Confidence Intervals

Marc Los Huertos

October 14, 2019

An important considerations with univariate data is the development of confidence intervals and forms the foundation for hypothesis testing. As you can probably guess, confidence intervals are based on the distribution of the data. Data with lots of variability will have wide confidence intervals, while data with low variability will have narrow confidence intervals.

There are two main approaches in developing confidence intervals. One relies on the use of theoretical probability distributions. Another method uses the data themselves to create randomized sample subsets (bootstrapping) to create confidence intervals. We will learn several methods because each are used in environmental sciences to varying degrees. I have decribed the former in this handout, but may add bootstrapping soon.

Why Confidence Intervals?

Confidence intervals are used to indicate the reliability of an estimate. How likely the interval is to contain the parameter is determined by the confidence level. Increasing the desired confidence level will widen the confidence interval (see Figure 5). Based on a foundational understanding of confidence intervals allows us to extend our interpretations of the data toward a hypothesis testing perspective, which we'll get into more during the semester.

Where are Confidence Intervals Used?

Confidence intervals can be calculated for a range of statistics, including the mean, slope and intercept of a linear model, among other things. We'll concentrate on the what confidence intervals of the population and sample mean at this point, but keep in mind the concept can be applied to many parameter estimates.

MIKE LUCKOVICH
ATLANTA JOURNAL-CONSTITUTION
© 2002
AJC.com
100% SURE WE'RE DOOMED
WALL ST
I'VE NEVER SEEN INVESTOR CONFIDENCE HIGHER....

Figure 1: Confidence abounds without bounds.

Defining Some Terms

Populations and Samples

In general, environmental scientists can't measure the entire population. For example, a complete audit of all the recycling would be impossible for our class! Instead, we sample from the population. Statisticians have have developed semi-consistant symbology for various statistics based on populations and samples.

A population mean for example is usually referred to by the Greek letter, μ. While for a sample, the mean might be referred to as $\bar{y}$ (or $\bar{x}$). The spread of data, or variance, in a population is referred to as σ^2, again a Greek letter. The population variance is often referred to as s^2. In general, we don't know μ or σ^2, so we can estimate them and develop confidence intervals that probably include the true μ. Notice the word, "probably." In other words, we our intervals in terms of probabilities![1]

The difference between μ and $\bar{x}$

To illustrate the difference, let's create a dataset of 1000 random numbers from a normal distribution with a mean of 10 and standard deviation of 1. We'll let this represent an entire population.[2]

```
set.seed(123)
N1000 = round(rnorm(n= 1000, mean = 10, sd = 1), 1)
```

The distribution of the data (Figure 2), resembles a normal distribution, which is reassuring since we created it using a random number generator that uses the normal probability distribution.

These data will represent the population (N) and has a mean or μ of 10.02. The values range from 7.2 to 13.2 (Figure 2). From this dataset, we can create several dataset sets by random sampling the vector with different n (n = 100, n = 20, n = 10, n = 5).

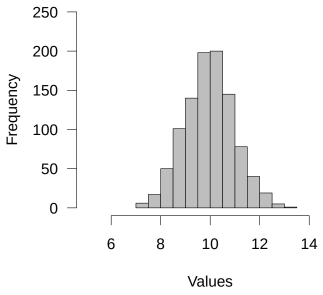


Figure 2: Frequency distribution of population.

```
n100 = sample(N1000, size = 100)
n20 = sample(N1000, size = 20)
n10 = sample(N1000, size = 10)
n5 = sample(N1000, size = 5)
```

With a size or n of ten values from our population, I have obtained the following result:

```
n10

##  [1] 10.2 11.4 10.1  9.3  9.8  9.4 10.6 11.0 10.4 10.0
```

and a mean or $\bar{y}$ = 10.22. These values certainly fall within the distribution of the population (Figure 2).

R's Default Boxplot

One of the confusing aspects when learning R and statistics at the same time is the range of subtle discrepencies between the vocabulary in statistics, parameter symbology, and how R implments statistics. But just as we learn a language we use content (e.g. literature),

[1] When you hear the word probably, it might be useful to think about how this implies a probability distribution.

[2] I rounded the values to the nearest tenth, to keep the printouts more managable.

I have found it best to learn statistics while learning to use the R language.

Nevertheless, the subtlties land hard on us when we are just trying to keep up with the statistical concepts. The `boxplot` function in R is easy to make and provides a lot of information, but it's not always clear what is being displayed. For example, the `boxplot()` function displays the median (for our ten sample vector = 10.15 and not the mean (10.22). In Figure 3A, the boxplot displays the median, interquarile range, and range. But in many cases, the sample might have some points that seem to be way outside the normal range. In those cases R creates a figure with a more complex set of rules. We won't go into these at this point, but not the differences between Figure 3A and Figure 3B.

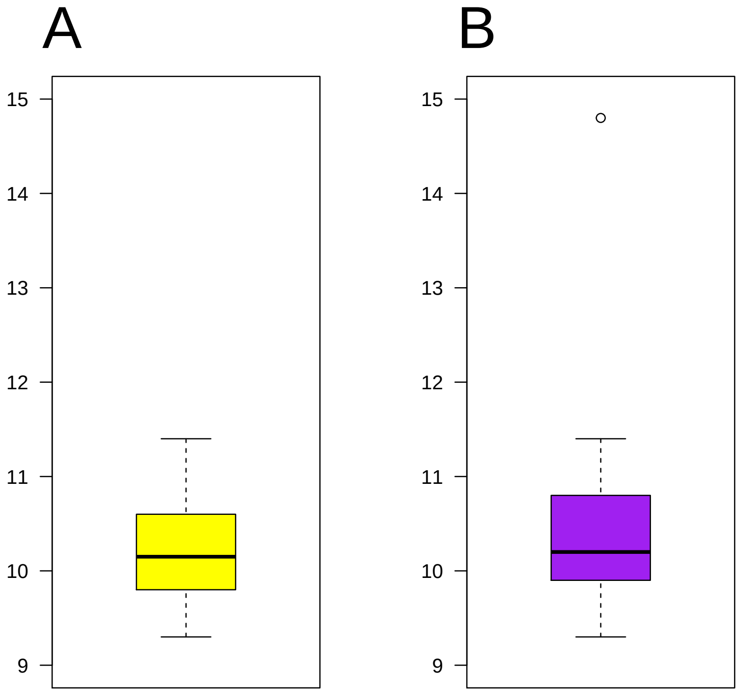


Figure 3: Boxplot where panel A displays the following information for our 10 sample vector: median =10.15; Range = 9.3, 11.4; and interquartile range = 9.85, 10.55. Panel B identifies an "outlier" using obscure rules and changing the range to exclude the outlier. I included the value of 14.8 with the vector to demonstrate this change. Weird. Note: I rotated the y-axis labels to make it easier to read. I try to do this with all my figures to make it easier to read.

