-values normally \* Order the P-values from smallest to largest \* Call any significant

**Pros**: Still pretty easy to calculate, less conservative (maybe much less)

**Cons**: Allows for more false positives, may behave strangely under dependence

Now we consider the smallest calculated p-value, denoted as (1), and order them up to the maximum p-value, denoted as (m). Let’s assume there are m hypothesis tests. For each p-value at the ith position, we check if it is less than or equal to alpha times i divided by m. If this condition is true, we deem it significant; otherwise, we do not. This procedure is designed to control the false discovery rate. It shares similarities with the Bonferroni correction in terms of ease of calculation, but it is less conservative. If there is a substantial amount of signal present and a few false positives can be tolerated, the false discovery rate correction may lead to the discovery of more genuine signals. However, it does allow for more false positives, especially if the error rate is set to a large value. Additionally, it may exhibit peculiar behavior when there is dependence among the hypothesis tests.

Example: In the following graph we will control all the error rates we discussed at a level of 0.2. Here, we have ten p-values ordered from smallest to largest. On the y-axis, we have the p-values themselves. The red line represents all the p-values that would be considered significant at an uncorrected alpha level of 0.2. It corresponds to the p-values below 0.2. Although this no-correction approach controls the false positive rate, it can lead to a significant number of false positives when conducting numerous hypothesis tests.



Figure : Figure 2.1: Plotting the power curve

Next, let’s examine the false discovery rate. This correction aims to control the proportion of false positives at a level of 0.2. To calculate this, we follow the gray line, which is a linear line with a slope determined by the alpha level. We compare each p-value with this line from smallest to largest. In this example, we would call the first three p-values significant, thus controlling a slightly different error measure. Lastly, the Bonferroni correction involves dividing 0.2 by the number of hypothesis tests (in this case, 10) to obtain 0.02. This corresponds to a horizontal line. According to the Bonferroni correction, only the first two p-values would be considered significant, and the remaining ones would be deemed insignificant. The Bonferroni correction imposes a more stringent control over the family-wise error rate. This example provides a conceptual understanding of how different procedures work and where the cutoffs are set for different sets of p-values.

Another approach is to adjust the p-values instead of adjusting the alpha level. Adjusted p-values, also known as corrected p-values, can be calculated to directly control error measures without modifying the alpha parameter. It is worth noting that adjusted p-values do not possess the same properties as classical p-values and should not be treated as such. However, they can be useful for controlling error measures.

Example: Using the Bonferroni correction show how adjusted p-values work. Suppose we have m p-values. To adjust them, we can calculate m times each p-value and take the maximum of that and 1. This ensures that the adjusted p-values do not exceed 1, just like the original p-values. By multiplying the p-values by m instead of dividing alpha by m, we can determine the number of adjusted p-values that are less than alpha. This yields the same set of significant results as before. Consequently, we can use these adjusted p-values, such as Bonferroni-adjusted p-values, to assess significance by comparing them to the original alpha level, such as 0.05. If we multiply the p-values by the number of tests performed and count how many are less than alpha, we effectively control the family-wise error rate at level alpha.

Now, let’s consider an example where there are no true positives.

set.seed(1010093)  
pValues <- rep(NA,1000)  
for(i in 1:1000){  
 y <- rnorm(20)  
 x <- rnorm(20)  
 pValues[i] <- summary(lm(y ~ x))$coeff[2,4]  
}  
  
# Controls false positive rate  
sum(pValues < 0.05)

In this simulation, 1,000 data sets are generated where the variables y and x have no relationship. Despite this absence of a relationship, the p-values for the relationship between y and x are calculated for all 1,000 simulated examples. Among these p-values, the number that is less than 0.05 is determined. In this case, even though there is no true relationship, approximately 5% of the performed tests are falsely identified as significant, amounting to 51 tests. To address this, we can adjust the p-values and apply corrections like the Bonferroni correction or the Benjamini-Hochberg correction for controlling the false discovery rate. Using the P.adjust function in R, the p-values are adjusted accordingly.

