Introduction to Statistics for Researchers

Derek Sonderegger

January 8, 2016

These notes were originally written for an introductory statistics course for grad students in biological sciences.

The problem with most introductory statistics courses is that they don't prepare the student for the use of advanced statistics. Rote hand calculation is easy to test, easy to grade, and easy for students to learn to do, but is useless for actually understanding how to apply statistics. Because students pursuing a Ph.D. will likely be using statistics for the rest of their professional careers, I feel that this sort of course should attempt to steer away from a "cookbook" undergraduate pedagogy, and give the student enough theoretical background to continue their statistical studies at a high level while staying away from the painful mathematical details that statisticians must work through.

Recent pedagogical changes have been made at the undergraduate level to introduce sampling distributions via permutation and bootstrap procedures. Because those are extremely useful tools in their own right and because of the ability to think about statistical inference from the very start of the course is invaluable, I've attempted to duplicate this approach. I am grateful to the ICOTS 9 organizers and presenters for their expertise, perspective, and motivation for making such a large shift in my teaching.

Statistical software has progressed by leaps and bounds over the last decades. Scientists need access to reliable software that is flexible enough to handle new problems, with minimal headaches. R has become a widely used, and extremely robust Open Source platform for statistical computing and most new methodologies will appear in R before being incorporated into commercial software. Second, data exploration is the first step of any analysis and a user friendly yet powerful mechanism for graphing is a critical component in a researchers toolbox. R succeeds in this area as R has the most flexible graphing library of any statistical software I know of and the basic plots can created quickly and easily. The only downside is that there is a substantial learning curve to learning a scripting language, particularly for students without any programming background. The R package `mosaic` attempts to overcome this difficulty by providing a minimal set of tools for doing introductory statistics that all follow the same syntactical formula. I've made every attempt to use this package to minimize the amount of R necessary.

Because the mathematical and statistical background of typical students varies widely, the course seems to have a split-personality disorder. We wish to talk about using calculus to maximize the likelihood function and define the expectation of a continuous random variable, but also must spend time defining how to calculate the a mean. I attempt to address both audiences, but recognize that it is not ideal.

As these notes are in a continual state of being re-written, I endeavor to keep the latest version available on the GitHub repository for this book at http://oak.ucc.nau.edu/dls354/Home/. In general, I recommend printing the chapter we are currently covering in class.

I encourage instructors to use these notes for their own classes and appreciate notification of the use to encourage me to keep tweaking the content and presentation. Finally, I hope these notes useful to a broad range of students.

Derek Sonderegger, Ph.D.
Department of Mathematics and Statistics
Northern Arizona University

Contents

Chapter 1

Summary Statistics and Graphing

When confronted with a large amount of data, we seek to summarize the data into statistics that somehow capture the essence of the data with as few numbers as possible. Graphing the data has a similar goal... to reduce the data to an image that represents all the key aspects of the raw data. In short, we seek to simplify the data in order to understand the trends while not obscuring important structure.

For this chapter, we will consider data from a the 2005 Cherry Blossom 10 mile run that occurs in Washington DC. This data set has 8636 observations that includes the runners `state` of residence, official `time` (gun to finish, in seconds), `net` time (start line to finish, in seconds), `age`, and `gender` of the runners.

```
library(mosaicData) # library of datasets we'll use
library(ggplot2)    # graphing functions
#library(dplyr)      # data summary tools
head(TenMileRace)   # examine the first few rows of the data

##   state time  net age sex
## 1    VA 6060 5978  12   M
## 2    MD 4515 4457  13   M
## 3    VA 5026 4928  13   M
## 4    MD 4229 4229  14   M
## 5    MD 5293 5076  14   M
## 6    VA 6234 5968  14   M
```

In general, I often need to make a distinction between two types of data.

- Discrete (also called Categorical) data is data that can only take a small set of particular values. For example a college student's grade can be either A, B, C, D, or F. A person's sex can be only Male or Female.[1] Discrete data could also be numeric, for example a bird could lay 1, 2, 3, ... eggs in a breeding season.

- Continuous data is data that can take on an infinite number of numerical values. For example a person's height could be 68 inches, 68.2 inches, 68.23212 inches.

To decided if a data attribute is discrete or continuous, I often as "Does a fraction of a value make sense?" If so, then the data is continuous.