In the end, the `boxplot()` function is very useful to quickly summarize the data, but in most cases, we tend to describe the data using a more precise set of measures. The standard deviation is the most commonly used parameter to measure the variability or spread of the data.

Exploring the Waste Audit Data

Let's 'read' the data into r, with the `read.csv()` function.

```
Unsorted.csv = "/home/CAMPUS/mwl04747/github/beginnersluck/Confidence_Intervals/2019_EA30F_Waste_Audit_Uns

Sorted.csv = "/home/CAMPUS/mwl04747/github/beginnersluck/Confidence_Intervals/2019_EA30F_Waste_Audit_Sorte
# Read Raw Data
Unsorted = read.csv(Unsorted.csv)
Sorted = read.csv(Sorted.csv)
```

Now that we have imported the data into R, we will not process some of the data to prepare for the analysis. For example, let's caculate the percentage of sorted items, remove the plastic film category, and shorten the compostable name to simply compost.

```
Sorted$Percent = (Sorted$NetMass/Sorted$Total)*100
Sorted = subset(Sorted, subset=Type!="Plastic Film")
levels(Sorted$Type)[levels(Sorted$Type)=="Compostable"] <- "Compost"
```

Be sure to check the results as you go and convince yourself how these work by looking online to see how these functions work.

Anyone who has worked with quantitiative data knows that data entry errors can be a major headache if they are not caught early. Thus, we'll use a couple of methods to evaluate potential data entry errors.

In the case of Figure 4, We can see that the trash is relatively high in both types of unsorted bags, e.g. there a fair amount trash in both the recyling and trash containers. But I wonder if the overlapping interval is statistically significant? If they are the same, then the percentage of trash is being put in both containers without regard of the intended use. The boxplot is useful to start with, but we can't draw statistical conclusions on this yet.

Measuring the Spread

We can calculate the variance of the sample using the following formula:

$$s^2 = \frac{\sum(y_i - \bar{y})^2}{n-1} \tag{1}$$

Let's apply this equation, so you can see how it works! First, we'll take the difference between a measurment and the mean.[3] Let's start with the first observation:

[3] We will use the vector index to subset each variable – see "Blue Book" for indexing if you want more information on this procedure.

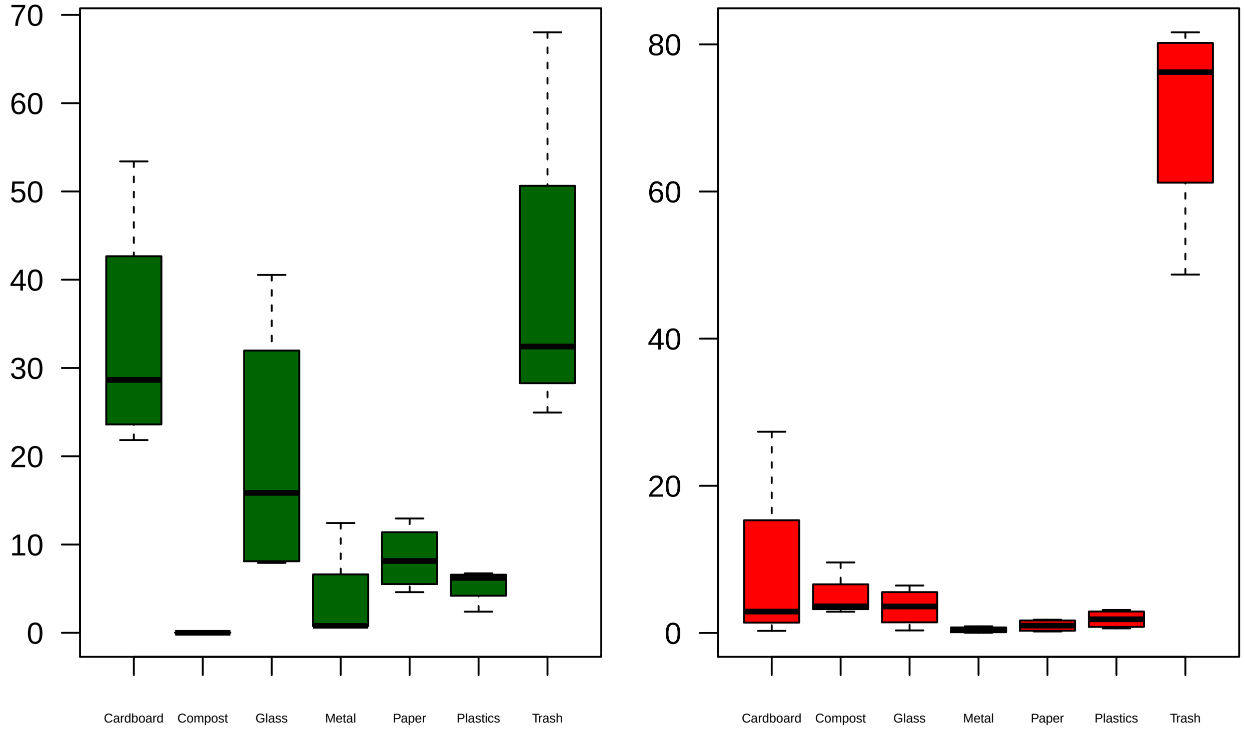


Figure 4: Sorted Percent Mass from Recycling Bags

```
n10[1] - mean(n10)

## [1] -0.02
```

Next we square the difference[4]

```
(n10[1] - mean(n10))^2

## [1] 4e-04
```

Now we'll do this for each of the remaining observations:

```
(n10[2] - mean(n10))^2

## [1] 1.3924

(n10[3] - mean(n10))^2

## [1] 0.0144

(n10[4] - mean(n10))^2

## [1] 0.8464

(n10[4] - mean(n10))^2

## [1] 0.8464
```

