Module 6

Estimating Uncertainty Confidently

James Baglin

Last updated: 13 July, 2020

Overview

##Summary

Our best, and often only, estimate for a population parameter is the sample estimate. However, due to sampling variability, the sample estimate is always uncertain. Module 6 will introduce the concept of confidence intervals as an interval estimate that expresses the degree of uncertainty associated with sample statistics.

Learning Objectives

The learning objectives associated with this module are:

- Differentiate between a point and interval estimate.
- Define the concept of a confidence interval.
- Discuss the major factors that impact the width of a confidence interval.
- Use technology to calculate confidence intervals for common statistics including means, proportions and rates.

Point and Interval Estimation

The **point estimate** of a sample statistic, such as the mean, median, proportion or rate are single, or point values. They are often our best estimate for a population parameter, but do not express the degree of uncertainty for an estimate associated with its sampling variability. Point estimates should be accompanied by additional information to assist with drawing inferences about the population.

Interval estimation serves to overcome this limitation. The idea is to supplement the point estimate with an interval that reflects the degree of uncertainty associated with a statistic. The most common type of interval estimator is called the **confidence interval**, or CI for short.

Let's consider a typical confidence interval using a hypothetical example. Let's assume height is normally distributed in the population with a standard deviation of 7 cm (This is unrealistic as we often do not know the population SD, but let's go with it for now). An investigator takes a random sample of 10 peoples' height (cm). Say the mean of the sample was found to be 176.5 cm. To calculate a **confidence interval for the mean of a normally distributed variable with a known standard deviation**, we use the following formula:

$$\bar{x} \pm z_{1-(\alpha/2)} \frac{\sigma}{\sqrt{n}}$$

We need to discuss the z critical value, $Z_{1-(\alpha/2)}$, in the above formula. This value is obtained from the standard normal distribution. Recall, the standard normal distribution has the following properties:

$$z \sim N(0, 1)$$

The α value refers to what is known as the **significance level**. Almost all studies will use a standard level of $\alpha = 0.05$. The α value is based on the level of the confidence interval. A CI is defined as $100(1 - \alpha)$ CI. If we use $\alpha = 0.05$, this means we're going to calculate a $100(1 - 0.05)$ = 95% CI.

The z critical value in the confidence interval formula is found by looking up the z-value associated with the $1 - \alpha/2 = 1 - 0.05/2 = 1 - 0.025 = .975$ percentile of the standard normal distribution. We might write:

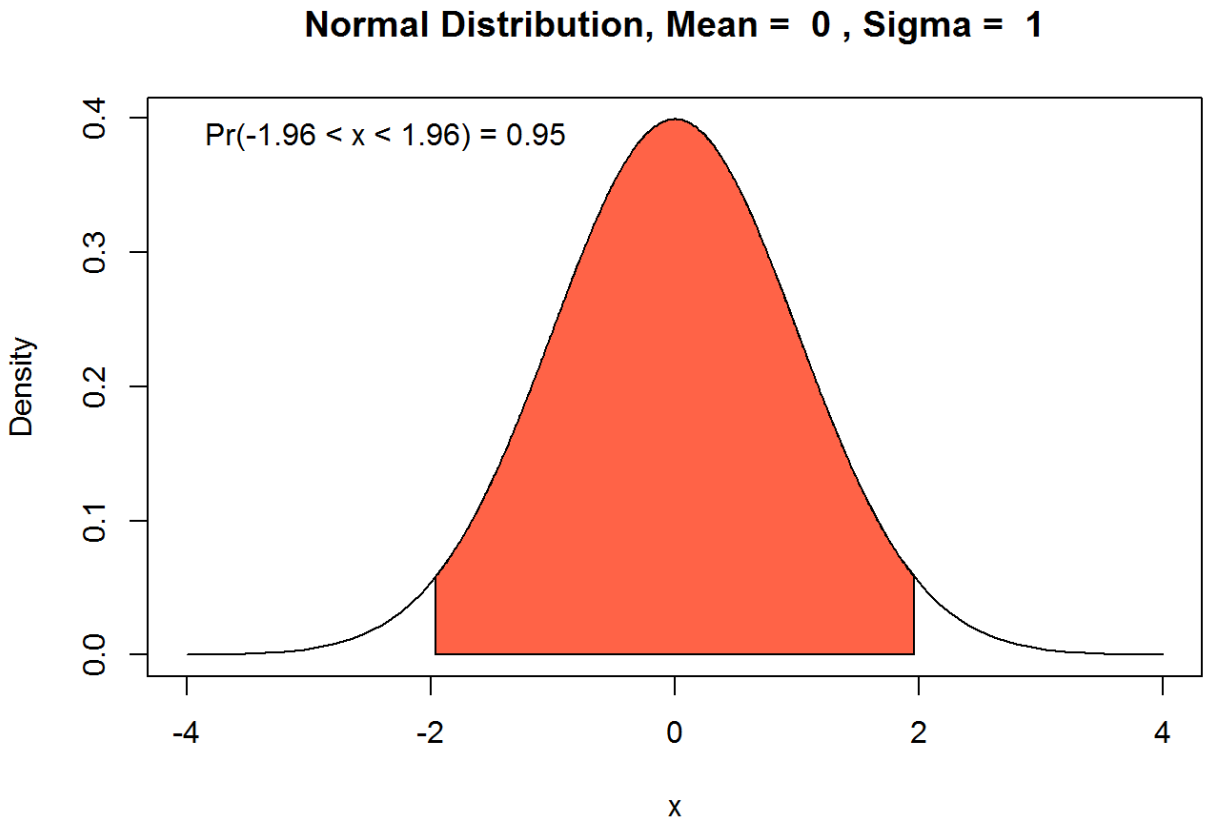
$$Pr(Z < z) = .975$$

We dealt with solving a similar problem in Module 4 (MATH1324_Module_04.html). Fortunately, this is easy to solve in R. To solve the formula above we type the following into R:

```
qnorm(p = .975)
```

```
## [1] 1.959964
```

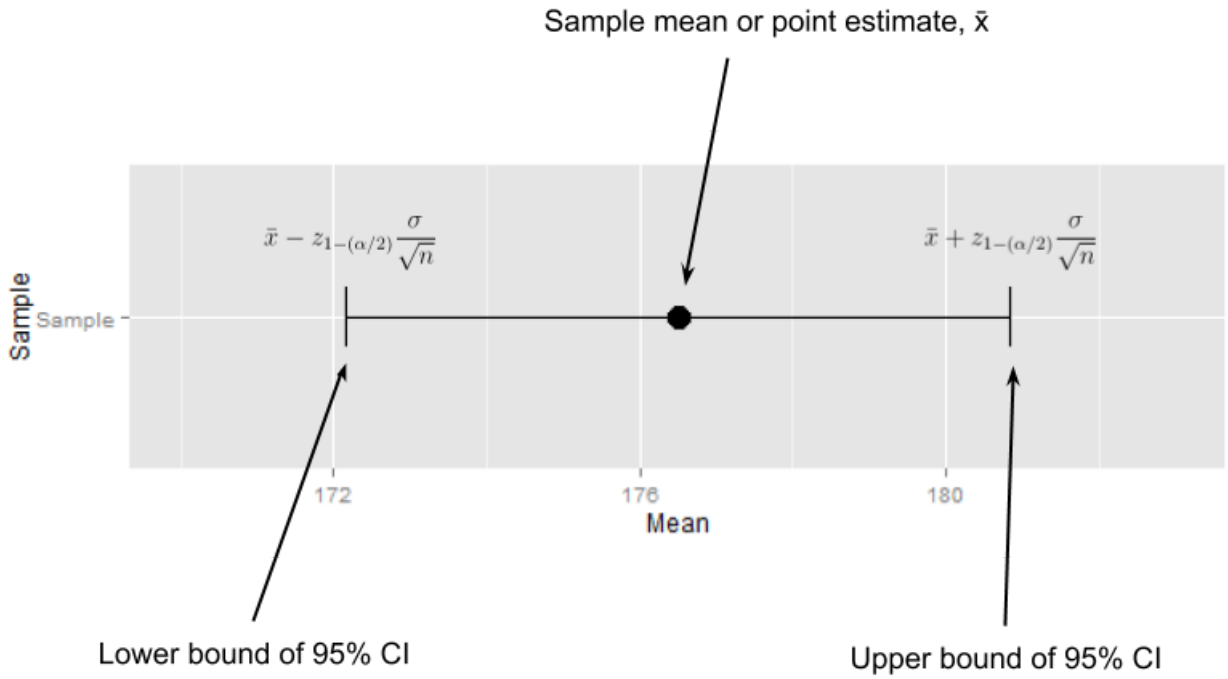
Note, that if we don't specify the mean and standard deviation for `qnorm()`, the R function reverts to a standard normal distribution with `mean = 0` and `sd = 1`. We discover $Pr(Z < 1.96) = .975$. The following figure shows how $z = 1.96$ relates back to the 95% CI. We can see that $Pr(-1.96 < z < 1.96) = .95$ or 95%. This means that 0.025 probability sits in the upper and lower tail of the distribution. As you will discover later on, the critical value is required in the conference interval formula to ensure the confidence interval achieves the desired level of coverage, e.g. 95%.



