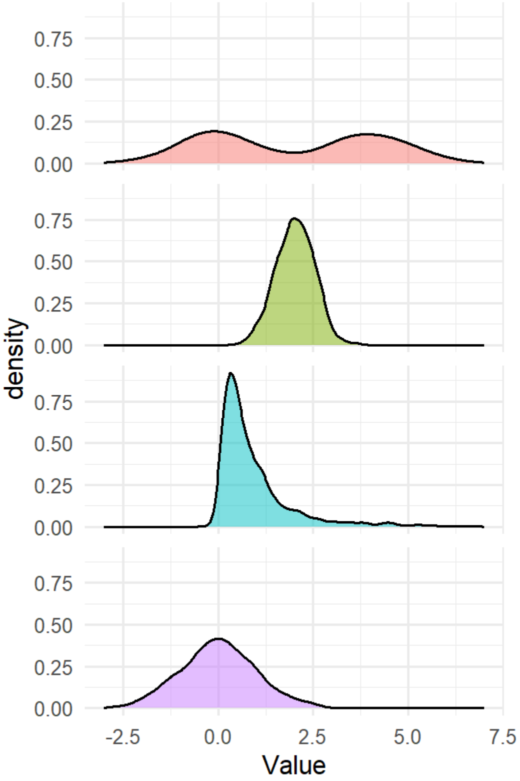
Section 7
Statistical Inference for Means

7.1 – The Central Limit Theorem



In the previous three sections, we explored statistical inference for a proportion. This involved testing hypotheses about a particular proportion and estimating a population proportion with a confidence interval. The key distribution that allowed us to find *p*-values and margins of error for these methods was the binomial distribution. All proportions follow a binomial distribution, regardless of the situation. The population proportion may vary from one situation to another, but this fully determines that distribution, and thus defines the methods we use for statistical inference.

But what if we wanted to do this for numerical data and conduct inference on a mean? Distributions for numerical data can widely vary in their center, spread, and shape. When you collected data on your group's study hours, the center, shape, and variation of your group's distribution likely varied greatly from others' groups. This makes it seem like it would be very difficult to have a consistent method to determine how likely it would be to get a sample of numerical data from a population, as the full population distribution is often unknown, and can vary so greatly!

Normality through sampling

What we hopefully did notice from the study hours activity was that even if we had very different data on how often we studied in our groups, the sampling distribution for the mean was more similar in its shape. To see how the shapes for these sampling distributions of the mean are made, let's examine the distribution of a simple random variable and see what happens if we look at an average of multiples of these random variables.

> *Example*: Consider a container with four slips of paper, each labeled with numbers 1 through 4. Find and graph the pmf of this distribution, and then find the mean and standard deviation.

Example: Consider the container from the previous example, but now draw 2 slips of paper **with replacement**. Find and graph the pmf of the sample mean $\bar{X}$, and then find the mean and standard deviation of $\bar{X}$.

We see the distribution of our sample mean taking an interesting shape even from changing our sample size from 1 to 2, but what happens when we increase the sample size to 3? This becomes increasingly difficult to do by hand. Let's use a simulation in R!

```
v = rep(0, 1000) #placeholder for random draws from the container
n = 3 #number of times we draw from the container
for (i in 1:1000) {
  v[i]=mean(sample(1:4, n, replace=TRUE))
}
hist(v)
mean(v)
var(v)
```

What if we try changing the *n* value in the above code? What happens as we take n larger and larger?

The central limit theorem
This phenomenon is known as the *Central Limit Theorem*. This theorem states that for an *large enough* collection of random variables $X_1, X_2, \ldots, X_n$ that are independent and identically distributed, each with some mean μ and standard deviation σ, $\bar{X}$ has an approximate normal distribution with the same mean μ and standard deviation $\frac{\sigma}{\sqrt{n}}$. In shorthand, this is written $\bar{X} \sim N\left(\mu, {}^{\sigma}/_{\sqrt{n}}\right)$.

Additionally, we can also use what we have learned about using the standard normal distribution and rearrange this statement to say that the standardized sample mean has an approximately sample standard normal distribution, that is, $\frac{\bar{X}-\mu}{{}^{\sigma}/_{\sqrt{n}}} \sim N(0, 1)$.