These are "deviates." Then we add the deviates together (which is what the Σ symbol is tell us to do). We'll call the sum of squared deviates:

```
(sum_of_squared_deviates = (n10[1] - mean(n10))^2 +
   (n10[2] - mean(n10))^2 + (n10[3] - mean(n10))^2 +
   (n10[4] - mean(n10))^2 + (n10[5] - mean(n10))^2 +
   (n10[6] - mean(n10))^2 + (n10[7] - mean(n10))^2 +
   (n10[8] - mean(n10))^2 + (n10[9] - mean(n10))^2 +
   (n10[10] - mean(n10))^2)

## [1] 3.936
```

We take the sum of these and divide by the degrees of freedom.[5]

```
sum_of_squared_deviates/(10-1)

## [1] 0.4373333

sqrt(sum_of_squared_deviates/(10-1))

## [1] 0.6613118
```

[4] R did scientific notation on this, I'll need to fix this in the near future.

[5] Degree of freedom is extremely complicated to explain without tons of theory. For now, just rest assured that it works. If we have time, we'll delve into this more.

Thankfully, we don't need to use all of this tedious code to generate the variance and standard deviation. We can use `var()` and `sd()`.

```
var(n10)

## [1] 0.4373333

sd(n10)

## [1] 0.6613118
```

This value is the sample variance. The s^2 for our sample is 0.44. The standard deviation is the $\sqrt{(s^2)}$ or just s. For our sample, the standard deviation is 0.66.

Then, we can calculate the standard error of the sample using the following equation:

$$\text{Standard Error of the Sample} = SE_s = \sigma / \sqrt{n} \tag{2}$$

```
#computation of the standard error of the sample mean
sem<-sd(n10)/sqrt(length(n10))
```

NOTE: If the sample size approaches the size of the population, we must use a "finite-population correction". Apparently the Central Limit Theorum gets wonky and the correction factor ensures better estimates. We don't need to worry about in in our case.

Now, it's important to note that we have calculated sample estimates. We can also calculate the population parameters as well.

Parameter	Symbol	Formula	Population value
Mean	μ	$\Sigma x_i / n$	10.02
Variance	σ^2	$\Sigma(x_i - \mu)^2 / n$	0.99
Standard Deviation	σ	$\sqrt{(\Sigma(x_i - \mu)^2 / n)}$	0.99

The Standard Deviation versus Standard Error

The terms “standard error” and “standard deviation” are often confused. The contrast between these two terms reflects the important distinction between data description and inference, we should appreciate.

The standard deviation (often s) is a measure of variability. When we calculate the standard deviation of a sample (which we did above), we are using it as an **estimate of the variability of the population** from which the sample was drawn. For normally distributed data the standard deviation has some extra information, namely the

68-95-99.7 rule which tells us the percentage of data lying within 1, 2 or 3 standard deviation from the mean. For now, try to remember about 95% of individuals will have values within 2 standard deviations of the mean, the other % being equally scattered above and below these limits.

Contrary to popular misconception, the standard deviation is a valid measure of variability regardless of the distribution. About 95% of observations of any distribution usually fall within the 2 standard deviation limits, though those outside may all be at one end. We may choose a different summary statistic, however, when data have a skewed distribution.

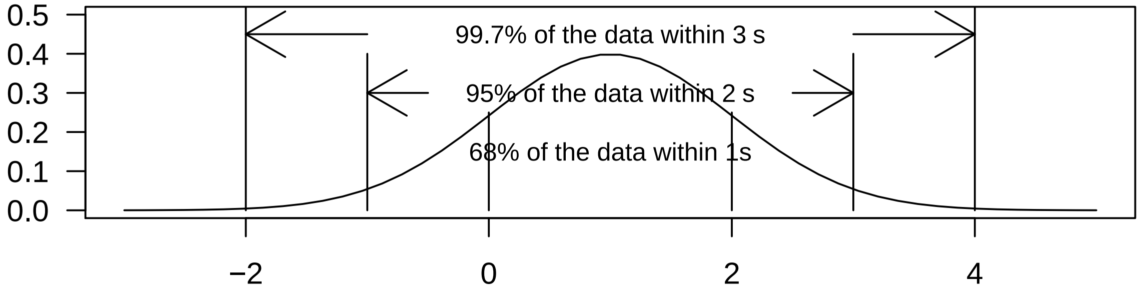


Figure 5: The variation of the population can be estimated with s, standard deviation.

When we calculate the sample mean we are usually interested not in the mean of this particular sample, but in the mean for individuals of this type—in statistical terms, of the population from which the sample comes. We usually collect data in order to generalise from them and so use the sample mean as an estimate of the mean for the whole population. Now the sample mean will vary from sample to sample; the way this variation occurs is described by the "sampling distribution" of the mean. We can estimate how much sample means will vary from the standard deviation of this sampling distribution, which we call the standard error (SE) of the estimate of the mean. As the standard error is a type of standard deviation, confusion is understandable. Another way of considering the **standard error is as a measure of the precision of the sample mean**.

The standard error of the sample mean depends on both the standard deviation and the sample size, by the simple relation $SE = SD/\sqrt{\text{sample size}}$. The standard error falls as the sample size increases, as the extent of chance variation is reduced—this idea underlies the sample size calculation for a controlled trial, for example. By contrast the standard deviation will not tend to change as we increase the size of our sample.

Selecting α: The Level of Confidence

An important to note that the selection of the interval depends on your decision as to what level of confidence you want. Since probability ranges from 0 to 1, the confidence intervals of the parameter can also vary within this range. However, we usually are trying to constrain the confidence interval to something narrow, for example, we usually specify confidence intervals of 0.90, 0.95, and 0.99.

In the context of a probability density function, these levels correspond to percentages of the area of the normal density curve. For example, a 95% confidence interval includes 95% of the normal curve's area – the probability of observing a value outside of this area is less than 0.05. So, following standards in statistics, we use α to signify the as the criterion, such that

$$\text{Confidence Interval \%} = 100 * (1 - \alpha) \tag{3}$$

Yet, this is still ambiguous. Because the normal curve is symmetric, half of the area is in the left tail of the curve, and the other half of the area is in the right tail of the curve. Thus, if we want to generate confidence intervals that cover both tails of the curve we need to split α for each side of curve. Thus for a for a 95% confidence interval, the area in each tail is equal to 0.05/2 = 0.025.[6]

[6] A figure here would be helpful!

