

# Confidence Intervals

Marc Los Huertos

October 8, 2019

One of the most important considerations with univariate data are the development of confidence intervals. As you can probably guess, confidence intervals are based on the distribution of the data. There are two main approaches in developing confidence intervals. One relies on the use of theoretical probability distributions but there are a number of methods that do not rely on theoretical distributions. We will learn several methods because each are used in environmental sciences to varying degrees.

## 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 or confidence coefficient. Increasing the desired confidence level will widen the confidence interval. These are key in presenting quantitative data because they allow us to interpret the data from a hypothesis 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 mean at this point, but keep in mind the concept can be applied to many parameter estimates.

## Working an Example

### 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 developed semi-consistent symbology for various statistics based on populations and samples.

A population mean for example is usually referred to by the Greek letter,  $\mu$ . While for a sample, it might be referred to as  $\bar{x}$ . The spread of data, or variance, in a population is referred to as  $\sigma^2$ , again a Greek letter. The population variance is often referred to as  $s^2$ . In



Figure 1: Confidence abounds without bounds.

general, we don't know  $\mu$  or  $\sigma^2$ , so we can estimate it and develop confidence intervals that probably include the true  $\mu$ . Notice the word, "probably." In other words, we our intervals in terms of probabilities!

### *The difference between $\mu$ and $\bar{x}$*

To illustrate the difference, let's create a dataset of random numbers from a normal distribution with a mean of 1 and standard deviation of 1.

```
set.seed(123)
N1000 = rnorm(1000, 1, 1)
```

These data will represent the population (N) and has a mean or  $\mu$  of 1.0161. The values range from -1.81 to 4.24 (Figure 2). From this dataset, we sample 10 numbers randomly.

```
n10 = sample(N1000, 10)
```

We get the following sample from our population:

```
n10
## [1] 0.6254191 -0.6015362 3.1001089 1.8824652 0.8942158
## [6] 0.9839975 0.3548860 2.5164706 2.7150650 -0.2847157
```

and a mean or  $\bar{y} = 1.219$ . These values certainly fall within the range of values in the population.

### *R's Default Boxplot*

Let's see if we can label the graphic directly!

### *Measuring the Spread*

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

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

(1.637) and the standard deviation (1.28), which is the square root of  $s^2$ . Then, we can calculate the standard error of the sample using the following equation:

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

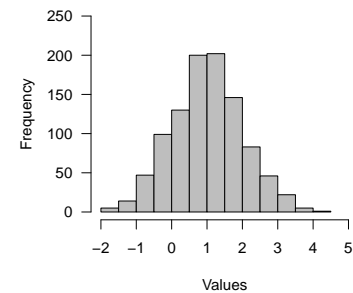
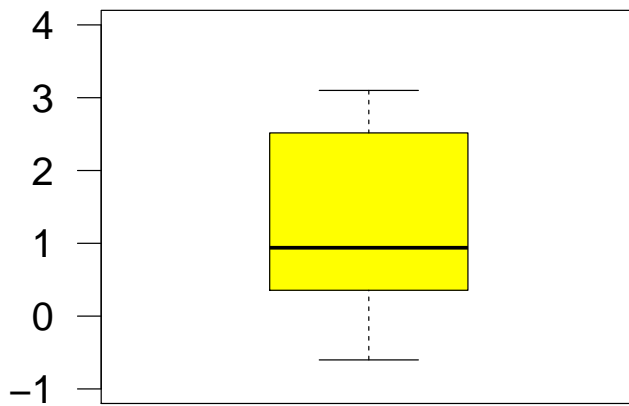


Figure 2: Frequency distribution of population.

Figure 3: Default Boxplot where the following are shown: median = 0.94; Range = -0.6, 3.1; and interquartile range = 0.42, 2.36.

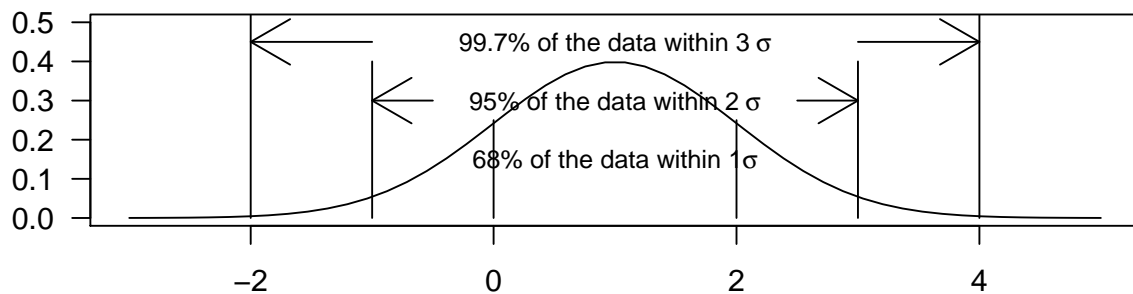


```
#computation of the standard error of the sample mean
sem<-sd(n10)/sqrt(length(n10))
```

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

### *Selecting $\alpha$ : The Level of Confidence*

First, it is 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. 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.