Beauty! Now we can calculate the 95% CI for the sample mean:

$$176.5 \pm z_{1-(\alpha/2)} \frac{\sigma}{\sqrt{n}} = 1.96 \frac{7}{\sqrt{10}} = 4.34$$

We would write $\bar{x} = 176.5$, 95% CI [172.161, 180.84]. The confidence interval captures a wide range of values for the mean height and reflects the high degree of uncertainty expected from a sample of $n = 10$. This interval is depicted as follows:



Confidence Intervals

Before we dig deeper, you need to have a definition of a CI in the back of your mind. When we refer to confidence in statistics, we need to understand very carefully what this means. In this course we will use a strict frequentist definition. Not all statistics instructors will be this stringent and in the past you may have been taught something different. However, for my course assessment, I expect you to understand and use this definition.

$100(1-\alpha)$% CI, is an interval estimate for a population parameter, based on a given sample statistic, where if samples of a certain size n were repeatedly drawn from the population and a CI for each sample's statistic was calculated, $100(1-\alpha)$% of these intervals would capture the population parameter, whereas the other $100(\alpha)$% would not.

Now with this strange and long-winded definition in mind, let's start digging deeper into the theory.

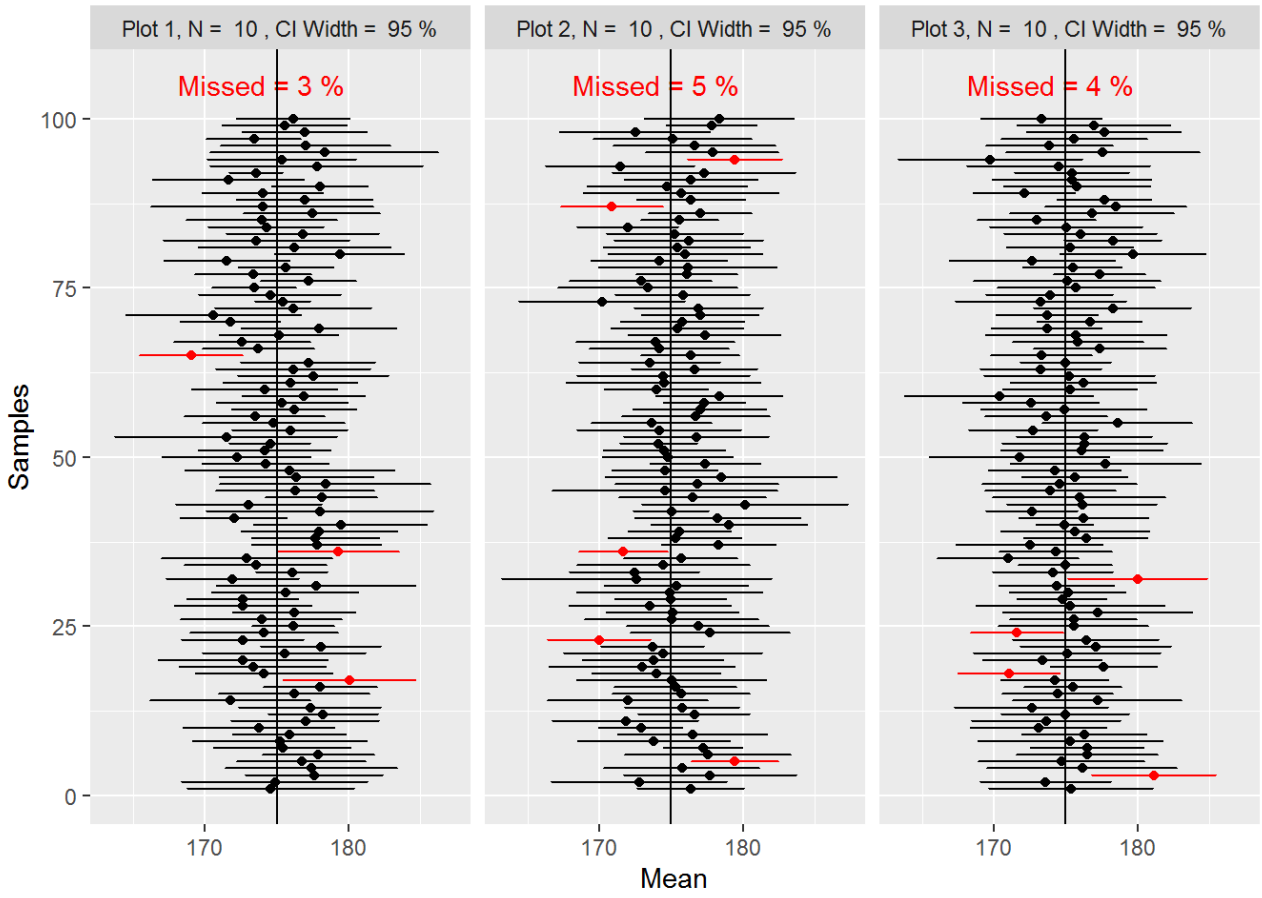
Theory

We will explore confidence interval theory using a simulation and visualisation. We will first make some assumptions about the population. Let's assume height is normally distributed with a mean of 175 and a standard deviation of 7:

$$\text{Height} \sim N(175, 7)$$

As investigators, we don't tend to know population parameters in advance, and hence the reason we need to gather a sample and estimate it. However, for the purpose of this lesson, we need to assume these values.

Now imagine drawing 100 random samples of size 10 from the population. For each of the 100 samples, you calculate the sample mean and 95% CI. You can plot all these means and 95% CIs like in Plot 1 below. Plot 2 and 3 repeat the same procedure, each displaying the means and 95% CIs of 100 random samples of size 10. At the top of each plot, the percentage of CIs that miss capturing the population mean, $\mu = 175$, are reported. Missed intervals are coloured red. For Plot 1, we see that Missed = 3 % of the CIs missed $\mu = 175$, for Plot 2, Missed = 5 %, and for Plot 3, Missed = 4 %. If we repeated this for many thousands of plots and samples, what do you think this missed percentage would average? If you guessed 5%, you're already on your way to understanding CIs. The 5% of CIs that will miss capturing the population mean is our $\alpha = 0.05$.



Think carefully about the plots above and now re-read the CI definition:

$100(1-\alpha)$% CI, is an interval estimate for a population parameter, based on a given sample statistic, where if samples of a certain size n were repeatedly drawn from the population and a CI for each sample's statistic was calculated,

$100(1-\alpha)$% of these intervals would capture the population parameter, whereas the other $100(\alpha)$% would not.

CIs should now be starting to make a little more sense. Basically, CIs are constructed in a way that in the long run, a certain percentage (e.g. 95%) of CIs calculated by repeating the same random sampling procedure will capture a population parameter, for example $\mu = 175$.

CIs can be calculated for a wide range of statistics using formulaic approaches. However, the methods vary depending on the type of statistic being estimated and the assumptions we make about the population from which the data are drawn. Later sections in this module explain calculating CIs for various situations and statistics.

Influencing Factors

Sample Size

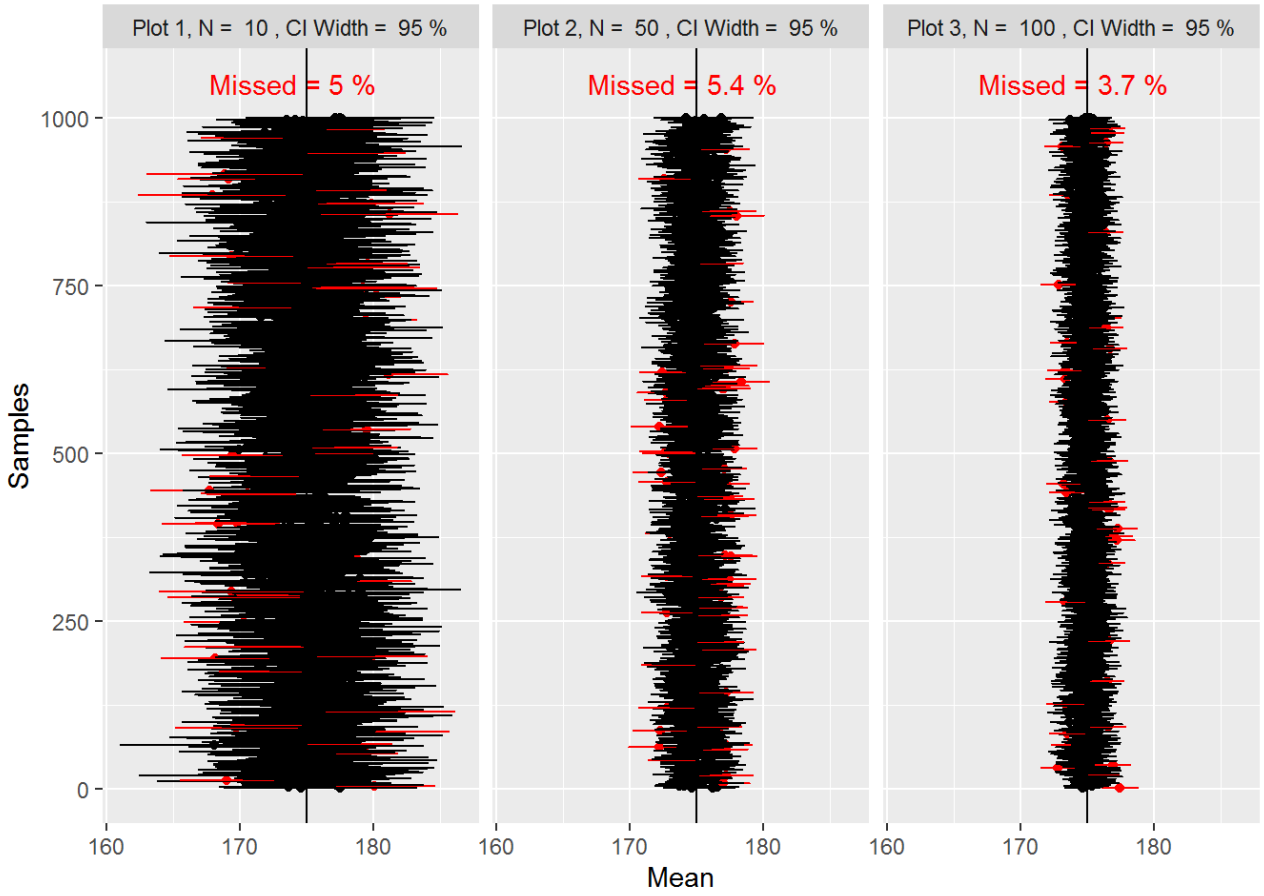
Look at the formula for a CI for the mean of a normally distributed variable with a known standard deviation:

$$\bar{x} \pm z_{1-(\alpha/2)} \frac{\sigma}{\sqrt{n}}$$

You can see the standard error of the mean in the formula:

$$SE = \sigma_{\bar{x}} = \frac{\sigma}{\sqrt{n}}$$

Therefore, it comes as no surprise that CIs share some of the same rules as sampling distributions. Consider the following plots. Note that the number of simulated samples have been increased to 1000 so we get better estimates of the “Missed” percentage. We know from the previous module that sample size shares an inverse relationship with SE. As N increases, SE decreases. Moving from Plot 1 to Plot 3, sample sizes are 10, 50 and100. What happens to the width of the CIs as sample size increases?



We see a dramatic decrease in the width of the intervals. Why? Mathematically, as sample size increases, the SE in the CI formula decreases, meaning that the lower and upper bounds fall closer to the sample mean.

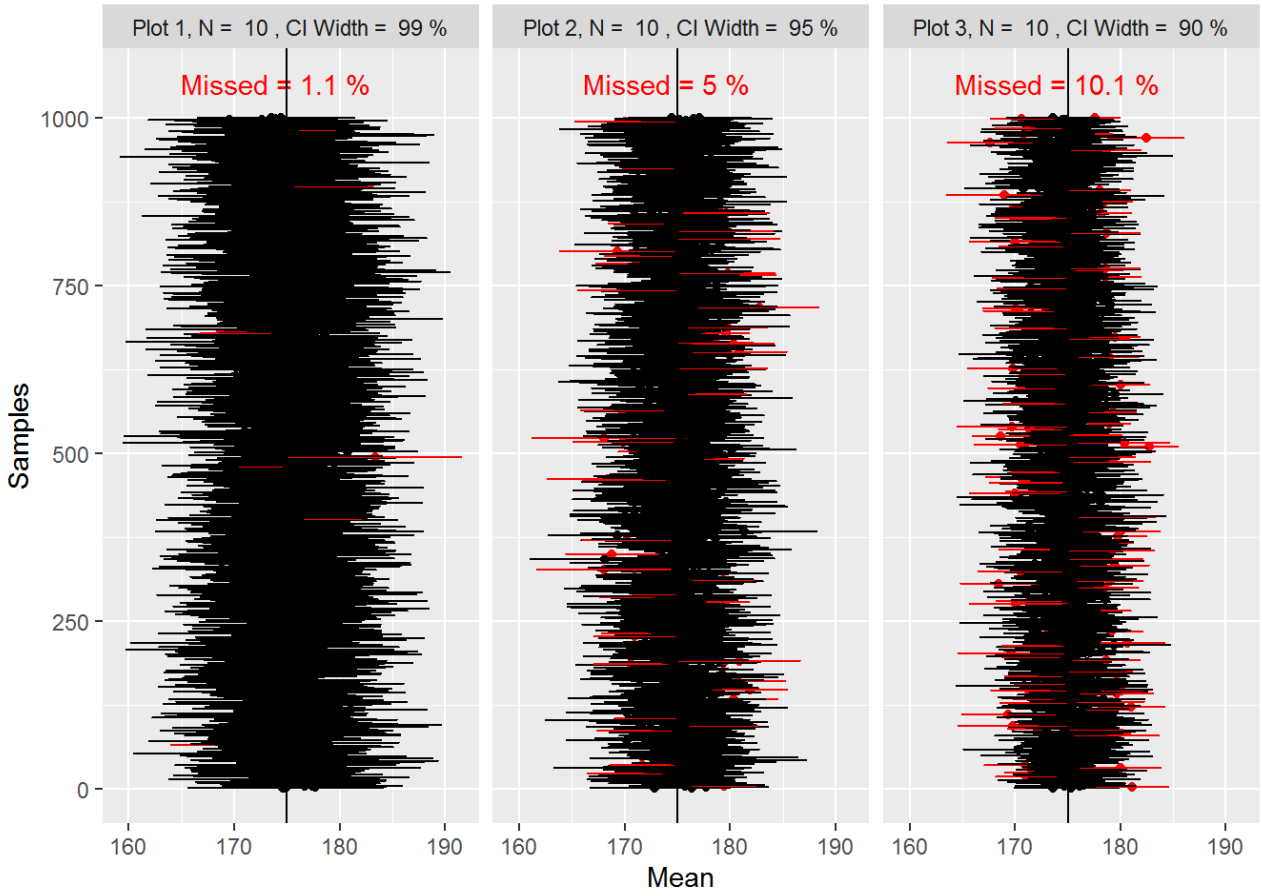
For example, if in the height example the investigator had used a sample size of 30, the confidence interval would become:

$$176.5 \pm z_{1-(\alpha/2)} \frac{\sigma}{\sqrt{n}} = 1.96 \frac{7}{\sqrt{30}} = 2.50$$

and therefore, $\bar{x} = 176.5$, 95% CI [174.00, 179.00]. This 95% CI is narrower than the interval calculated for $n = 10$, 95% CI [172.16, 180.84]. Conceptually, larger random samples are better estimates of the population and therefore, we can be more certain in their estimates over the use of smaller samples.

Confidenece Level

While the 95% CI is the most common, it is possible to use other levels of confidence. Let's compute a 90% and 99% confidence interval for the sample mean height when $n = 10$. This is depicted in the following plot. Plot 1 reports a 99% CI, Plot 2, 95% CI, and Plot 3, 90% CI.



As you can see, higher levels of confidence are associated with wider intervals. This makes sense. If you want to be more confident about capturing a population parameter, cast a wider interval!

Now let's look at the formula. We need to change the z critical value for the confidence interval formula:

$$\bar{x} \pm z_{1-(\alpha/2)} \frac{\sigma}{\sqrt{n}}$$

We will begin with the 90% CI. This CI width corresponds to a significance level of $\alpha = 0.1$. We need to find $Pr(Z < z) = 1 - \alpha/2 = 1 - 0.1/2 = 1 - 0.05 = .95$. In R:

```
qnorm(p = .95)
```

```
## [1] 1.644854
```

The z-value is found to be 1.64. Now completing the 90% CI equation:

$$176.5 \pm z_{1-(\alpha/2)} \frac{\sigma}{\sqrt{n}} = 1.64 \frac{7}{\sqrt{10}} = 3.64$$

We calculate $\bar{x} = 176.5$, 90% CI [172.86, 180.14]. Now compare this 90% CI to the 95% CI [172.16, 180.84]. The 90% CI is narrower. Why? Because as we are less certain about the CI capturing the population parameter in the long run, the width of the confidence

interval will reduce. Think of it like casting a smaller net to capture the population mean.

For a 99% CI, we must change the z critical value again. This CI width corresponds to a significance level of $\alpha = 0.01$. Therefore, we need to find $Pr(Z < z) = 1 - \alpha/2 = 1 - 0.01/2 = 1 - 0.005 = .995$. In R:

```
qnorm(p = .995)
```

```
## [1] 2.575829
```

The z-value is found to be 2.58. Next...

$$176.5 \pm z_{1-(\alpha/2)} \frac{\sigma}{\sqrt{n}} = 2.58 \frac{7}{\sqrt{10}} = 5.70$$

The result is $\bar{x} = 176.5$, 99% CI [170.80, 182.20]. Comparing this to the 95% CI [172.16, 180.84], we note that the 99% CI is wider. As we are trying to be extra certain (99% of confidence interval will capture the population mean in the long run), we need to widen the interval. In other words, the 99% CI is casting a wider net to catch the population parameter.