Estimating Confidence Intervals for Waste Audit – Preview

Calculating Summary Statitics

First, we'll create summary statistics by each type of material:

```
Sorted.mean <- aggregate(Sorted$Percent,
  by=list(Type = Sorted$Type,  Trash_Recycle = Sorted$Trash_Recycle),
  mean)
Sorted.sd <- aggregate(Sorted$Percent,
  by=list(Type = Sorted$Type, Trash_Recycle = Sorted$Trash_Recycle),
  sd)
Sorted.n <- aggregate(Sorted$Percent,
  by=list(Type = Sorted$Type, Trash_Recycle = Sorted$Trash_Recycle),
  length)
```

Here's the output of one of these lines:

```
Sorted.mean

##          Type Trash_Recycle           x
## 1  Cardboard             R 33.13571287
## 2    Compost             R  0.01949318
## 3      Glass             R 20.03979710
## 4      Metal             R  3.68737021
## 5      Paper             R  8.45764838
## 6   Plastics             R  5.39145002
## 7      Trash             R 39.45790786
## 8  Cardboard             T  8.35691793
## 9    Compost             T  4.92356044
## 10     Glass             T  3.49363016
## 11     Metal             T  0.42348864
## 12     Paper             T  0.99133653
## 13  Plastics             T  1.86514707
## 14     Trash             T 70.69710911
```

Next we'll put change the name of the variables and merge the datasets:[7]

```
names(Sorted.sd)= c("Type", "Trash_Recycle", "sd"); head(Sorted.sd)

##         Type Trash_Recycle          sd
## 1 Cardboard             R 14.14448025
## 2   Compost             R  0.03898635
## 3     Glass             R 15.45730315
## 4     Metal             R  5.83196515
## 5     Paper             R  3.70017992
## 6  Plastics             R  2.02099651

names(Sorted.n)= c("Type", "Trash_Recycle", "n"); head(Sorted.n)

##         Type Trash_Recycle n
## 1 Cardboard             R 4
## 2   Compost             R 4
## 3     Glass             R 4
## 4     Metal             R 4
## 5     Paper             R 4
## 6  Plastics             R 4

Sorted.mean

##          Type Trash_Recycle           x
```

[7] Rarely do you need to do all this, but these data are a bit complicated. The example in the next section will be much eaiser to follow, so you can let your eyes glaze over the details here.

```
## 1  Cardboard             R 33.13571287
## 2    Compost             R  0.01949318
## 3      Glass             R 20.03979710
## 4      Metal             R  3.68737021
## 5      Paper             R  8.45764838
## 6   Plastics             R  5.39145002
## 7      Trash             R 39.45790786
## 8  Cardboard             T  8.35691793
## 9    Compost             T  4.92356044
## 10     Glass             T  3.49363016
## 11     Metal             T  0.42348864
## 12     Paper             T  0.99133653
## 13  Plastics             T  1.86514707
## 14     Trash             T 70.69710911
```

```
Sorted.SEM = merge(Sorted.sd, Sorted.n)
Sorted.Confidence = merge(Sorted.mean, Sorted.SEM)
Sorted.Confidence$SEM = Sorted.Confidence$sd/sqrt(Sorted.Confidence$n)

Sorted.Confidence <- droplevels(Sorted.Confidence)
```

In the next section, we are going to use the t-distribution to estimate our confidence intervals.

Calculating a Parametric Confidence Interval

To explore our waste audit data, we will determine the 95% confidence intervals for the mean for each mean. As usual there are dozens of way to accomplish this, but for now, let's start by getting a t-statistic based on a criteria α value of 0.05. Since we calculate CI for the lower and upper limit, we need to split the probability in half ($\alpha/2$) and determine the intervals for 0.025 and 0.975 of the probably distribution.

We begin by using the t-Distribution, which is a specialized case of the normal distribution (standard normal distribution that is corrected for sample size with change in the degrees of freedom).[8].

$$\bar{x} - t_{\alpha/2,n-1}(sd/\sqrt{n}) < \mu < \bar{x} + t_{\alpha/2,n-1}(sd/\sqrt{n}) \tag{4}$$

```
alpha = 0.05
degfree = 4 - 1
qt(alpha/2, degfree)

## [1] -3.182446
```

[8] A figure here would be useful!

```
par(cex=1.3, cex.axis=1)
plot(xvalues, ylim=Ylim, xlim=c(0.5,7.5),ty="n", xaxt='none',
    xlab="Type", las=1, ylab="% of Unsorted Mass")
axis(side=1, at=1:7, label=levels(Sorted.Confidence$Type),
    cex.axis=.5)
points(xvalues,
    Sorted.Confidence$x[Sorted.Confidence$Trash_Recycle=="R"],
    col="darkgreen", pch=19)
points(x1, Sorted.Confidence$x[Sorted.Confidence$Trash_Recycle=="T"],
    col="Red", pch=2)
with(Sorted.Confidence[Sorted.Confidence$Trash_Recycle=="R",],
     arrows(xvalues, CI.low, xvalues, CI.high, length=0.05,
            angle=90, code=3, lwd=2, col="darkgreen"))
with(Sorted.Confidence[Sorted.Confidence$Trash_Recycle=="T",],
     arrows(x1, CI.low, x1, CI.high, length=0.05,
            angle=90, code=3, lwd=2, col="red"))
# Add a legend
legend(2, 95, legend=c("Recycling", "Trash"),
      col=c("darkgreen", "red"), pch=c(19, 2), cex=.8, bty='n')
```