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 covers 95% of the normal curve – the probability of observing a value outside of this area is less than 0.05. So, following standards in statistics, we use  $\alpha$  to signify the criteria, such that

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

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  $\alpha$  for each side of curve. Thus for a for a 95% confidence interval, the area in each tail is equal to  $0.05/2 = 0.025$ .

### *Estimating Confidence Intervals for Waste Audit*

```
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 calculate 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.

### *Exploring the Data*

Anyone who has worked with quantitative 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 easily see that there is a problem with the Trash sources. We simple are not getting all the trash out of the bags and weighed. This is something that we'll need to sort out how the problem was created, how it might be remedied, or if the work needs to be done over to improve the quaulity of the results.

### *Calculating Summary Statistics*

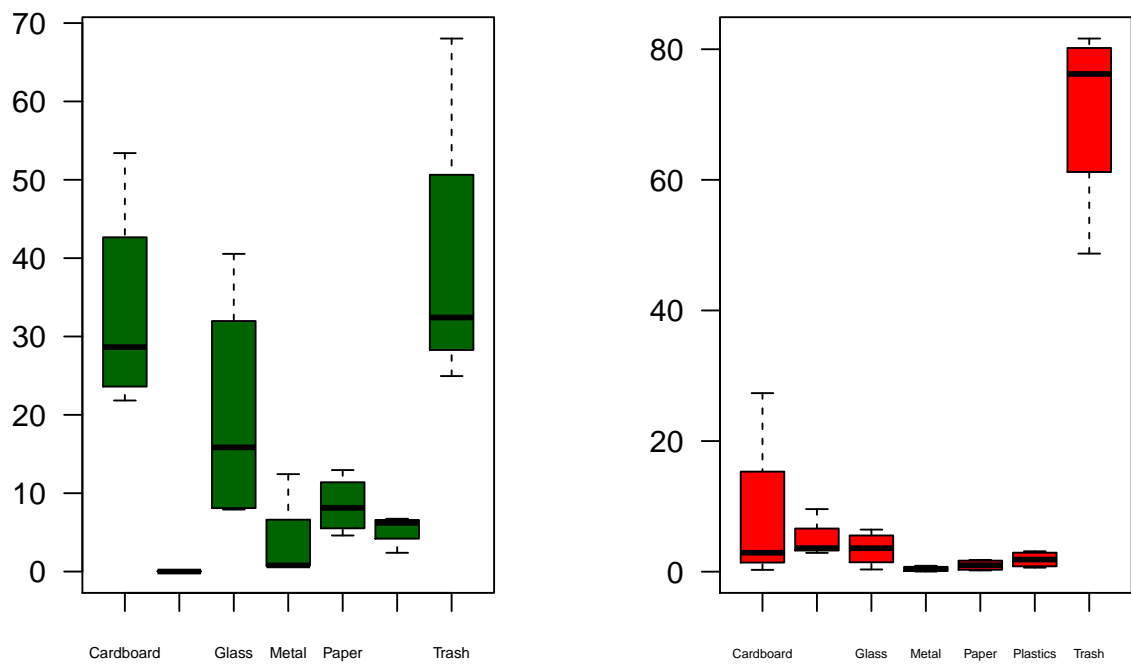


Figure 4: Sorted Percent Mass from Recycling Bags