It's important to note here that if you are collecting data from a population that is normal itself, then these facts are true for samples of any size. This is because the sum of two normal distributions is still a normal distribution, or more intuitively, if you are starting with a normal population, the shape-smoothing process of the CLT is already done!

For those samples that aren't from a known normal population, "large enough" typically means a sample size $n \geq 25$. However, for most sources of data that are unimodal, the distribution of their sample means will appear quite normal even for small samples. Let's look at a simulation applet that illustrates this theorem before trying the examples below.

> *Example*: Consider taking a random sample of 25 electronic components. This certain electronic component has a lifetime whose mean is 2 years and standard deviation is 0.5 years. What is the probability that a single component lasts 2.5 years?

A note regarding a common misconception to note regarding the Central Limit Theorem – even if we take a really large sample from our population, we cannot assume that the distribution of a single observation is normal. It would be really cool if that were the case – statistics would transcend itself from an awesome discipline to an awesome reality-altering discipline! A man can dream.

> *Example*: Consider taking a random sample of 25 electronic components. This certain electronic component has a lifetime whose mean is 2 years and standard deviation is 0.5 years. What is the probability that the mean lifetime of these 25 components is at least 2.5 years?

7.2 - Hypothesis Testing for One Mean

Theoretical background

So what makes the CLT so important for conducting a hypothesis test about numerical data? Well, we may not know what the population looks like, however, we know that in most circumstances, the distribution of possible sample means, $\bar{x}$, follows a normal shape. This gives us a consistent model to be able to compute a *p*-value or margin of error for doing inference!

Thus, for conducting a hypothesis test about a mean, we take a similar approach when deriving hypotheses, which will have the following form:

$$H_0: \mu = \mu_0 \qquad\qquad H_a: \mu\ (>, <, \text{ or } \neq)\ \mu_0$$

Our goal now is to determine a possible test statistic to use for evaluating how far our sample mean $\bar{x}$ is from our hypothesized population mean μ_0. We can directly leverage CLT to do this, as we know that for a large enough sample size, our sample mean has the following distribution:

$$\bar{X} \sim N\left(\mu_0, {}^{\sigma}\!/_{\sqrt{n}}\right)$$

If we standardize the sample mean, we get that the following quantity follows a standard normal distribution:

$$\frac{\bar{X} - \mu_0}{{}^{\sigma}\!/_{\sqrt{n}}} \sim N(0, 1)$$

This gives us a standardized quantity we can use to evaluate how far the sample mean is from the hypothesized population mean, but we have an issue: the formula requires us to know the population standard deviation, which would require having data for the entire population. But if you have data for the entire population, you'd just be able to compute the population mean. So why even test hypotheses about the population mean then if you know it? Ideally, we'd like to be able to use the sample standard deviation here to avoid this catch-22 situation.

Thankfully, a statistician named William Gosset found a way to do this using the *t*-distribution. He found that

$$\frac{\bar{X} - \mu_0}{{}^{s}\!/_{\sqrt{n}}} \sim t(n-1),$$

that is, the standardized sample mean follows a *t-distribution on n-1 degrees of freedom*. This carries the same assumptions from the Central Limit Theorem. Thus, you must either have a large enough sample ($n \geq 25$), or your data must have originally come from a population that's normally distributed.

What exactly is a *t*-distribution? Well, it has a very familiar shape – it looks like the bell curve of a normal distribution, but has slightly heavier tails. As the *degrees of freedom* of the *t*-distribution get larger and larger, the closer the *t* gets to our standard normal distribution.

Let’s draw some rough examples below:

In summary, we carry out our hypothesis test for a mean using the following *t-test statistic*:

$$t = \frac{\bar{x} - \mu_0}{s/\sqrt{n}}$$

Next, we would need to find the probability or *p*-value associated with this test statistic. We can do this with the following R function for probabilities of a *t*-distribution:

```
dt(x, df) #gives the height of the density function
pt(x, df) #gives the probability P(X ≤ x)
qt(x, df) #finds the value k where P(X ≤ k) = x
rt(x, df) #randomly generates x data points that come from a
                t-distribution with df degrees of freedom
```

Example: Starbucks "Grandé" size coffee cups have room for 16 ounces of coffee. However, they don't fill the cup completely up to the brim because that would result in coffee catastrophes! Let's say Starbucks claims they put in 14.50 ounces of coffee on average. To test this claim, you have randomly selected 40 customers who ordered a grandé coffee from the Starbucks at the bookstore to have their coffee content measured. In that sample, the average was 14.42 oz with a standard deviation of 0.19. We are investigating whether there is evidence that Starbucks pours less than 14.50 oz on average into their grandé coffee drinks. Carry out this hypothesis test at α = 0.05 and write out a conclusion.