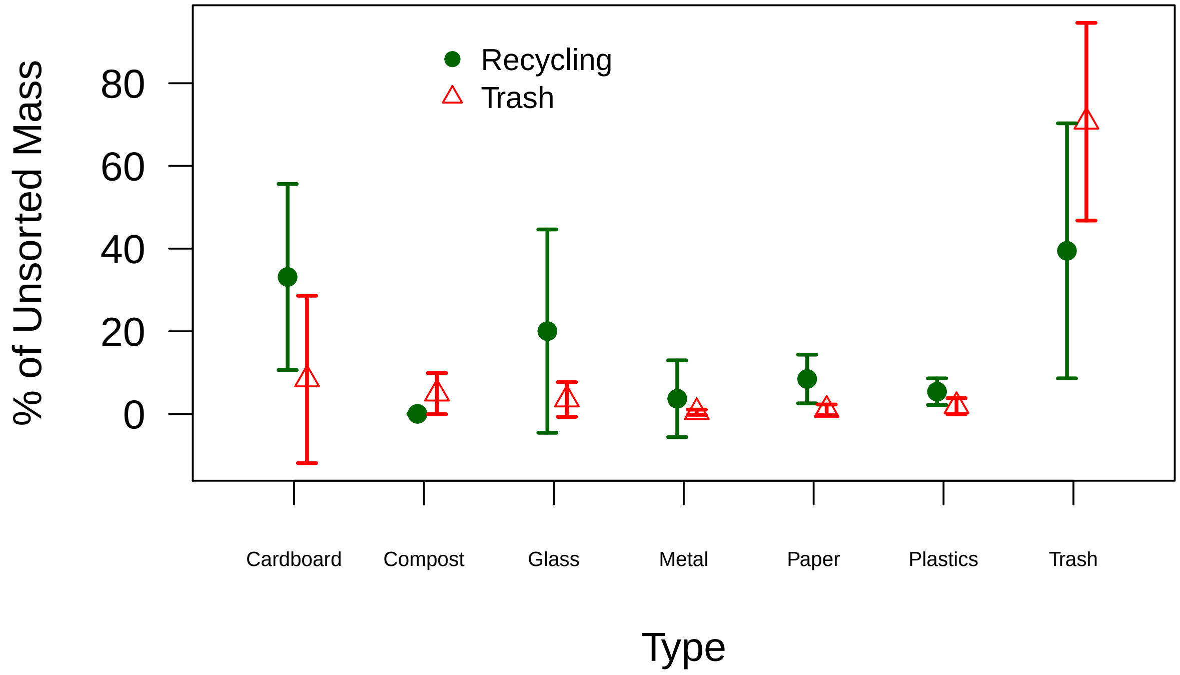


Figure 6: % material from unsorted recycling and trash mass (95% Confidence Intevals).

Implementing Code to Create Confidence Intervals

Steps to Create Confidence Intervals

This example assumes that the samples are drawn from a Normal distribution. The basic procedure for calculating a confidence interval for a population mean is as follows:

1. Identify the sample mean, $\bar{x}$.
2. Identify whether the population standard deviation is known, σ, or is unknown and is estimated by the sample standard deviation s.

   - If the population standard deviation is known then $z^* = \Phi^{-1}\left(1 - \frac{\alpha}{2}\right) = -\Phi^{-1}\left(\frac{\alpha}{2}\right)$, where $C = 100(1-\alpha)$ is the confidence level and Φ is the CDF of the standard normal distribution, used as the critical value. This value is only dependent on the confidence level for the test. Typical two sided confidence levels are:

C	z*
99%	2.576
98%	2.326
95%	1.96
90%	1.645

   - If the population standard deviation is unknown then the Student's t distribution is used as the critical value. This value is dependent on the confidence level (C) for the test and degrees of freedom. The degrees of freedom are found by subtracting one from the number of observations, $n1$. The critical value is found from the t-distribution table. In this table the critical value is written as $t^* = t_\alpha(r)$, where r is the degrees of freedom and $\alpha = \frac{1-C}{2}$.

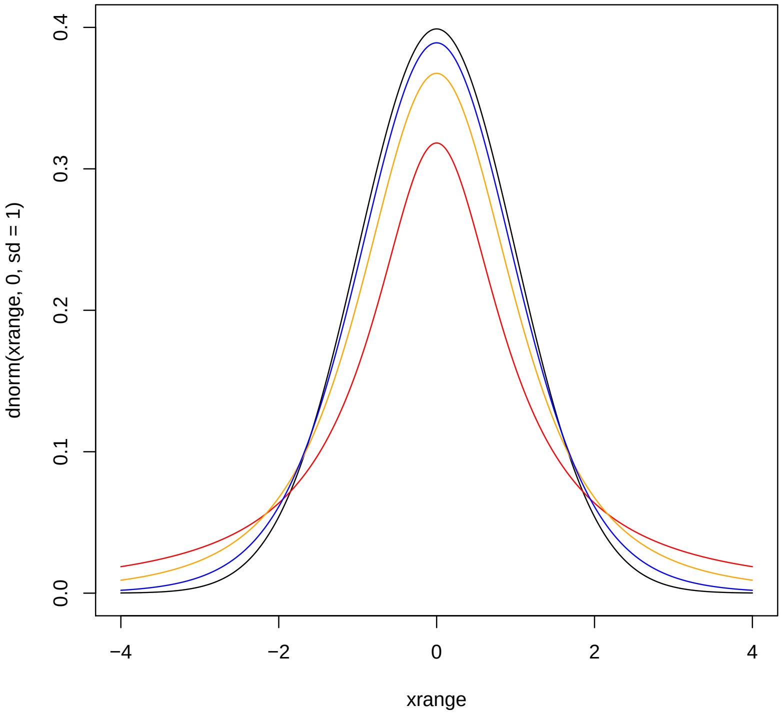
3. Plug the found values into the appropriate equations:

   - For a known standard deviation: $\left(\bar{x} - z^* \frac{\sigma}{\sqrt{n}}, \bar{x} + z^* \frac{\sigma}{\sqrt{n}}\right)$
   - For an unknown standard deviation: $\left(\bar{x} - t^* \frac{s}{\sqrt{n}}, \bar{x} + t^* \frac{s}{\sqrt{n}}\right)$

Normal versus the t-Distribution

For demonstration purposes, we should compare the standard normal distribution to the t-distribibuition so we understand what the implications of distribution might mean in hypothesis testing!

```r
xrange = seq(-4, 4, by=.02)
plot(xrange, dnorm(xrange, 0, sd=1), ty='l', ylim=c(0,.4), xlim=c(-4,4))

lines(xrange, dt(xrange, df=1), ty='l', col='red')
lines(xrange, dt(xrange, df=3), ty='l', col='orange')
lines(xrange, dt(xrange, df=10), ty='l', col='blue')
```



Notice that the t-distribution is squatter in the middle and fatter at the edges. This means there is more probability "area" at the ends. But as you approach the n=20, the curves are increasingly overlapping.