My Confidence Interval app below can be used explore confidence intervals and the effect of changing different parameters. If the app does not load in this page, you can view the app here (https://jbaglin.shinyapps.io/Confidence_Intervals/).

Confidence Intervals App

Samples to Simulate

100

Plot 1 n:

10

Plot 1 sig:

0.05

Plot 2 n

10

Plot 2 sig:

0.05

Plot 3 n

10

Plot 3 sig:

0.05

Go!

Go!

Go!

Calculating Confidence Intervals

Pizza Data

This section will compare the mean diameter for pizzas made by Dominos and Eagle Boys. This is an issue close to many students' hearts.

Eagle Boys claim their pizzas are larger than their main competitor, Dominos. The `Pizza` (data/Pizza.csv) dataset contains the diameters (cm) of 125 random pizzas from each company. The dataset is available from the data repository. You can read all about the data here (https://ww2.amstat.org/publications/jse/v20n1/dunn/pizzasize.txt).

Specifically, the `Pizza` (data/Pizza.csv) dataset contains the following variables:

- **ID**: An identifier
- **Store**: The pizza store/company; one of Dominos or EagleBoys
- **Crust**: The crust type for the pizza; DeepPan, Mid and Thin.
- **Topping**: The pizza topping: BBQMeatlovers, Hawaiian and Supreme
- **Diameter**: The pizza diameter in centimetres

Here is a random sample of the full dataset:

Show 10 ▼ **entries** **Search:**

	ID	Store	Crust	Topping	Diameter
1	74	Dominos	Thin	BBQMeatlovers	29.11
2	219	Dominos	Mid	Supreme	26.79
3	18	EagleBoys	Thin	Hawaiian	30.01
4	197	Dominos	Mid	Supreme	27.12
5	5	Dominos	Mid	Hawaiian	26.59
6	98	EagleBoys	DeepPan	Hawaiian	29.68
7	76	Dominos	Mid	Supreme	27.04
8	51	EagleBoys	Mid	Supreme	28.72
9	214	EagleBoys	Thin	Supreme	29.45
10	107	EagleBoys	Mid	BBQMeatlovers	28.86

Showing 1 to 10 of 100 entries

Previous 1 2 3 4 5 … 10 Next

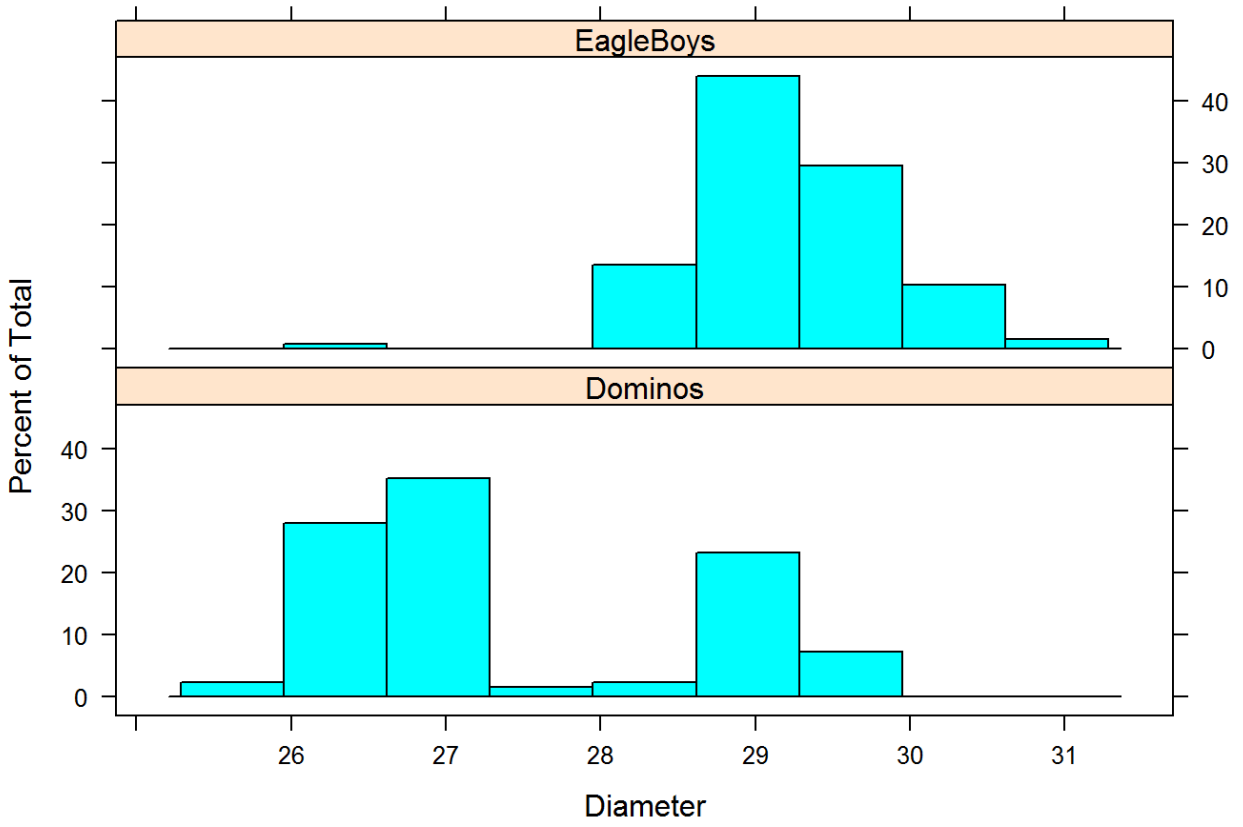
Comparing Means

We start with the descriptive statistics and histograms comparing the pizza diameters (cm) of the two companies.

```
library(dplyr)
Pizza %>% group_by(Store) %>% summarise(Min = min(Diameter,na.rm = TRUE),
                                        Q1 = quantile(Diameter,probs = .
  25,na.rm = TRUE),
                                        Median = median(Diameter, na.rm
   = TRUE),
                                        Q3 = quantile(Diameter,probs = .
  75,na.rm = TRUE),
                                        Max = max(Diameter,na.rm = TRUE
  ),
                                        Mean = mean(Diameter, na.rm = TR
  UE),
                                        SD = sd(Diameter, na.rm = TRUE),
                                        n = n(),
                                        Missing = sum(is.na(Diameter)))
```

```
## # A tibble: 2 x 10
##   Store       Min    Q1 Median    Q3   Max  Mean    SD     n Missing
##   <fct>     <dbl> <dbl>  <dbl> <dbl> <dbl> <dbl> <dbl> <int>   <int>
## 1 Dominos    25.5  26.6   26.9  28.8  29.7  27.4 1.17    125       0
## 2 EagleBoys  26.6  28.8   29.1  29.5  31.1  29.2 0.626   125       0
```

```
library(lattice)
library(ggplot2)
Pizza %>% histogram(~Diameter | Store, data = .,layout=c(1,2))
```