R code for testing a mean

R has built in functionality to carry out the computations for a hypothesis test of a mean as well, as long as your data that you are testing is stored in the vector `x`.

```
t.test(x, mu=mu0, alternative=ALT)
```

The other arguments of this function work similarly to the ones we used for the `binom.test` function. If we don't have the full data set and only have summary statistics, like in the last example, you need to use the `pt` function defined on the last page to find probabilities.

Example: You are writing a computer program to quickly process and clean data scraped from the internet. The project manager gives you the task of making sure this program takes no more than half a second to execute, on average. You write the program and run it 15 times, recording the execution time each time, and get the following data:

```
0.528, 0.374, 0.446, 0.346, 0.423, 0.265, 0.464, 0.422,
0.213, 0.593, 0.445, 0.529, 0.342, 0.434, 0.516
```

Carry out a test at $\alpha = 0.01$ to determine if your program meets your project manager's specifications.

QQ plots

Remember that the theory behind this test assumed that the central limit theorem applied. The CLT only applies when you are either sampling from a normal distribution, or the sample size is large enough (at least 25). While we didn't know the distribution of the Starbucks drinks in the first example, we at least had a large enough sample size.

What if you want to carry out a test on a small data set, but don't know if it came from a normal distribution? One tool we can use to determine how well data fits a normal distribution is a graphical display called a *QQ plot*. The QQ plot plots quantiles of your sample data against quantiles of the probability model (e.g. normal) that you are comparing to. Ideally, if the data fits the model you are comparing to, the quantiles should match, so these points should fall on the 45-degree line going through the origin.

QQ plots are typically evaluated by eye-test – if the points fall close to the line, it is reasonable to assume the data is coming from population that fits the distribution model you chose. If not, then the model you chose might not be a good assumption.

Below is some example code to show how to make QQ plots for a normal distribution. Two data sets are generated, one coming from a normal distribution, and one from a different distribution (the exponential distribution) that is very right skewed.

```
norm_data = rnorm(100, 68, 4)
exp_data = rexp(100, 3)

qqnorm(norm_data)
qqline(norm_data)

qqnorm(exp_data)
qqline(exp_data)
```

I used this on randomly generated data from R so you can see how QQ plots look when we know the distributions they came from. Even data that is known to be from a normal population will have some deviations from the line! Remember though, we typically use these QQ plots on real data! Try these functions on the data from the computer program example to see if that data appears to potentially come from a normal population.

7.3 – Confidence Interval for a Mean

Theoretical background

Just as we did for proportions, we can derive a confidence interval for the mean in a similar manner: start with bounds for a middle range of the distribution, then rearrange the terms so we are bounding the population parameter. Recall that

$$\frac{\bar{X} - \mu}{\sigma / \sqrt{n}} \sim N(0, 1)$$

If we were interested in a 95% confidence interval, we would bound this term by 1.96 and -1.96 as seen below, and then would solve for μ.

$$-1.96 < \frac{\bar{x} - \mu}{\sigma / \sqrt{n}} < 1.96$$

However, we run into a similar issue here as we did in testing: we can't use the population standard deviation. However, if we substitute for the sample standard deviation, we simply now use the t-distribution instead of the ±1.96 values from the normal distribution. Thus, more generally, we could write out the (1 – α)100% confidence interval for a mean μ based on a sample of size n as

$$\bar{x} \pm t_{1-\alpha/2} \frac{s}{\sqrt{n}}$$

where $t_{1-\alpha/2}$ is the 1 – $\alpha/2$ percentile of the *t*-distribution on n – 1 degrees of freedom. For example, for a 95% confidence interval (α = 0.05 or 5%), we would find the 97.5th percentile of the *t*-distribution.

> *Example*: A random sample of 36 fastening bolts from a machine were given an extensive stress test, and the resulting length of these bolts was measured after removing them from the machine. The mean length of the bolts was 25 mm, and the sample standard deviation was 5 mm. Find a 95% confidence interval for the mean length of the bolts.