How to make these plots

Notice that I used `dnorm()` and `dt()` functions to create the figures. What are these? These functions create probability density curves. It's cool to see that that there are several variants of these functions that are related.

`rnorm()` generates n random numbers from a normal distribution with a defined mean and sd (standard deviation).

`dnorm()` generates probability densities based on the quantile, mean, and sd. This can be based on an upper or lower side of the distribution. More on this below.

`qnorm()` generates quantiles based on the probabilities, mean and sd. This can also be based on the upper or lower side of the distribution.

`pnorm()` generates probabilities based on the quantiles, mean and sd.

To test demonstrate these functions, let's try each out. First, I'll create a set of twenty random numbers (let's use the standard normal, which as a mean of zero and a sd of one) and round them to the nearest 100th.

```
(randomnumbers = round(rnorm(20, mean=0, sd=1), 2))

##  [1]  0.16  0.88  0.04 -0.91 -1.65 -1.29 -0.29  0.54  1.81
## [10] -0.99 -0.52  0.93 -0.25 -1.15  0.54  2.70  2.33 -0.16
## [19] -0.13 -0.19

#hist(randomnumbers)
```

Now, let's uses these figure out what the probabilies, using the default for lower tail, i.e. `lower.tail=T`. In other words, what is the probability that the number will be higher than the lower side of the tail.

```
round(pnorm(randomnumbers, mean=0, sd=1, lower.tail=T), 2)

##  [1] 0.56 0.81 0.52 0.18 0.05 0.10 0.39 0.71 0.96 0.16 0.30
## [12] 0.82 0.40 0.13 0.71 1.00 0.99 0.44 0.45 0.42
```

finally, let's see what the qauntiles look like based on various probabilities

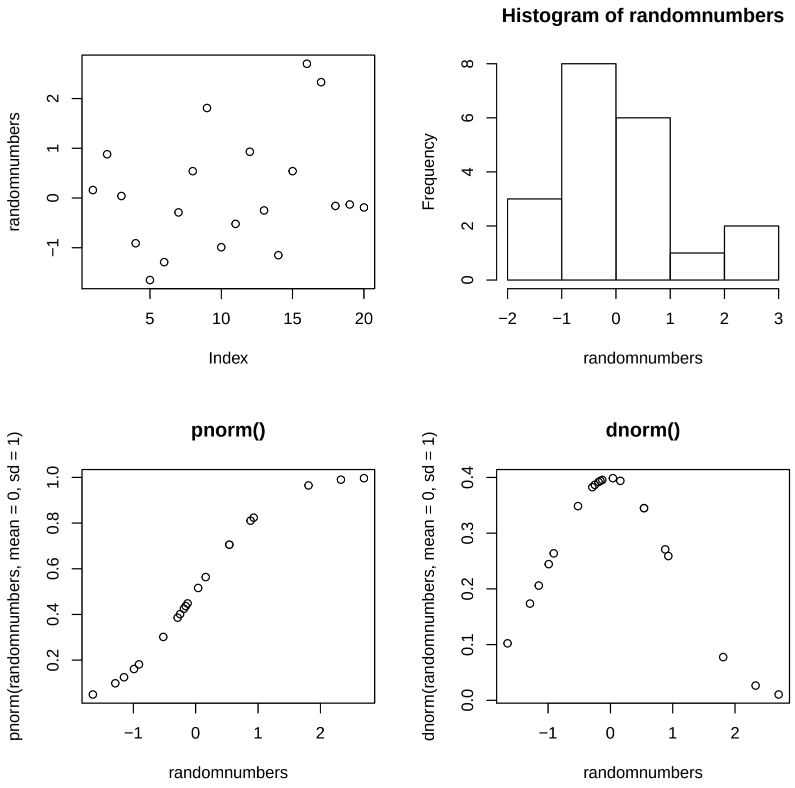
```
round(qnorm(c(0.01, 0.025, .05, .1, .5, .9, .95, .975, 0.99),
            mean=0, sd=1, lower.tail=T), 2)

## [1] -2.33 -1.96 -1.64 -1.28  0.00  1.28  1.64  1.96  2.33
```

If we switch these numbers to the 68-95-99.7 rule probabilities. First, we need to divide the tails in 1/2 and then we can put them in the `qnorm()` function. We can see where the the 1, 2 and 3 standard deviations came from (rounded right?). Pretty Neat!

```
percent.68 = (1 - 0.68)/2
percent.95 <- (1 - 0.95)/2
percent.99 <- (1 - 0.997)/2
round(-qnorm(c(percent.68, percent.95, percent.99), mean=0, sd=1), 2)

## [1] 0.99 1.96 2.97
```

Okay, let's plot these functions and see how they work!

```
par(mfrow=c(2,2))
plot(randomnumbers); hist(randomnumbers)
plot(randomnumbers, pnorm(randomnumbers, mean=0, sd=1), main="pnorm()")
plot(randomnumbers, dnorm(randomnumbers, mean=0, sd=1), main="dnorm()")
```



Calculating Confidence Intervals with known σ^2.

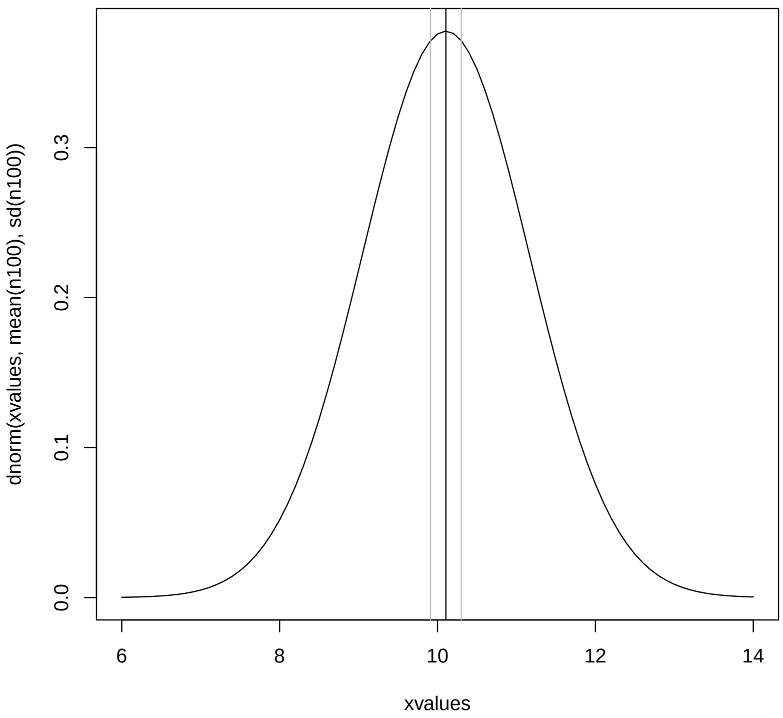
Because we have the values for the entire population, we can estimate confidence intervals using the normal distribution (z* scores) using the formula above.

