Confidence Intervals

Marc Los Huertos

October 7, 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.

Figure 1: Confidence abounds without bounds.

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 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, it might be referred to as $\bar{x}$. 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 it and develop confidence intervals that probably include the true μ. Notice the word, "probably." In other words, we our intervals in terms of probabilities!

The difference between μ 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)
```

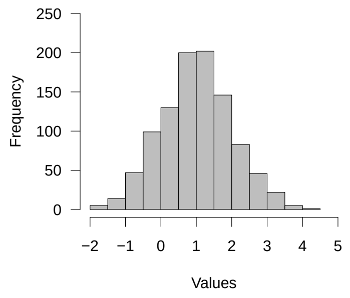


Figure 2: Frequency distribution of population.

These data will represent the population (N) and has a mean or μ 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

Measuring the Spread

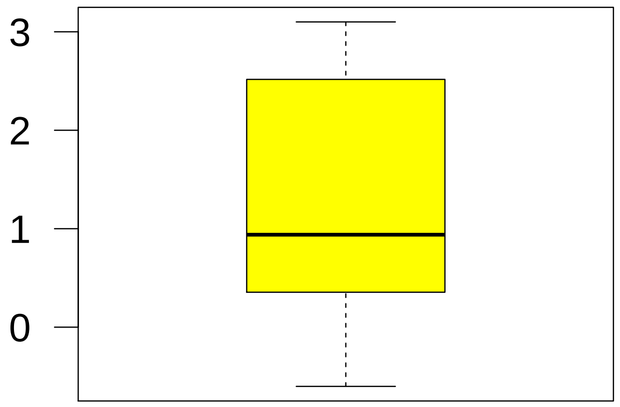
We can calculate the variance of the sample using the following formula:

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

(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:

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

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

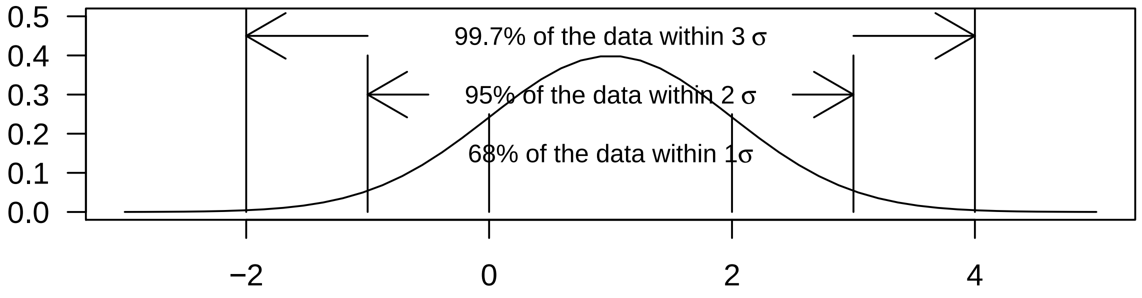


```
#computation of the standard error of the sample mean
sem<-sd(n10)/sqrt(length(n10))
```

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

Selecting α: 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 α to signify the as the criteria, 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 $0.05/2 = 0.025$.

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 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"
```

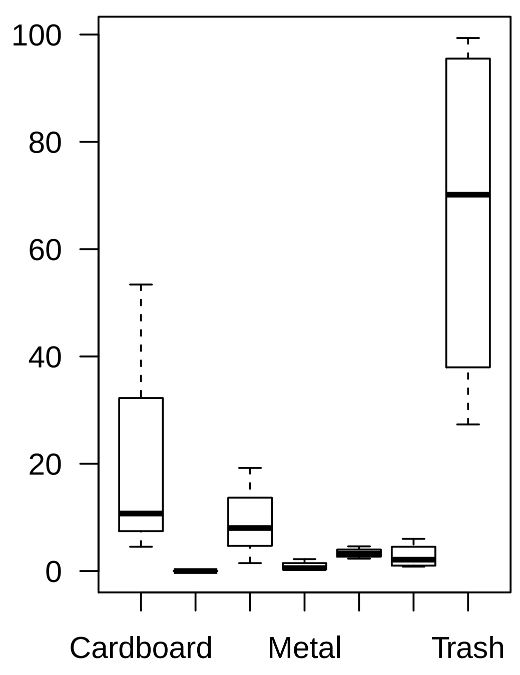
Be sure to check the results as you go and convince yourself how these work by looking online to see how these functions work.

Exploring the Data

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 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 Statitics