# Controls FWER   
sum(p.adjust(pValues,method="bonferroni") < 0.05)  
# Controls FDR   
sum(p.adjust(pValues,method="BH") < 0.05)

For the Bonferroni correction, the adjusted p-values are compared to the alpha level of 0.05, resulting in zero significant discoveries when there are no true positives. Similarly, the Benjamini-Hochberg correction can be applied by adjusting the p-values and examining the number of adjusted p-values less than 0.05. Again, in the case of no significant relationships, no discoveries should be made, and the result reflects this expectation.

Now, let’s consider another simulated scenario to further illustrate the concepts. In this case, we will have a relationship between the two variables 50% of the time.

set.seed(1010093)  
pValues <- rep(NA,1000)  
for(i in 1:1000){  
 x <- rnorm(20)  
 # First 500 beta=0, last 500 beta=2  
 if(i <= 500){y <- rnorm(20)}else{ y <- rnorm(20,mean=2\*x)}  
 pValues[i] <- summary(lm(y ~ x))$coeff[2,4]  
}  
trueStatus <- rep(c("zero","not zero"),each=500)  
table(pValues < 0.05, trueStatus)

To simulate this, we generate 1,000 sets of y and x variables. For the first 500 sets, the y values are independent of x, while for the last 500 sets, the y values have a mean equal to 2 times x, indicating a relationship between y and x. We calculate a p-value for each case and define the true status as beta = 0 for the first 500 sets and beta ≠ 0 for the last 500 sets. This allows us to analyze the results.

First, let’s examine the number of p-values less than 0.05 without any correction. We find that, in cases where there is actually no relationship between the variables, we still have around 5% false positive results. On the other hand, for the cases where there is a relationship, all the p-values are less than 0.05, correctly identifying the true signals.

# Controls FWER   
table(p.adjust(pValues,method="bonferroni") < 0.05,trueStatus)  
# Controls FDR   
table(p.adjust(pValues,method="BH") < 0.05,trueStatus)

Next, if we apply the Bonferroni correction by adjusting the p-values using the P.adjust function with the method set to Bonferroni, we observe slightly fewer significant results. In other words, we miss 23 cases where there should be a signal. However, we have zero false positives because we are controlling the probability of any false positive to be less than 0.05.

Now, let’s consider the false discovery rate (FDR) and apply the Benjamini-Hochberg correction. By setting the method to “bh” in the P.adjust function, we discover all the significant results, but we identify fewer false positives compared to the uncorrected analysis. In this case, around 5% of the cases are falsely called significant, while only about 5% of the time is there actually no true relationship.

To visualize the effects of the p-value adjustment, we can plot the p-values versus the adjusted p-values for both the Bonferroni correction and the Benjamini-Hochberg correction. For the Bonferroni method, we multiply each p-value by the number of tests performed (1,000 in this case). As a result, the smallest p-values remain less than one, but beyond a certain point, all the p-values multiplied by 1,000 become equal to or greater than one. Since adjusted p-values cannot exceed one, we observe a flat line in the plot.



Figure : Figure 1.12: Plotting the p\_values

par(mfrow=c(1,2))  
plot(pValues,p.adjust(pValues,method="bonferroni"),pch=19)  
plot(pValues,p.adjust(pValues,method="BH"),pch=19)

Overall, these analyses demonstrate the usefulness of p-value adjustment methods in understanding and controlling error rates, even though they may not directly impact the performance of hypothesis tests.

In contrast, when applying the Benjamini-Hochberg correction, we observe a slightly increasing function between the p-value and the adjusted p-value. The adjusted p-value is generally larger across the entire range compared to the actual p-value, although not significantly larger in this particular case. This behavior occurs because there are many significant results present.

Now, let’s discuss some additional notes and resources related to multiple testing. Multiple testing is a subfield of statistics with various correction methods available depending on factors such as the dependence structure and specific choices made in the statistical model.

For most standard problems, the basic Bonferroni or Benjamini-Hochberg correction is usually sufficient. However, if there is strong dependence between the tests, you may want to consider using the “method = BY” option in the p.adjust function or explore other direct adjustments tailored to the dependence between hypothesis tests. We have conducted some research in this area, and I invite you to explore my papers for more information.