We'll create confidence intervals using our n=100 sample.

The $\bar{x}$ = 10.11 and σ = 0.993148. First, we'll create a function to calculate our confidence intervals that has the following inputs: sample vector, level of confidence (α), and μ.

```
cl<- function(sample, alpha, sigma){
  sem <- sigma/sqrt(length(sample))
  ucl <- mean(sample) + qnorm(1-alpha/2)*sem
  lcl <- mean(sample) - qnorm(1-alpha/2)*sem
  ci <- c(lcl, ucl)
  return(ci)
}

(cl95.n100 <- cl(n100, 0.05, sd(N1000)))

## [1]  9.911347 10.300653
```

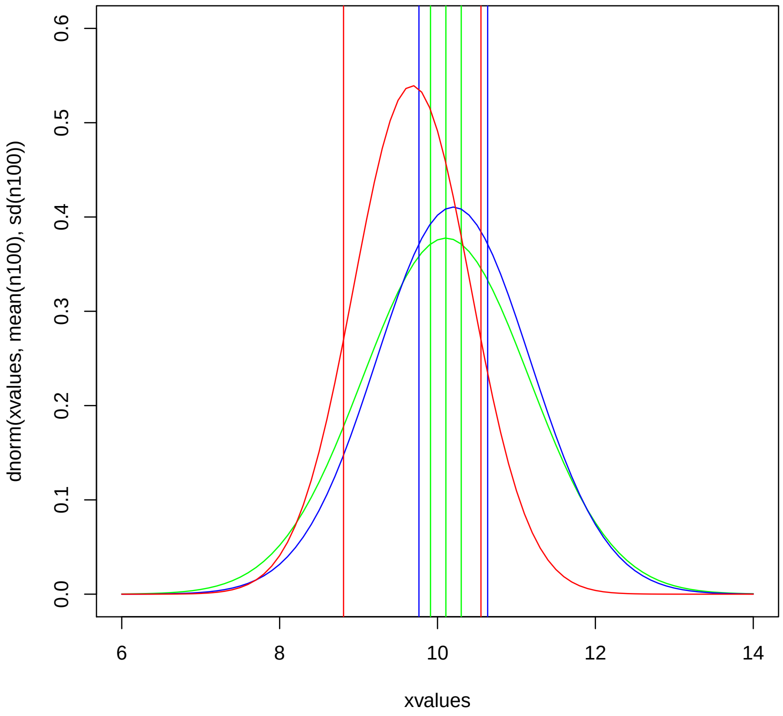
```
xvalues = seq(6,14,.1)
plot(xvalues, dnorm(xvalues, mean(n100), sd(n100)), ty='l')
abline(v=mean(n100))
abline(v=cl95.n100[1], col='grey')
abline(v=cl95.n100[2], col='grey')
```



How does n influence confidence intervals?

```
(cl95.n20 <- cl(n20, 0.05, sd(N1000)))

## [1]  9.764742 10.635258

(cl95.n20 <- cl(n20, 0.05, sd(N1000)))

## [1]  9.764742 10.635258

(cl95.n5 <- cl(n5, 0.05, sd(N1000)))

## [1]  8.809483 10.550517

xvalues = seq(6,14,.1)
plot(xvalues, dnorm(xvalues, mean(n100), sd(n100)), ty='l', col='green', ylim=c(0, 0.6))
```

```
abline(v=mean(n100), col='green')
abline(v=cl95.n100[1], col='green')
abline(v=cl95.n100[2], col='green')

lines(xvalues, dnorm(xvalues, mean(n20), sd(n20)), ty='l', col='blue')
abline(v=cl95.n20[1], col='blue')
abline(v=cl95.n20[2], col='blue')

lines(xvalues, dnorm(xvalues, mean(n5), sd(n5)), ty='l', col='red')
abline(v=cl95.n5[1], col='red')
abline(v=cl95.n5[2], col='red')
```



Confidence Intervals with an unknown σ

```
qt((1-.95)/2, 4)*sd(n5)/sqrt(5)+mean(n5)

## [1] 8.761672
```

@

Misunderstandings about Confidence Intervals

Confidence intervals and levels are frequently misunderstood, and published studies have shown that even professional scientists often misinterpret them. I find myself tripped up the the vocabularly regularly.

A 95% confidence level does not mean that for a given realized interval there is a 95% probability that the population parameter lies within the interval (i.e., a 95% probability that the interval covers the population parameter). According to the strict frequentist interpretation, once an interval is calculated, this interval either covers the parameter value or it does not; it is no longer a matter of probability.

The 95% probability relates to **the reliability of the estimation procedure**, not to a specific calculated interval.

Neyman himself (the original proponent of confidence intervals) made this point in his original paper:

> It will be noticed that in the above description, the probability statements refer to the problems of estimation with which the statistician will be concerned in the future. In fact, I have repeatedly stated that the frequency of correct results will tend to α. Consider now the case when a sample is already drawn, and the calculations have given [particular limits]. Can we say that in this particular case the probability of the true value [falling between these limits] is equal to α? The answer is obviously in the negative. The parameter is an unknown constant, and no probability statement concerning its value may be made . . .