```
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)
```

```
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
## 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)
```

Now, we are going to use the t-distribution instead of our estimates so we can create exact probabilities.

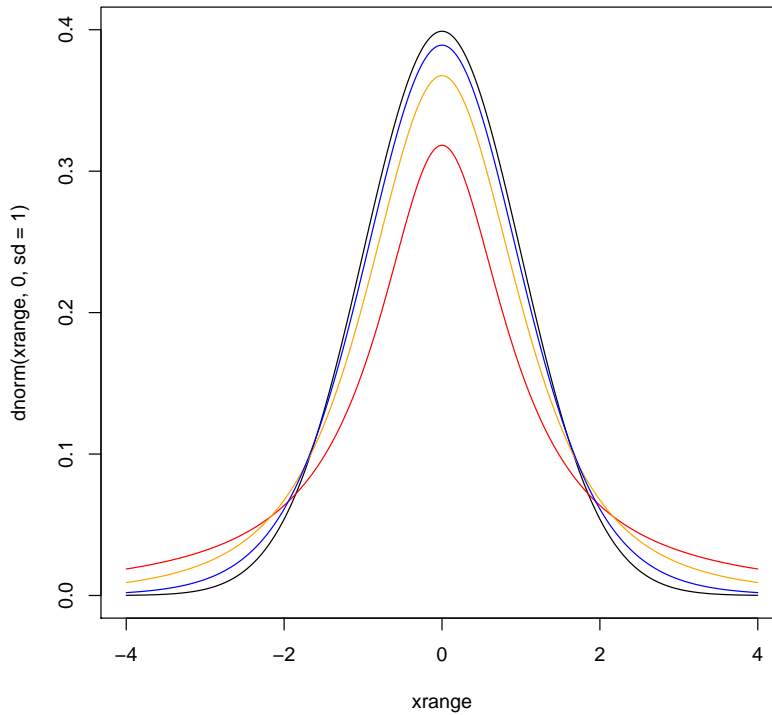
### *Normal versus the t-Distribution*

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

```
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.

Let's create a legend for this...

### *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 by our is that we use our  $\alpha$  value of 0.05. Since we calculate CI for the lower and upper limit, we need to split the probability in half 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).

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

```
alpha = 0.05  
degfree = 4 - 1  
qt(alpha/2, degfree)  
  
## [1] -3.182446
```

```

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')

```

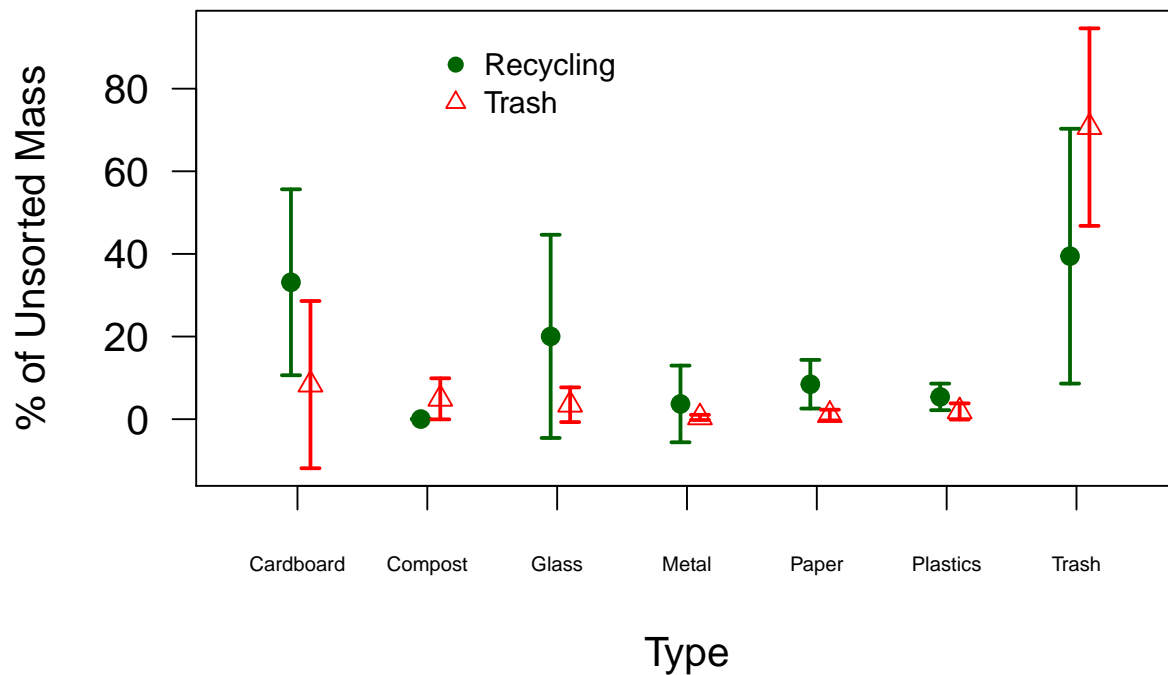


Figure 5: % material from unsorted recycling and trash mass (95% Confidence Intervals).

*t-test**What is a (statistical) 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.<sup>1</sup>

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 recycling and the weight of the trash in the recycling

<sup>1</sup> 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 significant 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 recyclable materials are being put into the trash in a significant 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

```