Regarding resources, there is a comprehensive and informative paper titled “A Gentle Introduction to Multiple Testing Procedures with Applications in Genomics.” It focuses on genomics, an area where multiple testing has seen significant development as a statistical discipline. Similarly, “Statistical Significance for Genome-Wide Studies” provides a gentle introduction to multiple testing, even for readers not familiar with molecular biology.

* [Multiple testing procedures with applications to genomics](http://www.amazon.com/Multiple-Procedures-Applications-Genomics-Statistics/dp/0387493166/ref=sr_1_2/102-3292576-129059?ie=UTF8&s=books&qid=1187394873&sr=1-2)
* [Statistical significance for genome-wide studies](http://www.pnas.org/content/100/16/9440.full)
* [Introduction to multiple testing](http://ies.ed.gov/ncee/pubs/20084018/app_b.asp)

Lastly, if you’re interested in delving deeper into multiple testing, We recommend a comprehensive introduction that covers the basics in more depth. It serves as a valuable resource for expanding your understanding of this topic. ## Resampling Resampling-based procedures offer a means to perform statistical inferences based on the data at hand, without relying heavily on population parameters. Data scientists tend to favor resampling-based inferences due to their data-centric nature, scalability to large studies, and minimal reliance on assumptions. These procedures allow us to make robust inferences by leveraging the available data and are particularly valuable when dealing with complex or uncertain scenarios. ### Bootstrapping The bootstrap method, invented by the renowned statistician Bradley Efron in 1979, is an incredibly valuable tool for constructing confidence intervals, estimating standard errors, and performing inferences that would otherwise be challenging. In fact, it is considered one of the most important procedures in the history of statistics, as it has liberated data analysts from relying heavily on complex mathematical calculations to obtain distributional properties of statistics.

The term “bootstrap” originates from the phrase “pulling oneself up by one’s bootstraps.” The concept of the bootstrap procedure is not about achieving the physically impossible, but rather about leveraging available information to make reliable inferences. It enables us to directly connect the information we have with the inferences we make, making it a fitting name for this statistical technique. The bootstrap method is both important and liberating, aligning well with the spirit of data science. Example: Imagine we want to evaluate the behavior of the average of 50 die rolls. The population distribution is depicted by the equally weighted bars representing the numbers one through six.



Figure : Figure 1.14: Die roll

There are a couple of approaches we could take. One option is to mathematically derive the distribution of the average of 50 die rolls without resorting to simulation. Another approach is to roll the die 50 times, calculate the average, and repeat this process numerous times to obtain multiple averages. However, this method assumes that the true population distribution is one-sixth for each number. While it may be challenging to determine the probability of obtaining a value like 5.5 through calculations alone, rolling the die multiple times and averaging the results provides a practical and intuitive way to estimate it. For certain statistics like the average, we have some knowledge about their distribution, such as the centering around the population mean and the variance being equal to sigma squared divided by n. However, we will focus on the average initially to demonstrate how the bootstrap method extends to other statistics beyond the average.

Consider a similar problem where we are uncertain about the fairness of the die generating our data. We don’t know the probabilities assigned to each number (one, two, three, four, five, or six). We only have a single sample of size 50, so we don’t have a distribution of averages from the true die that generated the data. On the left-hand side, you can see a histogram representing the occurrences of each number based on one sample realization from this unknown population.



Figure : Figure 1.15: Die roll

In this scenario, we cannot evaluate the behavior of averages of 50 die rolls from this population since we don’t know the population distribution or which die to roll from. Here’s where bootstrapping comes into play. Instead of sampling from the true distribution, we can sample repeatedly from our empirical distribution. We create collections of 50 die rolls using the probabilities estimated from our sample. In this case, we would sample from the blue bars, considering their respective probabilities, and repeat this process to obtain averages from these samples. By doing so, we can understand the distribution of averages, even though we only have one true average from the real population. Mechanically, the bootstrap procedure involves taking each observation (one, two, three, four, five, or six) and putting them into a bag. Then, we draw samples of size 50 from this bag with replacement, meaning the same data point might be selected multiple times. We calculate the average for each sample and repeat this process several times. The fundamental idea is to precisely replicate the simulation experiment we would have conducted if we had access to the true population die and used the observed distribution generated by the specific realization of 50 die rolls we obtained.