Deborah Mayo expands on this further as follows:

> It must be stressed, however, that having seen the value [of the data], Neyman-Pearson theory never permits one to conclude that the specific confidence interval formed covers the true value of 0 with either $(1-\alpha)100\%$ probability or $(1-\alpha)100\%$ degree of confidence. Seidenfeld's remark seems rooted in a (not uncommon) desire for Neyman-Pearson confidence intervals to provide something which they cannot legitimately provide; namely, a measure of the degree of probability, belief, or support that an unknown parameter value lies in a specific interval. Following Savage (1962), the probability that a parameter lies in a specific interval may be referred to as a measure of final precision. While a measure of final precision may seem desirable, and while confidence levels are often (wrongly) interpreted as providing such a measure, no such interpretation is warranted. Admittedly, such a misinterpretation is encouraged by the word 'confidence.'

- A 95% confidence level does not mean that 95% of the sample data lie within the confidence interval.

- A confidence interval is not a definitive range of plausible values for the sample parameter, though it may be understood as an estimate of plausible values for the population parameter.

- A particular confidence level of 95% calculated from an experiment does not mean that there is a 95% probability of a sample parameter from a repeat of the experiment falling within this interval.

What what the heck is a confidence interval?

Various interpretations of a confidence interval can be given (taking the 90% confidence interval as an example in the following).

- The confidence interval can be expressed in terms of samples (or repeated samples): "Were this procedure to be repeated on numerous samples, the fraction of calculated confidence intervals (which would differ for each sample) that encompass the true population parameter would tend toward 90%."
- The confidence interval can be expressed in terms of a single sample: "There is a 90% probability that the calculated confidence interval from some future experiment encompasses the true value of the population parameter." Note this is a probability statement about the confidence interval, not the population parameter. This considers the probability associated with a confidence interval from a pre-experiment point of view, in the same context in which arguments for the random allocation of treatments to study items are made. Here the experimenter sets out the way in which they intend to calculate a confidence interval and to know, before they do the actual experiment, that the interval they will end up calculating has a particular chance of covering the true but unknown value. This is very similar to the "repeated sample" interpretation above, except that it avoids relying on considering hypothetical repeats of a sampling procedure that may not be repeatable in any meaningful sense.
- The explanation of a confidence interval can amount to something like: "The confidence interval represents values for the population parameter for which the difference between the parameter and the observed estimate is not statistically significant at the 10% level." In fact, this relates to one particular way in which a confidence interval may be constructed.

In each of the above, the following applies: If the true value of the parameter lies outside the 90% confidence interval, then a sampling event has occurred which had a probability of 10% (or less) of happening by chance.

t-test

What is a (statisical) hypothesis?

A hypothesis is a prediction — and using the frequentists approach, we make a negative prediction, i.e. that there is no pattern, to test. In our example with the waste, we can test was the the amount recycling stastically significant compared to the total waste. In other words, were students diverting waste. We can test if the confidence intervals include 100%.

First, we'll subset the data!

```
trash = subset(Sorted, Type == "Trash",
        select = c(Trash_Recycle, Type, Percent)); trash

##    Trash_Recycle  Type  Percent
## 8              R Trash 24.95127
## 16             R Trash 33.24468
## 24             R Trash 31.60622
## 32             R Trash 68.02947
## 40             T Trash 48.70641
## 48             T Trash 73.69888
## 56             T Trash 78.74251
## 64             T Trash 81.64062
```

```
t.test(trash[1:4,3], mu=100)

##
##  One Sample t-test
##
## data:  trash[1:4, 3]
## t = -6.2471, df = 3, p-value = 0.008275
## alternative hypothesis: true mean is not equal to 100
## 95 percent confidence interval:
##   8.616218 70.299597
## sample estimates:
## mean of x
##  39.45791
```

The t.test output is a bit confusioning. But let's start with the null hypothesis, which is the mean is equal to 100. The alternative is "true mean is not equal to 100." Next, let's check out the p-value: 0.0083.[9]

Thus, we can reject the null hypothesis, the mean is equal to 100%. Thus, we are testing if there is a difference between the total unsorted weight of the recyling and the weight of the trash in the recycling

[9] What should you report? We don't need all these decimals! Thus, we usually report p values as < 0.05, <0.01, and <0.001. It's a good practices and sufficiently informative for most purposes.

bins. We hope there is a difference! And there is because we can reject the null hypothesis.

Thus, evidence students are diverting not just dumping stuff randomly in the recycling bins. I guess that is good news!

A second question is what there a signficant amount of recyclable material that should have been recycled? What the amount of potentially diverted material statistically significant?

```
t.test(trash[5:8,3], mu=100)

##
##  One Sample t-test
##
## data:  trash[5:8, 3]
## t = -3.901, df = 3, p-value = 0.0299
## alternative hypothesis: true mean is not equal to 100
## 95 percent confidence interval:
##  46.79183 94.60239
## sample estimates:
## mean of x
##  70.69711
```

This is not surprising, but disappointing. The amount of recycling in the trash is significantly below 100%, that suggests recyclelable materials are being put into the trash in a signficant fashion. We might notice that one observation # is a big problem. What is #40?

```
Sorted[35,] # Not sure why this isn't #40, still sorting that out... !!!

##    Sample.Date Trash_Recycle Location  Type Mass_Container
## 40   9/23/2019             T Dialynas Trash           43.3
##    Container NetMass Total  Percent
## 40         0    43.3  88.9 48.70641
```

No surprise – the parties is Dialynas lead to terrible recycling behavior. However, with the new rules from WMI, perhaps they were anticipating the change and are actually better performers than we think!

Comparing Means

We can then explore if the % trash in recycling was different than the % trash in the recycling. We test this with a null hypothesis, that there is no difference between the percentage.

```
trash.RT = subset(Sorted, Type == "Trash",
            select = c(Trash_Recycle, Percent)); trash
```

```
##    Trash_Recycle  Type  Percent
## 8              R Trash 24.95127
## 16             R Trash 33.24468
## 24             R Trash 31.60622
## 32             R Trash 68.02947
## 40             T Trash 48.70641
## 48             T Trash 73.69888
## 56             T Trash 78.74251
## 64             T Trash 81.64062
```

```
t.test(trash.RT[1:4,2], trash.RT[5:8,2])
```

```
##
##  Welch Two Sample t-test
##
## data:  trash.RT[1:4, 2] and trash.RT[5:8, 2]
## t = -2.5478, df = 5.6487, p-value = 0.04602
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -61.7001179  -0.7782846
## sample estimates:
## mean of x mean of y
##  39.45791  70.69711
```