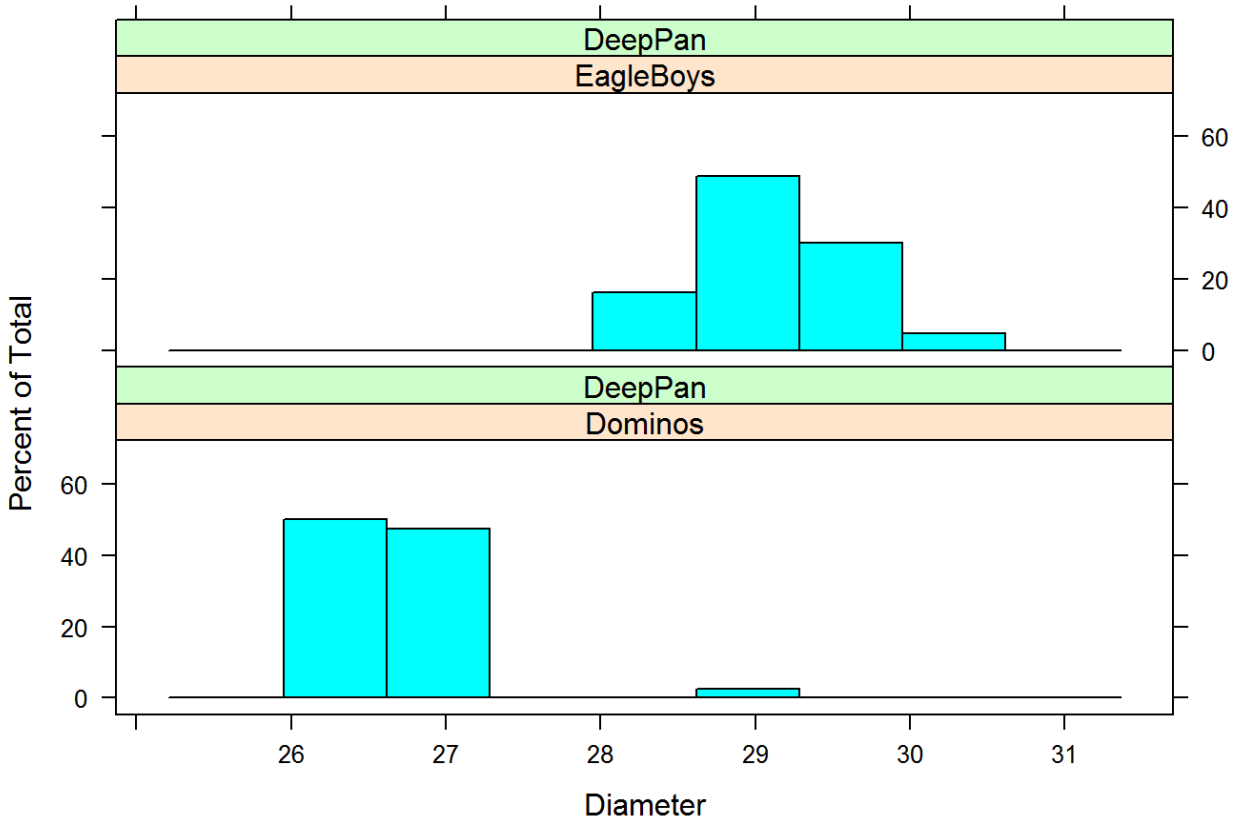


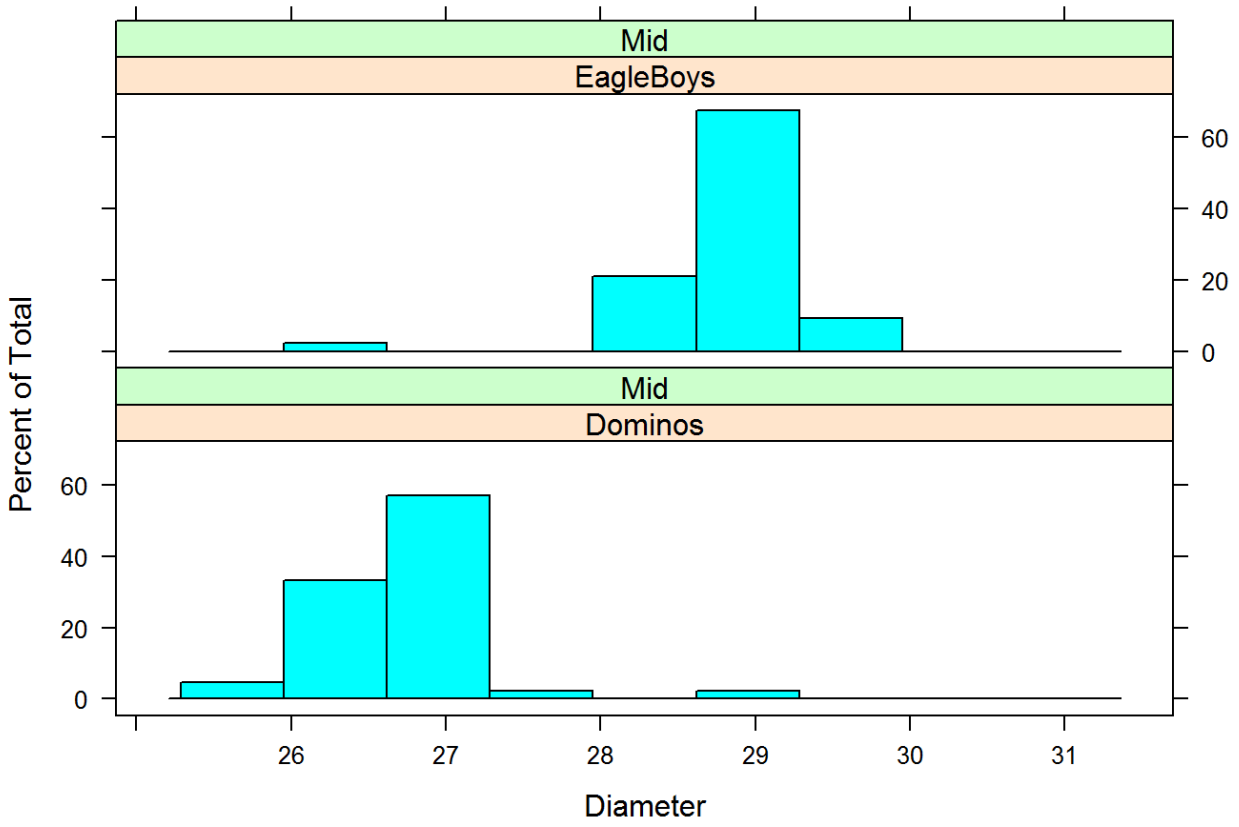
There appears to be a strange looking distribution for the Dominos' pizzas. The distribution appears to be bi-modal. Does the crust type has something to do with it?

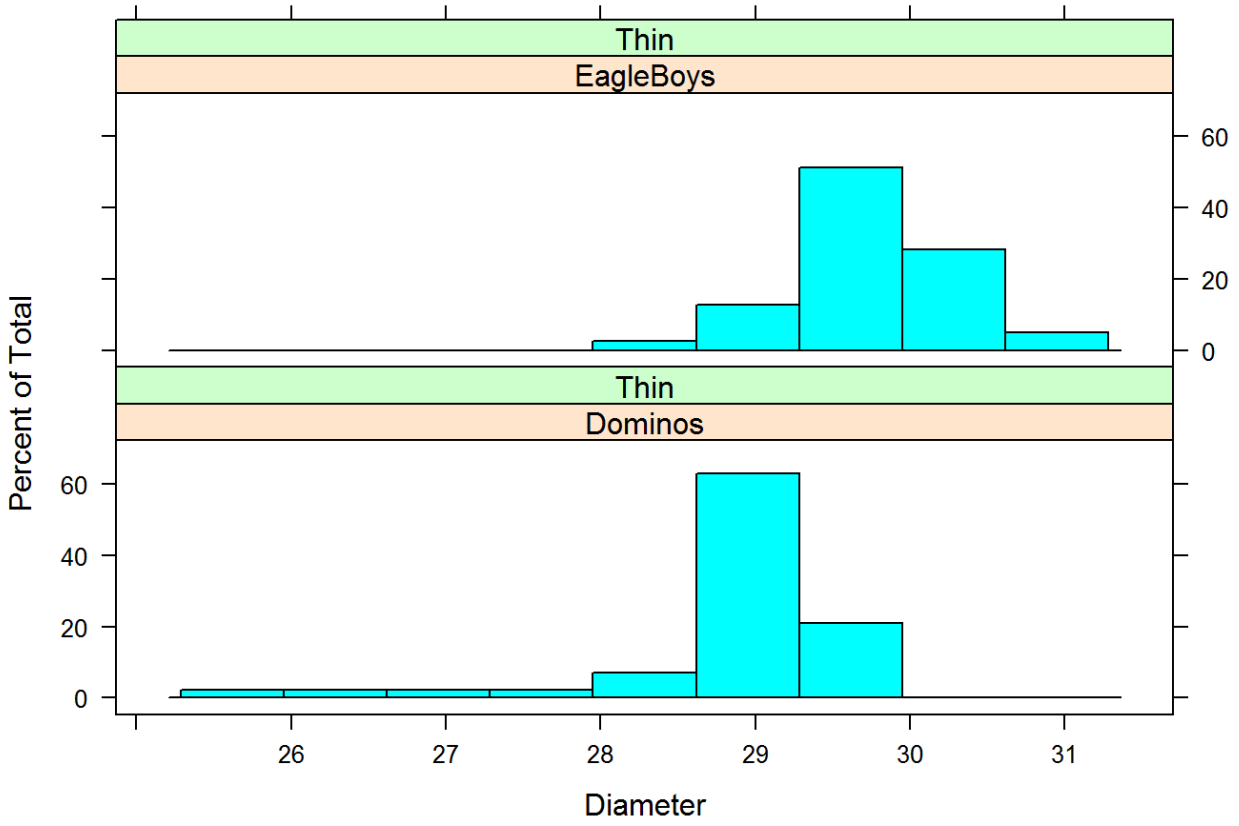
```
Pizza %>% group_by(Store, Crust) %>% summarise(Min = min(Diameter,na.rm =
  TRUE),
                                                 Q1 = quantile(Diameter,probs = .
  25,na.rm = TRUE),
                                                 Median = median(Diameter, na.rm
   = TRUE),
                                                 Q3 = quantile(Diameter,probs = .
  75,na.rm = TRUE),
                                                 Max = max(Diameter,na.rm = TRUE
  ),
                                                 Mean = mean(Diameter, na.rm = TR
  UE),
                                                 SD = sd(Diameter, na.rm = TRUE),
                                                 n = n(),
                                                 Missing = sum(is.na(Diameter)))
```

```
## # A tibble: 6 x 11
## # Groups:   Store [2]
##   Store     Crust     Min    Q1 Median    Q3   Max  Mean    SD     n M
issing
##   <fct>     <fct>   <dbl> <dbl>  <dbl> <dbl> <dbl> <dbl> <dbl> <int>
<int>
## 1 Dominos   DeepPan  26.0  26.4   26.6  26.8  28.8  26.7 0.462    40
0
## 2 Dominos   Mid      25.5  26.6   26.7  27.0  28.9  26.8 0.510    42
0
## 3 Dominos   Thin     25.6  28.8   29.0  29.2  29.7  28.8 0.801    43
0
## 4 EagleBoys DeepPan  28.2  28.7   29.0  29.4  30.4  29.1 0.479    43
0
## 5 EagleBoys Mid      26.6  28.6   28.8  29    29.8  28.8 0.483    43
0
## 6 EagleBoys Thin     28.5  29.4   29.7  30.0  31.1  29.7 0.550    39
0
```

```
library(lattice)
Pizza %>% histogram(~Diameter | Store + Crust, data = .,layout=c(1,2))
```