This is the basic principle of bootstrapping: we utilize our observed data to construct an estimated population distribution. Then, we simulate from this estimated distribution to understand the distribution of a statistic we are interested in. In the upcoming examples, we will explore less contrived scenarios to illustrate the application of bootstrapping.

Now, we will explore the data using the R programming language and specifically the father.son dataset. To simplify the process, I will define the variable x to represent the sons’ heights from the dataset. We’ll denote n as the number of observations, and for our bootstrapping procedure, we will perform 10,000 bootstrap resamples.

library(UsingR)  
data(father.son)  
x <- father.son$sheight  
n <- length(x)  
B <- 10000  
resamples <- matrix(sample(x,  
 n \* B,  
 replace = TRUE),  
 B, n)  
resampledMedians <- apply(resamples, 1, median)

First, we will use the sample command. When we specify sample(x, n, replace = TRUE), it means that we draw a sample of size n from the variable x with replacement.



Figure : Figure 4.1: Plot of the histogram of the resamples

In other words, each time we select an observation, we put it back into the pool before drawing the next one. This process simulates resampling from an empirical distribution defined by our data, where each observed data point has a probability of 1/n. This is known as the empirical distribution.

Next, we arrange these resamples into a matrix with the number of bootstrap rows and the number of sample size columns. Each row of this matrix, called “resamples,” represents a completely resampled dataset. We take the original data and draw a sample of size “n” with replacement to create each resampled dataset. This is equivalent to regenerating a sample of size “n” from a distribution that assigns a probability of 1/n to each observed data point.

Suppose we are interested in a specific statistic, such as the median. For each resampled dataset, we calculate the median using the apply statement. The density estimate obtained from the resampled medians is visualized, with a vertical line representing the observed median. In this case, we have repeatedly drawn new datasets of the same length from the collection of sons’ heights, applying the median calculation each time. This process is repeated 10,000 times, resulting in a collection of medians. The density estimate provides an approximation of the distribution of medians based on our observed data when we lack knowledge of the actual population distribution. The median serves as a central quantity in bootstrapping.

Using the resampled distribution, we can perform various calculations. For example, we can compute the standard deviation of this distribution to estimate the standard error of the median. Additionally, we can use quantiles to obtain confidence interval estimates. These are the types of analyses commonly performed using bootstrap resampled distributions.

### 4.2.1 Notes on the bootstrap

The bootstrap principle states that if you have a statistic that estimates a population parameter of interest, but you don’t know the sampling distribution of that statistic, you can use the distribution defined by the observed data to investigate and estimate the sampling distribution. While the bootstrap principle doesn’t necessarily require simulation, it is often easier and more practical to use simulation methods. For instance, if you want to determine the distribution of the average of 50 die rolls, it would be challenging to mathematically derive it precisely, making simulation a more feasible approach.

In the context of bootstrapping, we will focus on a few aspects, such as creating confidence intervals and estimating standard errors. The field of bootstrapping offers a wealth of possibilities beyond what we will cover here.

Let’s review the general procedure. You start with your observed data and simulate complete datasets by drawing from the observed data with replacement. This process is an approximation of drawing from the sampling distribution of the statistic of interest, as far as the observed data approximates the real population distribution. As a reminder, we calculate the statistic for each simulated dataset and then use these simulated statistics to define a confidence interval or compute the standard deviation of the distribution to estimate the standard error.

Now, let’s dive into the bootstrap algorithm for calculating a confidence interval or the bootstrap standard error for the median based on a set of N observations. We begin with a vector of length N representing our data. The next step is to resample N observations with replacement from the observed data to create a resampled complete dataset. It’s crucial to sample with replacement to allow for the possibility of repeated observations in the resampled dataset. If we didn’t sample with replacement, we would essentially end up with a copy of the original dataset, albeit with a different order. In our example, we calculate the median for each resampled dataset. If you’re interested in a different statistic, you would simply compute that statistic for the simulated dataset. We repeat this resampling step B times, where B is the number of bootstrap resamples. It is advisable to choose a large value for B to minimize the Monte Carlo error, which refers to the error introduced by the approximation involved in using resampling. Ideally, we would know the exact bootstrap distribution without resorting to resampling, but in practice, we rely on Monte Carlo methods. Setting B to a sufficiently large value, such as 10,000 or more, ensures that the medians obtained from the resampling process are drawn approximately from the sampling distribution of the median based on N observations.