R code for estimating a mean

When summary statistics are already provided, we cannot easily leverage R functions like `t.test` to compute a confidence interval. However, in situations where we have access to the data, we can use R to quickly carry out the computation of a confidence interval. When doing this, make sure to either not specify an alternative hypothesis, or that it is specified as two-sided, or else the output will not produce a confidence interval.

```
t.test(x, conf.level=0.95)
```

Let's go back and try to compute a confidence interval for a previous example:

> *Example:* You are writing a computer program to quickly process and clean data scraped from the internet. The project manager gives you the task of making sure this program takes no more than half a second to execute, on average. You write the program and run it 15 times, recording the execution time each time, and get the following data:
>
> ```
> 0.528, 0.374, 0.446, 0.346, 0.423, 0.265, 0.464, 0.422,
> 0.213, 0.593, 0.445, 0.529, 0.342, 0.434, 0.516
> ```
>
> Construct a 98% confidence interval for the population average time for your program to run. How does this interval compare to your manager's standard of 0.5 seconds?

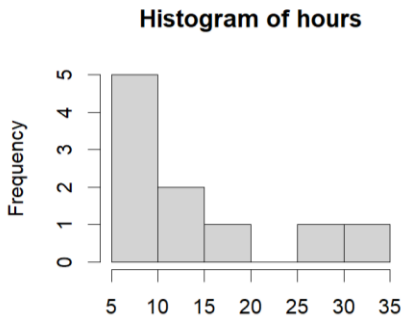
Bootstrap confidence intervals

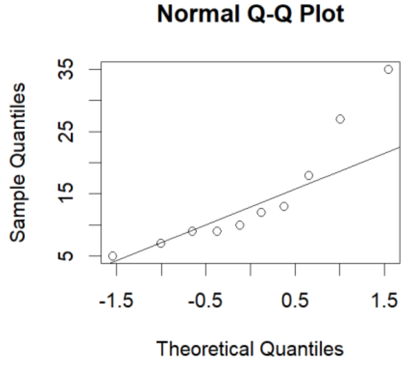
In our last activity, we found confidence intervals for the population mean number of study hours at Illinois through bootstrapping. Just like for proportions, we re-sampled from our data, using this as a stand-in for our population. So if you had a group of 10 students, and one of them studied 12 hours per week, we are assuming that 1/10 (10%) of our population also studies 12 hours per week. This is the same assumption we had for percentages, but it feels strange when applied to numerical data. We only had two possible outcomes in the case of percentages, which may have made using our sample data as a stand-in for the population more sensible. For numerical data, it is possible that all values we re-sample from are unique, and then we are just potentially sampling some values multiple times and some others not at all. Even considering this, the averages of those bootstrapped samples do provide an accurate sampling distribution of the sample mean, which can be used to generate a confidence interval.

Another thing to consider – when using the confidence interval methods discussed in this class, we noted that one of two assumptions needed to be met: that our data came from a normal distribution, or that the sample size was large enough that the central limit theorem would apply for our distribution of the sample mean. What would happen if we had data that did not meet these conditions?

> *Example*: Suppose that a group of 10 Illinois students collected the following data on their weekly study hours: 18, 9, 10, 27, 12, 9, 13, 35, 5, 7. Find a 95% confidence interval for the population mean number of study hours using both the *t*-distribution and bootstrapping methods.

```
hours = c(18, 9, 10, 27, 12, 9, 13, 35, 5, 7)
```





Now, we need to keep in mind that this interval should be used with caution, as it does not meet the assumptions for this test. We can check this by looking at a QQ plot or histogram:

```
hist(hours)
qqnorm(hours)
qqline(hours)
```

To conduct a bootstrap simulation, we would create a sample function again as we did before. We want to take the mean of this new bootstrapped sample:

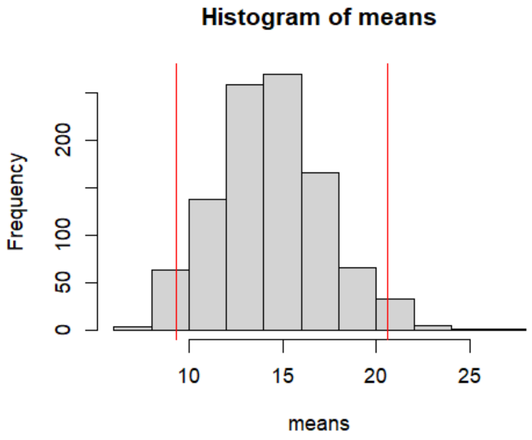
```
mean(sample(hours, 10, replace=TRUE))
```