Mid
EagleBoys
Mid
Dominos
Percent of Total
0
20
40
60
26
27
28
29
30
31
Diameter


Thin
EagleBoys
Thin
Dominos
Percent of Total
0
20
40
60
26
27
28
29
30
31
Diameter

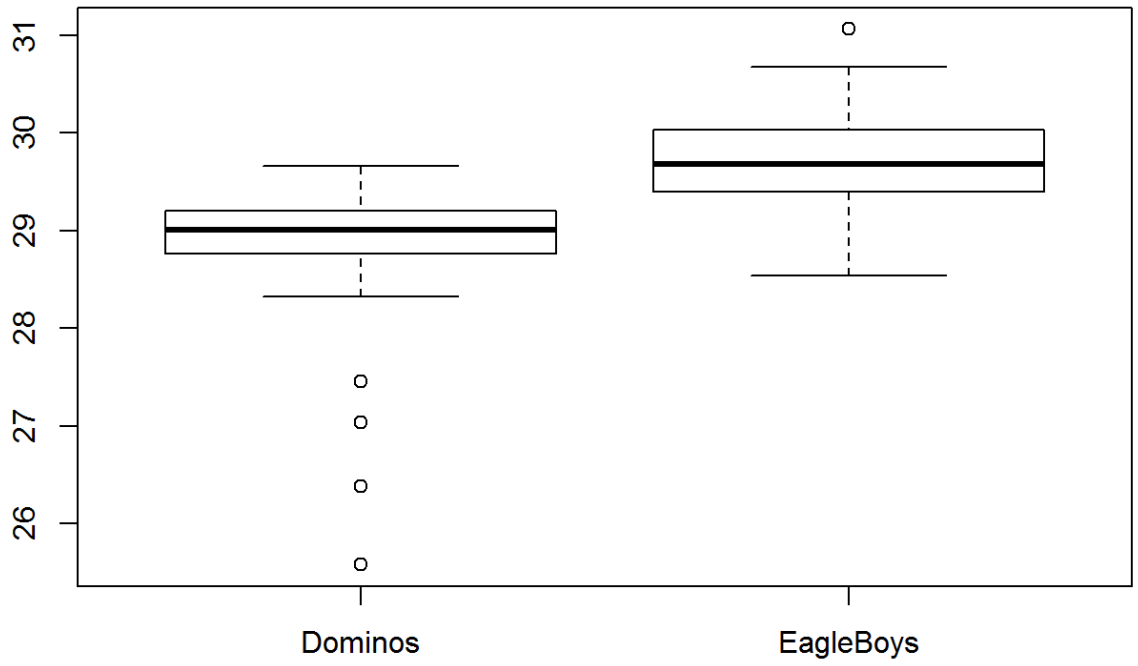
The bi-modal appearance of the Dominos' pizza diameter can be accounted for by the large difference between Dominos' DeepPan and Mid pizzas when compared to their This pizzas. Because of this confounding variable, let's focus on comparing only the thin crusts between Eagle Boys and Dominos.

```
Pizza_thin <- Pizza %>% filter(Crust == "Thin")

Pizza_thin %>% group_by(Store) %>% summarise(Min = min(Diameter,na.rm = T
  RUE),
                                             Q1 = quantile(Diameter,probs = .
  25,na.rm = TRUE),
                                             Median = median(Diameter, na.rm
   = TRUE),
                                             Q3 = quantile(Diameter,probs = .
  75,na.rm = TRUE),
                                             Max = max(Diameter,na.rm = TRUE
  ),
                                             Mean = mean(Diameter, na.rm = TR
  UE),
                                             SD = sd(Diameter, na.rm = TRUE),
                                             n = n(),
                                             Missing = sum(is.na(Diameter)))
```

```
## # A tibble: 2 x 10
##   Store       Min    Q1 Median    Q3   Max  Mean    SD     n Missing
##   <fct>     <dbl> <dbl>  <dbl> <dbl> <dbl> <dbl> <dbl> <int>   <int>
## 1 Dominos    25.6  28.8   29.0  29.2  29.7  28.8 0.801    43       0
## 2 EagleBoys  28.5  29.4   29.7  30.0  31.1  29.7 0.550    39       0
```

```
Pizza_thin %>% boxplot(Diameter ~ Store, data = .)
```



There are also a few outliers that we should pay some attention to. We have four values in the Dominos' group and one in the Eagle Boys' group. These outliers can have unwanted effects on our estimates, so it is best to deal with them at this point.

There are a few ways that we could do this. A simple approach is to exclude them. When you exclude outliers, you must be sure to explain why you did so. In this example, we will assume that these were due to poor measurements.

We can use the following code to remove outliers using the following definition reported back in Module 2.

$$\text{Lower outlier} < Q_1 - 1.5 * IQR$$

$$\text{Upper outlier} > Q_3 + 1.5 * IQR$$

The following code exploits the `boxplot()` function which saves a list of the outliers plotted in the box plot.

```
boxplot <- Pizza_thin %>% boxplot(Diameter ~ Store, data = ., plot = FALS
  E)
boxplot
```

```
## $stats
##        [,1]  [,2]
## [1,] 28.320 28.54
## [2,] 28.765 29.40
## [3,] 29.010 29.68
## [4,] 29.200 30.03
## [5,] 29.660 30.67
##
## $n
## [1] 43 39
##
## $conf
##          [,1]     [,2]
## [1,] 28.90519 29.52061
## [2,] 29.11481 29.83939
##
## $out
## [1] 27.04 25.58 27.45 26.38 31.06
##
## $group
## [1] 1 1 1 1 2
##
## $names
## [1] "Dominos"   "EagleBoys"
```

Now we can create a filter matrix to identify and remove the outliers in `Pizza_thin`.

```
Filt_mat <- data.frame(group = boxplot$group, outliers = boxplot$out)
Filt_mat$group <- Filt_mat$group %>% factor(levels = c(1,2),
                                            labels = c("Dominos","EagleBo
  ys"))
Filt_mat
```

```
##       group outliers
## 1   Dominos    27.04
## 2   Dominos    25.58
## 3   Dominos    27.45
## 4   Dominos    26.38
## 5 EagleBoys    31.06
```

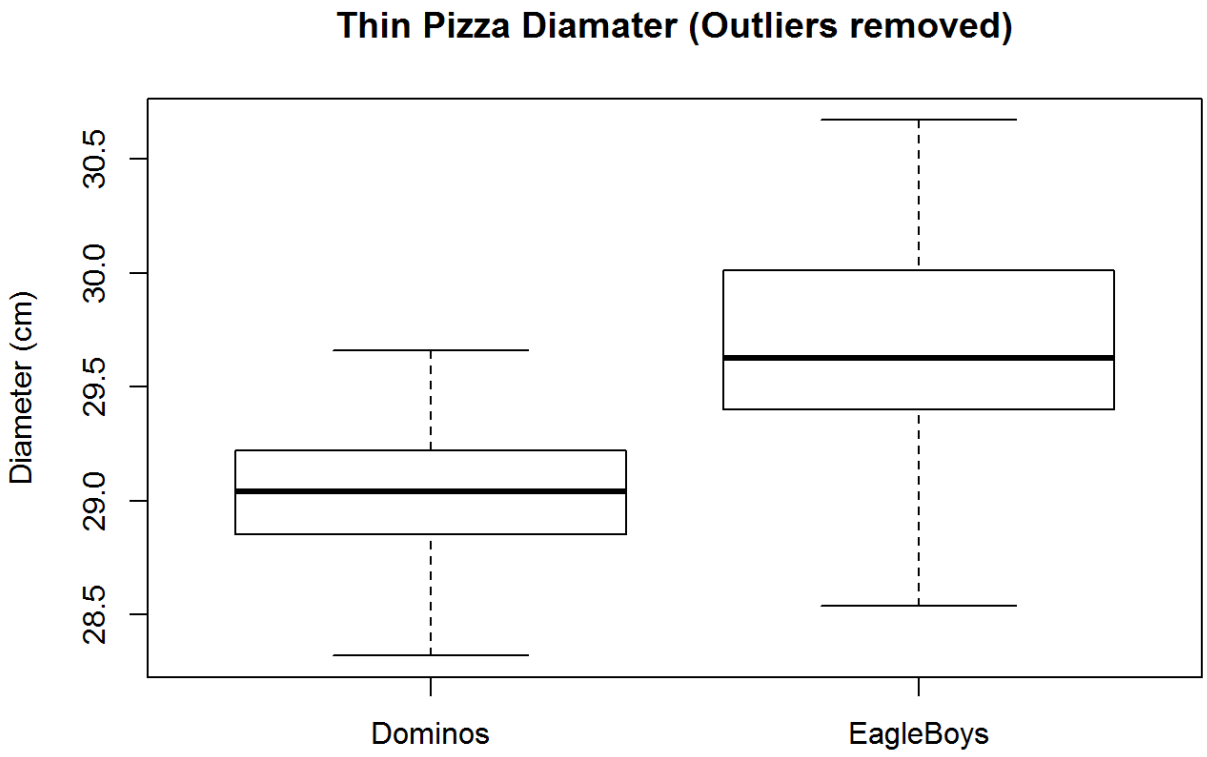
Now we can use `dplyr` and `%in%` operator to filter the outliers.

```
Pizza_thin_filt <- Pizza_thin %>% filter(!(Store %in% Filt_mat$group)
                                         | !(Diameter %in% Filt_mat$outlie
  rs))
```

This filter uses the filter matrix to identify values that are not considered outliers. The `!` symbol mean "not" and the `%in%` operator check a range of values in a vector. Therefore the filter systematically searches the `Pizza_thin` dataset for outliers and filters them out, saving the results to `Pizza_thin_filt`.