The bootstrap procedure allows us to approximate the population distribution, which is often sufficient for our purposes. There is substantial theoretical evidence supporting the effectiveness of the bootstrap method. With bootstrap resamples, we can perform various analyses. First, it is common to visualize the distribution using density estimates or histograms. Additionally, we can calculate the standard deviation of the bootstrap resamples, providing an estimate of the standard error of the median. Moreover, by taking quantiles of the bootstrap resampled medians, such as the 2.5th and 97.5th percentiles, we can construct a bootstrap confidence interval for the median. This approach provides a straightforward way to develop a confidence interval without relying on complex statistical techniques.

Now let’s go through a quick example code using the father-son data. We have a vector X representing the sons’ heights, B as the number of bootstrap resamples (set to 10,000), and N as the length of X. To obtain complete resampled datasets, we use the sample command in R with replace = TRUE, sampling N \* B elements from X. The resulting vector is then reshaped into a matrix with B rows and N columns, where each row represents a bootstrap resample. Next, we calculate the median for each row, and the standard deviation of these medians provides an estimate of the standard error of the median.

B <- 10000  
resamples <- matrix(sample(x,  
 n \* B,  
 replace = TRUE),  
 B, n)  
medians <- apply(resamples, 1, median)  
sd(medians)  
quantile(medians, c(.025, .975))

For the construction of a confidence interval, we extract the vector of resampled medians and compute the quantiles. In this example, we take the 2.5th and 97.5th percentiles to obtain a 95% confidence interval. The histogram of the bootstrap resamples is plotted using ggplot, which takes a data frame as input. The resulting plot provides an estimate of the sampling distribution of the median based on the bootstrap resamples.



Figure : Figure 4.2: Histogram of bootstrap resamples

It’s worth noting that the bootstrap procedure we discussed here is the non-parametric bootstrap. The confidence interval obtained by taking the percentiles can be improved upon by using the bias-corrected and accelerated (BCa) interval. This interval accounts for bias and is easy to implement using the bootstrap package in R. This introduction provides a basic understanding of the bootstrap method. It is an incredibly useful procedure with numerous variations and wide applicability. When dealing with specific scenarios such as time series, regression models, longitudinal or multilevel data, there are additional considerations and techniques to apply the bootstrap effectively.

### 4.2.2 Permutation tests

We’ll conclude the class by discussing a highly valuable tool for data scientists known as permutation tests. These tests are particularly useful for group comparisons, so let’s start with an example to illustrate their application.



Figure : Figure 2.3: Insect spray data

In this study, researchers conducted experiments with batches of insects, applying different pesticides labeled as sprays. The number of dead insects in each batch was recorded for each spray. Let’s focus on comparing insect spray B with insect spray C. The null hypothesis states that the distributions of observations from each group are the same, implying that the specific labels of the counts are irrelevant. To operationalize this, we can imagine a data frame with counts in one column and spray group labels in another column. We would calculate a statistic such as the difference in the average number of insects killed between group B and group C. This observed test statistic represents the difference we observe in the data.

Now, let’s consider permuting the group labels. We can randomly shuffle the vector of labels using a command like sample in R. After permuting, we recalculate the test statistic for each permutation. You can choose any statistic you prefer, such as the mean difference in counts, the geometric means, or even a t-statistic. Rather than comparing the test statistic with a t-null distribution, we compare it to a permutation-based null distribution. To calculate a p-value, we determine the percentage of simulations in which the simulated statistic was more extreme, favoring the alternative hypothesis, compared to the observed statistic. In the case of a difference in means, “more extreme” would mean a greater difference in means in the direction of the alternative hypothesis. This process yields a permutation-based p-value.

| Data type | Statistic | Test name |
| --- | --- | --- |
| Ranks | rank sum | rank sum test |
| Binary | hypergeometric prob | Fisher’s exact test |
| Raw data |  | ordinary permutation test |

Permutation tests are powerful and have been reinvented in various settings. For example, the rank sum test, a well-known test statistic, is a permutation test where the data is replaced by ranks instead of the original raw values. Fisher’s exact test, which you may have heard of, is another permutation-based test in which the data is binary, and a specific test statistic is employed. If you use the raw data directly, you are essentially conducting an ordinary permutation test. It’s worth mentioning that randomization tests exist as a separate entity, specifically when group labels are explicitly randomized. In our insect spray example, batches could have had the sprays randomized to them, making it a randomized test.