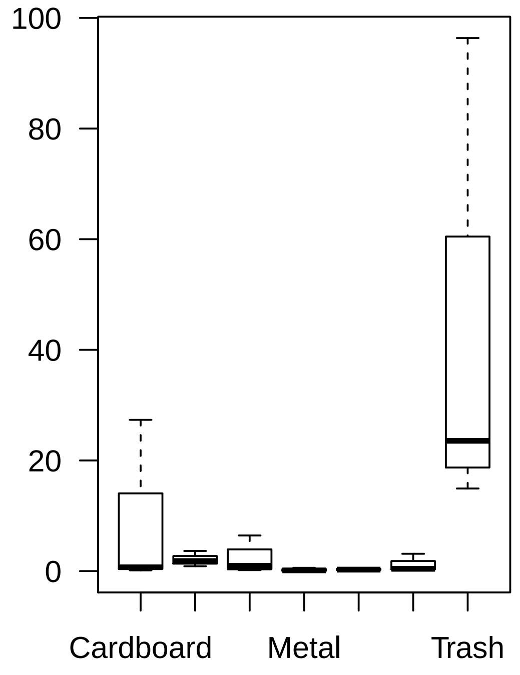


Figure 4: Sorted Percent Mass from Recycling Bags

```
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 22.56671944
## 2   Compost             R  0.01848429
## 3     Glass             R  7.36818369
## 4     Metal             R  0.90058168
## 5     Paper             R  0.95707298
## 6  Plastics             R  2.36304529

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 19.844355824
## 2    Compost             R  0.009242144
## 3      Glass             R  9.188614632
## 4      Metal             R  0.900428236
## 5      Paper             R  3.346022133
## 6   Plastics             R  2.771715635
## 7      Trash             R 66.744612153
## 8  Cardboard             T  7.219242440
## 9    Compost             T  2.036823227
## 10     Glass             T  2.129499618
## 11     Metal             T  0.214627476
## 12     Paper             T  0.304364752
## 13  Plastics             T  1.065179585
## 14     Trash             T 39.601450481

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.

Calculating a Parametric Confidence Interval

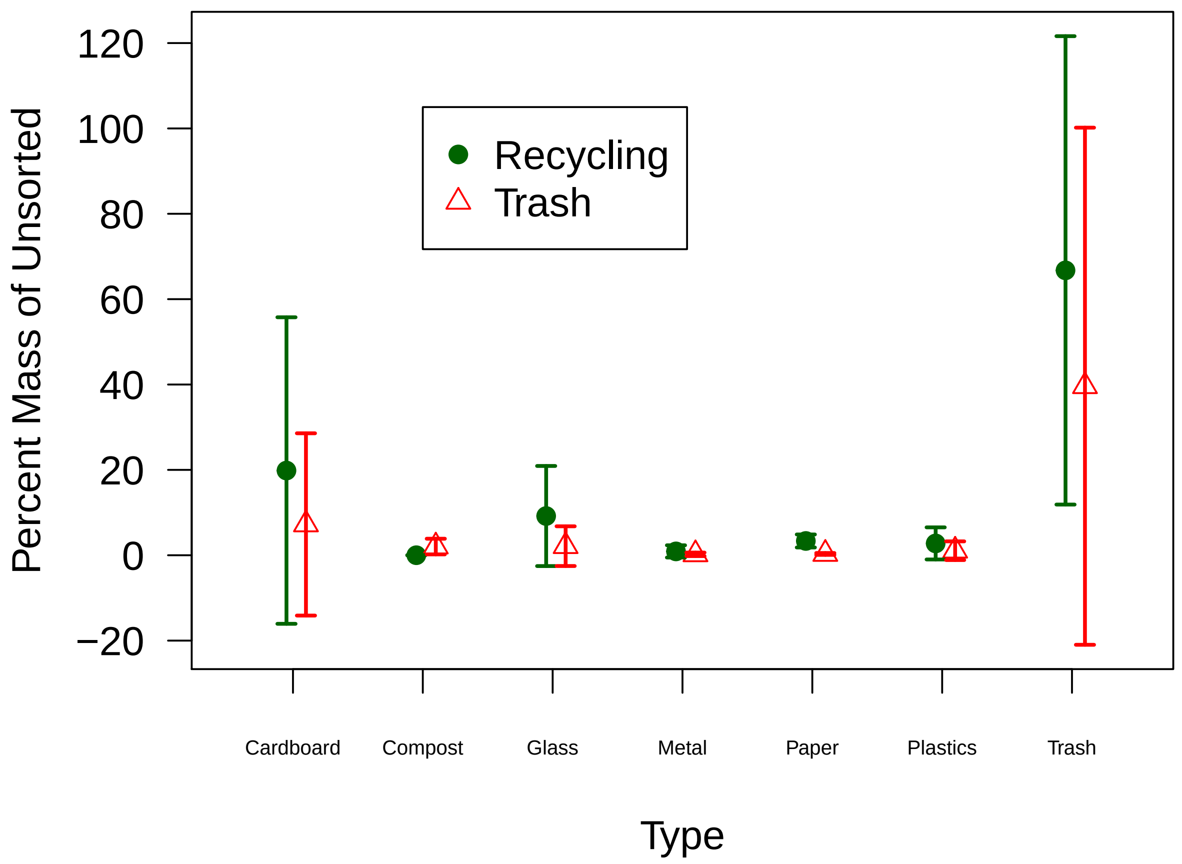
To explore our waste audit data, we will determine the 95% confidence intervals for the mean for each mean. As usual there are dozens of way to accomplish this, but for now, let's start by getting a t-statistic by our is that we use our α value of 0.05. Since we calculate CI for the lower and upper limit, 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).

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

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

## [1] -3.182446
```

```r
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="Percent Mass of Unsorted")
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, 105, legend=c("Recycling", "Trash"),
      col=c("darkgreen", "red"), pch=c(19, 2), cex=1)
```


Percent Mass of Unsorted
● Recycling
△ Trash
Cardboard
Compost
Glass
Metal
Paper
Plastics
Trash
Type