Now let's recheck the box plot:

```
Pizza_thin_filt %>% boxplot(Diameter ~ Store, data = .,
                            main = "Thin Pizza Diamater (Outliers remove
  d)",
                            ylab = "Diameter (cm)")
```



```
Pizza_thin_filt %>% group_by(Store) %>% summarise(Min = min(Diameter,na.r
  m = TRUE),
                                                  Q1 = quantile(Diameter,probs = .
  25,na.rm = TRUE),
                                                  Median = median(Diameter, na.rm
   = TRUE),
                                                  Q3 = quantile(Diameter,probs = .
  75,na.rm = TRUE),
                                                  Max = max(Diameter,na.rm = TRUE
  ),
                                                  Mean = mean(Diameter, na.rm = TR
  UE),
                                                  SD = sd(Diameter, na.rm = TRUE),
                                                  n = n(),
                                                  Missing = sum(is.na(Diameter)))
```

```
## # A tibble: 2 x 10
##   Store       Min    Q1 Median    Q3   Max  Mean    SD     n Missing
##   <fct>     <dbl> <dbl>  <dbl> <dbl> <dbl> <dbl> <dbl> <int>   <int>
## 1 Dominos    28.3  28.8   29.0  29.2  29.7  29.0 0.307    39       0
## 2 EagleBoys  28.5  29.4   29.6  30.0  30.7  29.7 0.509    38       0
```

Sometimes when you are dealing with very large datasets, removing one set of outliers, can create additional outliers in the filtered dataset. This is due to the outlier definition being based on the estimates of the IQR and quartiles. When you remove outliers, you are changing these estimates, which then changes the location of your fences. This is a major problem when dealing with highly skewed distributions. If you're ever faced with this situation, look into applying a Box-cox power transformation (https://stat.ethz.ch/R-manual/R-devel/library/MASS/html/boxcox.html) prior to outlier detection and removal. The Box-Cox transformations do an amazing job of normalising distributions, which can greatly assist with outlier detection.

Eagle Boys do appear to have larger diameters. However, the data were taken from a random sample of 77 pizzas across both stores. Before drawing conclusions, we need to quantify our uncertainty of our statistical estimates.

Mean - Unknown Population Standard Deviation

In the first example of CIs, we made an unrealistic assumption that the population standard deviation, σ, is known for a normally distributed variable. This allows us to use the standard normal distribution to calculate the CIs. In real research, σ is rarely known and must be estimated from the sample standard deviation, s. As we are estimating two parameters from the sample, the population mean, μ, and standard deviation, σ, we need to take into account the extra uncertainty or error associated with s to ensure the expected coverage of the CI remains at the desired level, e.g. 95%. The family of t-distributions are used for this purpose.

The t-distribution has an extra parameter, called degrees of freedom, df, that can be altered to change the heaviness of the distribution's tails. Degrees of freedom for a t-distribution are typically calculated as:

$$df = n - 1$$

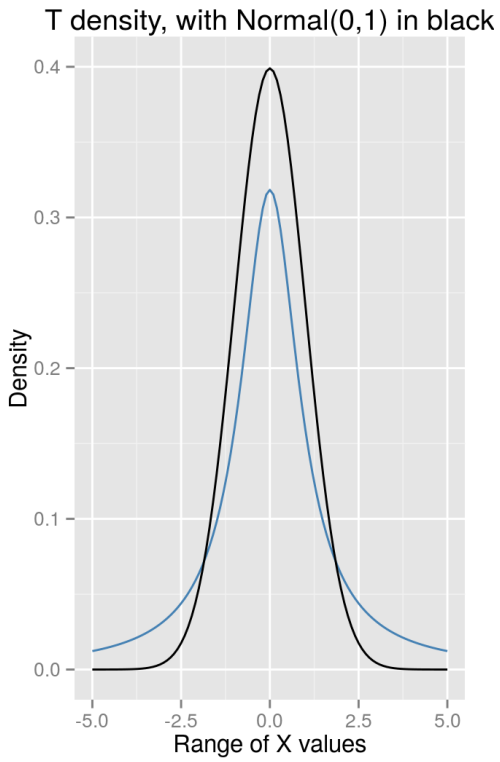
The following Shiny app compares the t-distribution for varying values of df to a standard normal distribution, $N(0, 1)$. Move the df slider from left to right to explore how the t-distribution changes based on df. The t-distribution looks very flat in comparison to a normal distribution when the df values are low. However, as df increases (i.e. sample size increases), the t-distribution will start to approximate a normal distribution. In fact, for sample sizes where df > 30, there is very little practical difference between them. As df approaches infinity, the t-distribution will become a normal distribution.

Interactive Student's T Density

Move the slider to change the degrees of freedom

1 30

Though the t with one degree of freedom looks different from the Normal(0,1), as the degrees of freedom increase, it becomes very close to a standard Normal



In all circumstances, the tails of the T distribution are "fatter," which means outliers are more likely under the T distribution.

In the Pizza example, we don't know the population standard deviation, so we have to estimate it using the sample. Do we need to worry about normality of the data to consider whether the confidence interval formula used previously is appropriate? The answer to this question depends on the sample size. If the sample size was small (e.g. $n < 30$), the non-normality of the data would prevent us from applying a regular CI formula. However, thanks to the larger sample size ($n = 39$ for Dominos and $n = 38$ for Eagle Boys) and the CLT introduced in the previous module, the standard CI formula works quite well. We only need to adjust the formula slightly to take into account that we don't know the population standard deviation.

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

Notice the inclusion of $t_n - 1, 1 - (\alpha/2)$. Instead of looking this value up using a normal distribution, we need to use a different formula in R namely `qt()`. For example, for Dominos' thin pizza (outliers removed):

```
qt(p = 0.975, df = 39 -1)
```

```
## [1] 2.024394
```

$$29.04 \pm 2.024 \frac{0.306}{\sqrt{39}} = 099$$

and for Eagle Boys:

```
qt(p = 0.975, df = 38 -1)
```

```
## [1] 2.026192
```

$$29.66 \pm 2.026 \frac{0.509}{\sqrt{38}} = 0.167$$

Store	Mean	SD	n	tcrit	SE	95 % CI
Dominos	29.04	0.307	39	2.024	0.049	[28.94, 29.14]
EagleBoys	29.66	0.509	38	2.026	0.083	[29.49, 29.83]

As the population standard deviation is rarely known, we should always default to using the t-distribution and t-critical values to calculate interval estimates for sample means (assuming Normality or the CLT applies). Only use a z-critical value if the population standard deviation is given (which is rare!).