Permutation tests have broad applicability and are a powerful tool in the data scientist’s toolkit. Operationally, a randomization test follows a similar procedure to a permutation test. However, the conclusions drawn from a randomization test may be stronger, and the interpretation of the results can differ slightly. It’s worth noting that permutation strategies can also be employed in regression. In this case, you would permute a regressor, providing a different approach to obtaining a null distribution compared to the normal distributions covered in regression classes.

Permutation tests are particularly effective in multivariate settings, as they allow for the calculation of maximum statistics that control family-wise error rates. While we won’t delve deeply into these topics, our aim is to guide you through a simple permutation test so that you can grasp the fundamental ideas. Let’s proceed with an example to illustrate how to conduct a permutation test.

To begin, let’s subset the data to focus on InsectSprays B and C. Our outcome variable, y, represents the count of dead insects, while the group variable represents the spray or pesticide labels. We align the y and group vectors accordingly, ensuring each element corresponds to the same batch. Our test statistic is the average difference in the mean number of dead insects between groups B and C across batches. The observed test statistic is obtained by applying the test statistic function to our outcome and group vectors, which are correctly aligned.

subdata <- InsectSprays[InsectSprays$spray %in% c("B", "C"),]  
y <- subdata$count  
group <- as.character(subdata$spray)  
testStat <- function(w, g) mean(w[g == "B"]) - mean(w[g == "C"])  
observedStat <- testStat(y, group)  
permutations <- sapply(1 : 10000, function(i) testStat(y, sample(group)))  
observedStat  
mean(permutations > observedStat)

Next, we break the alignment by permuting the group labels. We use the sample function to randomly permute the group labels, generating 10,000 test statistics where the association between y and group is disrupted due to the permuted labels. This process is performed under the null hypothesis that the group labels are unrelated to the outcome. The observed statistic of 13.25 represents the average count of dead insects for group B minus the count for group C, suggesting an average excess of 13 dead insects per batch for group B compared to group C.

We calculate the percentage of permuted test statistics that are larger or more extreme in favor of the alternative hypothesis compared to the observed statistic. In this particular dataset, we find that the percentage is zero, indicating that across 10,000 permutations, we couldn’t find a configuration of group labels that resulted in a more extreme test statistic than the observed one. Formally, the p-value is very small, close to zero, leading us to reject the null hypothesis for any reasonable significance level (). It’s worth mentioning that the p-value is not exactly zero because we can consider at least one permutation that yields a test statistic as large as the observed one, namely the permutation that restores our original data. However, this minor detail is inconsequential when comparing the p-value to a standard critical value like 5%.



Figure : Figure 2.4: Histogram of bootstrap resamples

In the graph, we display the null distribution generated by our permutations. This distribution plays a similar role to the t-distribution or standard normal distribution in hypothesis testing, where we assume the data is normal or appeal to the central limit theorem. In this case, the null distribution is centered at zero and ranges from -10 to +10. The vertical line represents the observed statistic, which lies far in the tail of the null distribution, suggesting that the null distribution is likely not the true distribution and that group B is indeed substantially different from group C. The histogram provides a visual representation of this observation, and we can now quantify it with a p-value and perform a formal hypothesis test.

Whether it’s in bootstrapping, where the focus is on the sampling distribution of a statistic, or in permutation testing, where we base formal inferences on the exchangeability of group labels, it is crucial to examine histograms or density estimates of the resampled distributions. These visualizations provide valuable insights into the behavior of the resampling procedure.