We can then use the for loop structure to simulate what happens by chance, storing our mean from each simulation in the `means` vector:

```
means = rep(0, 1000)
for (i in 1:1000) {
  means[i] = mean(sample(hours, 10, replace=TRUE))
}
```

We can then get our confidence interval like last time by using the quantile function to find the 2.5th and 97.5th percentiles of our simulated data.

```
CI_bounds = quantile(means, c(0.025, 0.975))
```



While this result will vary depending upon your simulation, it should be quite a bit different than what we found from `t.test`, especially on the lower bound. Let's visualize what our means look like to see why this might have happened!

```
hist(means)
abline(v=CI_bounds[1], col="red")
abline(v=CI_bounds[2], col="red")
```

It appears that this distribution is still somewhat skewed to the right, although we can see that even at *n* = 10 that the distribution is much less skewed than the original data! Because our distribution for the mean still exhibits some right skew, the interval we get from bootstrapping is also shifted to the right[1]. We would expect our interval to be centered around the sample mean of 14.5, but there is more to the right of 14.5 in this interval than the left.

Thus, the bootstrap method can be seen as more robust than using our standard confidence interval method that relies on a normal distribution assumption. It also highlights why we cannot use it for any sample of data, especially with skewed data!

[1] You might also notice that the bootstrap interval is overall narrower. This is because we're assuming our data (a sample) is actually a population, so the spread of the bootstrapped means is based on the population standard deviation of our data, which is smaller than the standard sample deviation. To line up our bootstrap data to the methods of a *t*-interval, we should actually scale the spread of the data up by $\sqrt{n/(n-1)}$ to have the standard error of our bootstrap match the *t*-interval.

7.4 – Additional Practice

Example: One instrument commonly used to measure depression in individuals is the Beck Depression Instrument (BDI), which is measured on a 0-63 scale. A research study examined individuals in five different countries and measured their depression using BDI. They found that the distribution of BDI scores for each country had the means and standard deviations shown in the table to the right. For each probability question given below, either provide the answer, or explain why the probability cannot be found.

Country	*Mean*	*sd*
Norway	5.62	6.97
Spain	3.12	4.84
Finland	6.10	6.82
UK	8.32	8.30
Ireland	8.51	9.16

What is the probability that an individual from Norway has a BDI score of 10 or more?

What is the probability that a random sample of 16 individuals from Spain has an average BDI score of 8 or more?

What is the probability that a random sample of 36 individuals from the UK has a BDI score of 10 or more?

What is the probability that a random sample of 25 individuals from Ireland has a BDI score of 13 or more?

Example: An American football should be inflated to a minimum pressure of 12.5 psi. There is, however, variability in individual footballs due to environmental conditions. This became a hot topic after the 2015 AFC Championship game between the New England Patriots and the Indianapolis Colts, when it was discovered that the footballs provided by the New England Patriots were of consistently lower pressure values. Softer footballs can make them easier to grip and catch.

Both of the head referees tested the footballs of each team during halftime. They were able to test eleven of the Patriots' footballs, and then only had time to test four of the Colts footballs before the game resumed. Two referees, Clete Blakeman and Dyrol Prioleau, each took their own measurement of the footballs. The data they collected can be found in the **deflategate.csv** data file. We will use Clete Blakeman's measurements for the purposes of this practice.

Is there evidence to suggest that the Patriots footballs were inflated below the 12.5 psi regulation on average? Conduct a hypothesis test at α = 0.05 and write out a conclusion.

Example: Suppose that the referees were only able to measure the pressures of 5 footballs during halftime, and that all summary statistics (mean, standard deviation) remained the same as our sample of 10. Would the *p*-value for this test be larger, smaller, or remain the same?

Example: Compute a 90% confidence interval for the air pressure of the Patriots' footballs measured in psi.

Example: Suppose that the referees were able to find an additional 10 footballs and measure their pressure during that halftime. Assume that all other summary statistics (sample mean, standard deviation) remain unchanged even after measuring these 10 new footballs. Would the resulting interval be narrower, wider, or remain the same width? Explain your reasoning.