We can use `dplyr` and some quick computations to generate tables of 95% CIs for means...

```
Pizza_thin_filt %>% group_by(Store) %>% summarise(Mean = round(mean(Diame
  ter, na.rm = TRUE),2),
                                                  SD = round(sd(Diameter,
  na.rm = TRUE),3),
                                                  n = n(),
                                                  tcrit = round(qt(p = 0.
  975, df = n - 1),3),
                                                  SE = round(SD/sqrt(n),3
  ),
                                                  `95% CI Lower Bound` =
  round(Mean - tcrit * SE,2),
                                                  `95% CI Upper Bound` =
  round(Mean + tcrit * SE,2))
```

```
## # A tibble: 2 x 8
##   Store     Mean    SD     n tcrit    SE `95% CI Lower Boun~ `95% CI U
pper Boun~
##   <fct>    <dbl> <dbl> <int> <dbl> <dbl>               <dbl>
<dbl>
## 1 Dominos   29.0 0.307    39  2.02 0.049                28.9
29.1
## 2 EagleBo~  29.7 0.509    38  2.03 0.083                29.5
29.8
```

We can also use the `t-test` if we filter the dataset first:

```
Dominos <- Pizza_thin_filt %>% filter(Store == "Dominos")
t.test(Dominos$Diameter)
```

```
##
##  One Sample t-test
##
## data:  Dominos$Diameter
## t = 591.66, df = 38, p-value < 2.2e-16
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
##  28.94089 29.13962
## sample estimates:
## mean of x
##  29.04026
```

```
EagleBoys <- Pizza_thin_filt %>% filter(Store == "EagleBoys")
t.test(EagleBoys$Diameter)
```

```
##
##  One Sample t-test
##
## data:  EagleBoys$Diameter
## t = 359.09, df = 37, p-value < 2.2e-16
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
##  29.49735 29.83212
## sample estimates:
## mean of x
##  29.66474
```

If you want to change the confidence level to 99%:

```
Dominos <- Pizza_thin_filt %>% filter(Store == "Dominos")
t.test(Dominos$Diameter, conf.level=0.99)
```

```
##
##  One Sample t-test
##
## data:  Dominos$Diameter
## t = 591.66, df = 38, p-value < 2.2e-16
## alternative hypothesis: true mean is not equal to 0
## 99 percent confidence interval:
##  28.90717 29.17335
## sample estimates:
## mean of x
##  29.04026
```

```
EagleBoys <- Pizza_thin_filt %>% filter(Store == "EagleBoys")
t.test(EagleBoys$Diameter, conf.level=0.99)
```

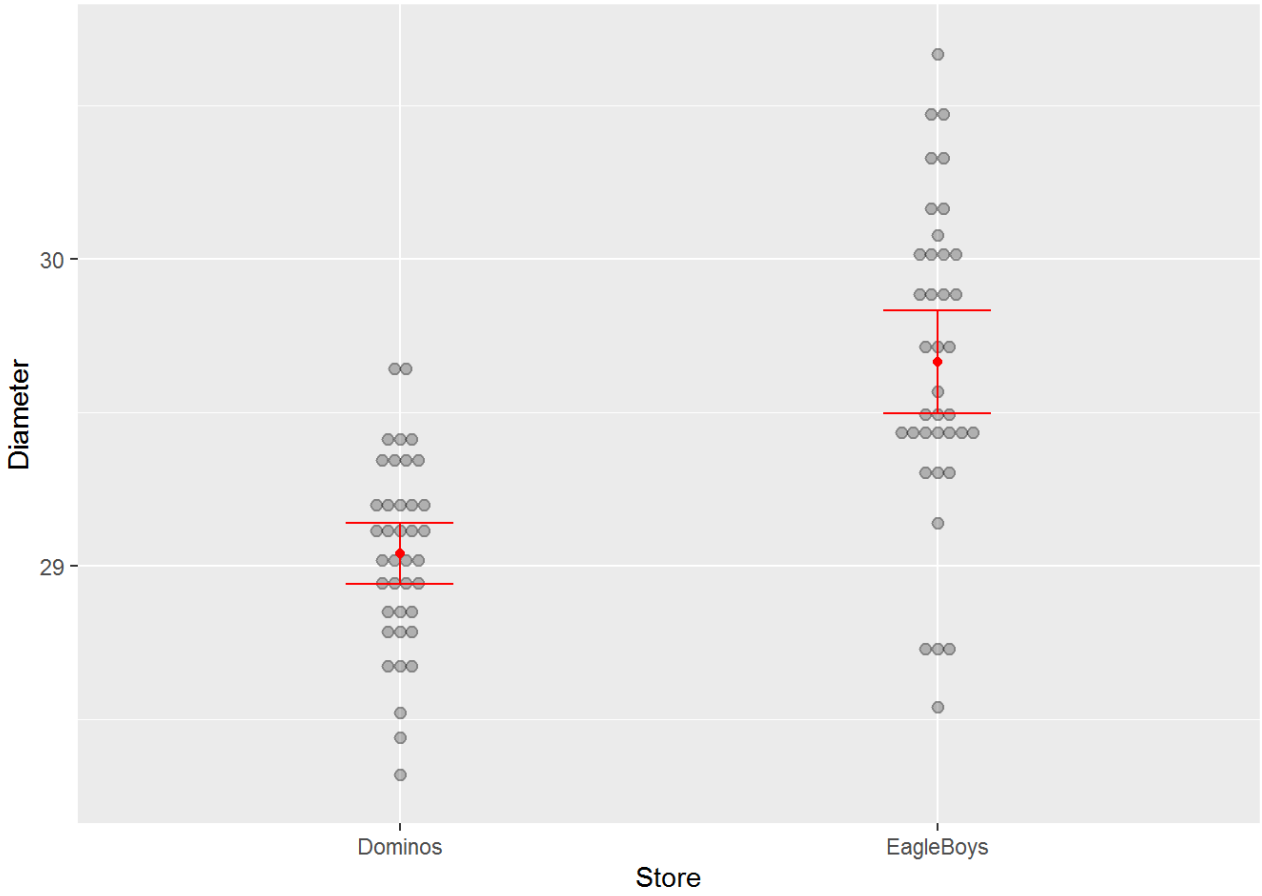
```
##
##  One Sample t-test
##
## data:  EagleBoys$Diameter
## t = 359.09, df = 37, p-value < 2.2e-16
## alternative hypothesis: true mean is not equal to 0
## 99 percent confidence interval:
##  29.44042 29.88906
## sample estimates:
## mean of x
##  29.66474
```

Visualising confidence intervals can be tricky. I recommend you display the confidence intervals overlaid upon the data. This ensures the viewer does not confuse the 95% CI as the range of the data. Here's some code that you can adapt to different situations. The visualisation makes use of the `ggplot2` and `Hmisc` package.

```
#install.packages("ggplot2") # If required
#install.packages("Hmisc") # If required

library(ggplot2)
library(Hmisc)

p1 <- ggplot(data = Pizza_thin_filt, aes(x = Store, y = Diameter))
p1 + geom_dotplot(binaxis = "y", stackdir = "center", dotsize = 1/2, alph
  a = .25) +
  stat_summary(fun.y = "mean", geom = "point", colour = "red") +
  stat_summary(fun.data = "mean_cl_normal", colour = "red",
               geom = "errorbar", width = .2)
```



Based on the 95% CIs for the mean pizza diameters of the competing stores, I think we can be very confident that Eagle Boy's pizzas tend to have a wider mean diameter.

Proportions

Interval estimates for other statistics can also be calculated using the following formulas. However, be warned. You can only guarantee the appropriate confidence interval converge will be met if the stated assumptions are satisfied. These methods are referred to as approximations. When approximations cannot be used, we must look to exact methods. Fortunately, R has you covered for both.

Interval estimates for proportions are based on a normal approximation to the binomial distribution. If $np(1-p) \geq 5$, we can approximate a CI using:

$$p \pm z_{1-(\alpha/2)}\sqrt{\frac{p(1-p)}{n}}$$

For example, suppose a researcher randomly selects 300 people from the population and finds that 2% ($x = 6$) have cancer. We confirm,
$np(1-p) = 300 * .02 * .98 = 5.88 \geq 5$. Therefore, we can proceed with the normal approximated CI. We will use the standard 95% CI.

$$.02 \pm z_{1-(\alpha/2)}\sqrt{\frac{p(1-p)}{n}} = 1.96\sqrt{\frac{.02(1-.02)}{300}} = 0.016$$

We calculate the prevalence of cancer to be 0.02, 95% CI [.004, .036].

In R, we can use the `binom.conf.int()` command from the `epitools` package to quickly calculate the 95% CI.

```
#install.packages("epitools") #If required
library(epitools)
binom.approx(6, 300, conf.level = 0.95)
```

```
##   x   n proportion     lower     upper conf.level
## 1 6 300       0.02 0.0041578 0.0358422       0.95
```

What do you do if $np(1-p) < 5$? For example, say the above example was changed so $n = 20, p = .3$ $(x = 6)$ and $1 - p = .7$, $np(1-p) = 20 * .3 * .70 = 4.2$. Therefore, the normal approximation does not apply. We can use the `binom.exact()` function to deal with this situation.

```
binom.exact(6, 20, conf.level = 0.95)
```

```
##   x  n proportion     lower     upper conf.level
## 1 6 20        0.3 0.1189316 0.5427892       0.95
```

Rates

Confidence intervals for rates following a Poisson distribution can be readily obtained using the `pois.conf.int()` function, which is also part of the `epitools` package. For example, we can confirm from the output below that the 95% CI for $\lambda = 56$ is [42.30, 72.72].

```
pois.exact(56, pt = 1, conf.level = 0.95)
```

```
##    x pt rate    lower    upper conf.level
## 1 56  1   56 42.30181 72.72068       0.95
```

When $\lambda > 100$, we can use the following normal approximation:

$$\lambda \pm z_{1-(\alpha/2)}\sqrt{\lambda}$$

Suppose a sample's point estimate is $\lambda = 145$. A normal approximated 95% CI for λ is calculated by:

$$145 \pm 1.96\sqrt{145} = 23.6$$

Therefore, $\lambda = 145$, 95% CI [121.4, 168.6]. Alternatively, in R:

```
pois.approx(145, pt = 1, conf.level = 0.95)
```

```
##     x pt rate    lower    upper conf.level
## 1 145  1  145 121.3989 168.6011       0.95
```

References