## 4.3 Course project

### 4.3.1 Part 1: Simulation Exercise Instructions

In this project you will investigate the exponential distribution in R and compare it with the Central Limit Theorem. The exponential distribution can be simulated in R with rexp(n, lambda) where lambda is the rate parameter. The mean of exponential distribution is 1/lambda and the standard deviation is also 1/lambda. Set lambda = 0.2 for all of the simulations. You will investigate the distribution of averages of 40 exponentials. Note that you will need to do a thousand simulations.

Illustrate via simulation and associated explanatory text the properties of the distribution of the mean of 40 exponentials. You should:

* Show the sample mean and compare it to the theoretical mean of the distribution.
* Show how variable the sample is (via variance) and compare it to the theoretical variance of the distribution.
* Show that the distribution is approximately normal.

In point 3, focus on the difference between the distribution of a large collection of random exponentials and the distribution of a large collection of averages of 40 exponentials.

As a motivating example, compare the distribution of 1000 random uniforms

hist(runif(1000))

and the distribution of 1000 averages of 40 random uniforms

mns = NULL  
for (i in 1 : 1000) mns = c(mns, mean(runif(40)))  
hist(mns)

This distribution looks far more Gaussian than the original uniform distribution! This exercise is asking you to use your knowledge of the theory given in class to relate the two distributions. Confused? Try re-watching video lecture 07 for a starter on how to complete this project.

### 4.3.2 Part 2: Basic Inferential Data Analysis Instructions

In the second portion of the project, we’re going to analyze the ToothGrowth data in the R datasets package.

* Load the ToothGrowth data and perform some basic exploratory data analyses
* Provide a basic summary of the data.
* Use confidence intervals and/or hypothesis tests to compare tooth growth by supp and dose. (Only use the techniques from class, even if there’s other approaches worth considering)
* State your conclusions and the assumptions needed for your conclusions.

## 4.4 Practical R Exercises in swirl

During this week of the course you should complete the following lessons in the Statistical Inference swirl course:

* Power
* Multiple Testing
* Resampling

# About the Authors

These credits are based on our [course contributors table guidelines](https://www.ottrproject.org/more_features.html#giving-credits-to-contributors).

| Credits | Names |
| --- | --- |
| **Pedagogy** |  |
| Lead Content Instructor(s) | [FirstName LastName](link%20to%20personal%20website) |
| Lecturer(s) (include chapter name/link in parentheses if only for specific chapters) - make new line if more than one chapter involved | Delivered the course in some way - video or audio |
| Content Author(s) (include chapter name/link in parentheses if only for specific chapters) - make new line if more than one chapter involved | If any other authors besides lead instructor |
| Content Contributor(s) (include section name/link in parentheses) - make new line if more than one section involved | Wrote less than a chapter |
| Content Editor(s)/Reviewer(s) | Checked your content |
| Content Director(s) | Helped guide the content direction |
| Content Consultants (include chapter name/link in parentheses or word “General”) - make new line if more than one chapter involved | Gave high level advice on content |
| Acknowledgments | Gave small assistance to content but not to the level of consulting |
| **Production** |  |
| Content Publisher(s) | Helped with publishing platform |
| Content Publishing Reviewer(s) | Reviewed overall content and aesthetics on publishing platform |
| **Technical** |  |
| Course Publishing Engineer(s) | Helped with the code for the technical aspects related to the specific course generation |
| Template Publishing Engineers | [Candace Savonen](https://www.cansavvy.com/), [Carrie Wright](https://carriewright11.github.io/), [Ava Hoffman](https://www.avahoffman.com/) |
| Publishing Maintenance Engineer | [Candace Savonen](https://www.cansavvy.com/) |
| Technical Publishing Stylists | [Carrie Wright](https://carriewright11.github.io/), [Ava Hoffman](https://www.avahoffman.com/), [Candace Savonen](https://www.cansavvy.com/) |
| Package Developers ([ottrpal](https://github.com/jhudsl/ottrpal)) [Candace Savonen](https://www.cansavvy.com/), [John Muschelli](https://johnmuschelli.com/), [Carrie Wright](https://carriewright11.github.io/) |  |
| **Art and Design** |  |
| Illustrator(s) | Created graphics for the course |
| Figure Artist(s) | Created figures/plots for course |
| Videographer(s) | Filmed videos |
| Videography Editor(s) | Edited film |
| Audiographer(s) | Recorded audio |
| Audiography Editor(s) | Edited audio recordings |
| **Funding** |  |
| Funder(s) | Institution/individual who funded course including grant number |
| Funding Staff | Staff members who help with funding |

## ─ Session info ───────────────────────────────────────────────────────────────  
## setting value   
## version R version 4.0.2 (2020-06-22)  
## os Ubuntu 20.04.5 LTS   
## system x86\_64, linux-gnu   
## ui X11   
## language (EN)   
## collate en\_US.UTF-8   
## ctype en\_US.UTF-8   
## tz Etc/UTC   
## date 2023-07-12   
##   
## ─ Packages ───────────────────────────────────────────────────────────────────  
## package \* version date lib source   
## assertthat 0.2.1 2019-03-21 [1] RSPM (R 4.0.5)   
## bookdown 0.24 2023-03-28 [1] Github (rstudio/bookdown@88bc4ea)   
## cachem 1.0.7 2023-02-24 [1] CRAN (R 4.0.2)   
## callr 3.5.0 2020-10-08 [1] RSPM (R 4.0.2)   
## cli 3.6.1 2023-03-23 [1] CRAN (R 4.0.2)   
## crayon 1.3.4 2017-09-16 [1] RSPM (R 4.0.0)   
## desc 1.2.0 2018-05-01 [1] RSPM (R 4.0.3)   
## devtools 2.3.2 2020-09-18 [1] RSPM (R 4.0.3)   
## digest 0.6.25 2020-02-23 [1] RSPM (R 4.0.0)   
## ellipsis 0.3.1 2020-05-15 [1] RSPM (R 4.0.3)   
## evaluate 0.20 2023-01-17 [1] CRAN (R 4.0.2)   
## fastmap 1.1.1 2023-02-24 [1] CRAN (R 4.0.2)   
## fs 1.5.0 2020-07-31 [1] RSPM (R 4.0.3)   
## glue 1.4.2 2020-08-27 [1] RSPM (R 4.0.5)   
## htmltools 0.5.5 2023-03-23 [1] CRAN (R 4.0.2)   
## knitr 1.33 2023-03-28 [1] Github (yihui/knitr@a1052d1)   
## magrittr 2.0.3 2022-03-30 [1] CRAN (R 4.0.2)   
## memoise 2.0.1 2021-11-26 [1] CRAN (R 4.0.2)   
## pkgbuild 1.1.0 2020-07-13 [1] RSPM (R 4.0.2)   
## pkgload 1.1.0 2020-05-29 [1] RSPM (R 4.0.3)   
## prettyunits 1.1.1 2020-01-24 [1] RSPM (R 4.0.3)   
## processx 3.4.4 2020-09-03 [1] RSPM (R 4.0.2)   
## ps 1.4.0 2020-10-07 [1] RSPM (R 4.0.2)   
## R6 2.4.1 2019-11-12 [1] RSPM (R 4.0.0)   
## remotes 2.2.0 2020-07-21 [1] RSPM (R 4.0.3)   
## rlang 1.1.0 2023-03-14 [1] CRAN (R 4.0.2)   
## rmarkdown 2.10 2023-03-28 [1] Github (rstudio/rmarkdown@02d3c25)  
## rprojroot 2.0.3 2022-04-02 [1] CRAN (R 4.0.2)   
## sessioninfo 1.1.1 2018-11-05 [1] RSPM (R 4.0.3)   
## stringi 1.5.3 2020-09-09 [1] RSPM (R 4.0.3)   
## stringr 1.4.0 2019-02-10 [1] RSPM (R 4.0.3)   
## testthat 3.0.1 2023-03-28 [1] Github (R-lib/testthat@e99155a)   
## usethis 1.6.3 2020-09-17 [1] RSPM (R 4.0.2)   
## withr 2.3.0 2020-09-22 [1] RSPM (R 4.0.2)   
## xfun 0.26 2023-03-28 [1] Github (yihui/xfun@74c2a66)   
## yaml 2.2.1 2020-02-01 [1] RSPM (R 4.0.3)   
##   
## [1] /usr/local/lib/R/site-library  
## [2] /usr/local/lib/R/library

# 5 References