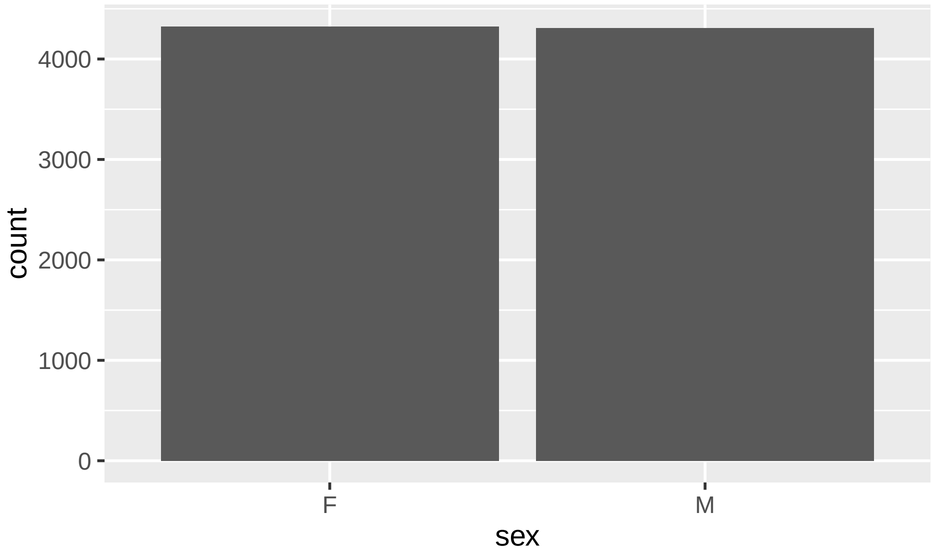
[1] Actually this isn't true as both gender and sex are far more complex. However from a statistical point of view it is often useful to simplify our model of the world. George Box famously said, "All models are wrong, but some are useful."

1.1 Graphical summaries of data

1.1.1 Univariate - Categorical

If we have univariate data about a number of groups, often the best way to display it is using barplots. They have the advantage over pie-charts that groups are easily compared. [2]

```
ggplot(TenMileRace, aes(x=sex)) + geom_bar()
```

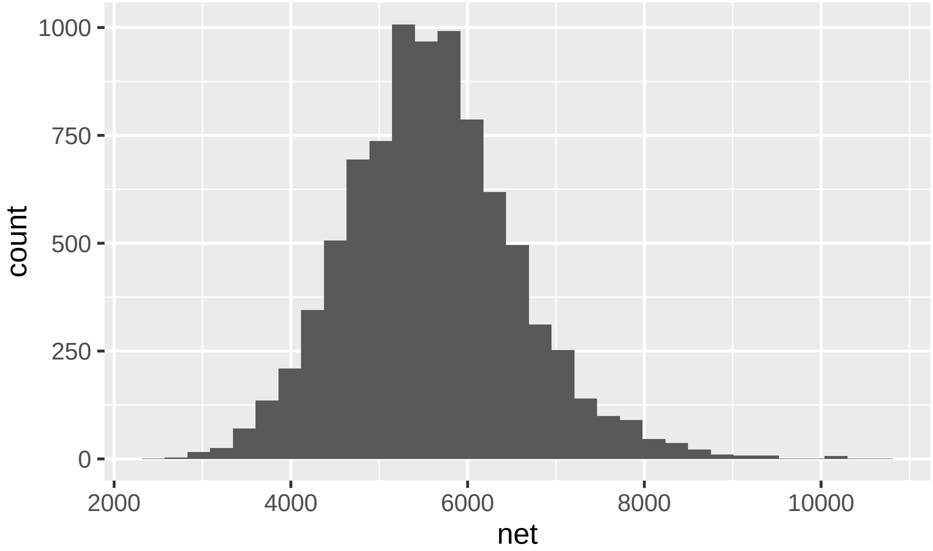


1.1.2 Univariate - Continuous

A histogram looks very similar to a bar plot, but is used to represent continuous data instead of categorical and therefore the bars will actually be touching.

[2] This is an example of a poorly labeled covariate, this really ought to be gender.

```
ggplot(TenMileRace, aes(x=net)) + geom_histogram()

## ‘stat_bin()‘ using ‘bins = 30‘.  Pick better value with ‘binwidth‘.
```



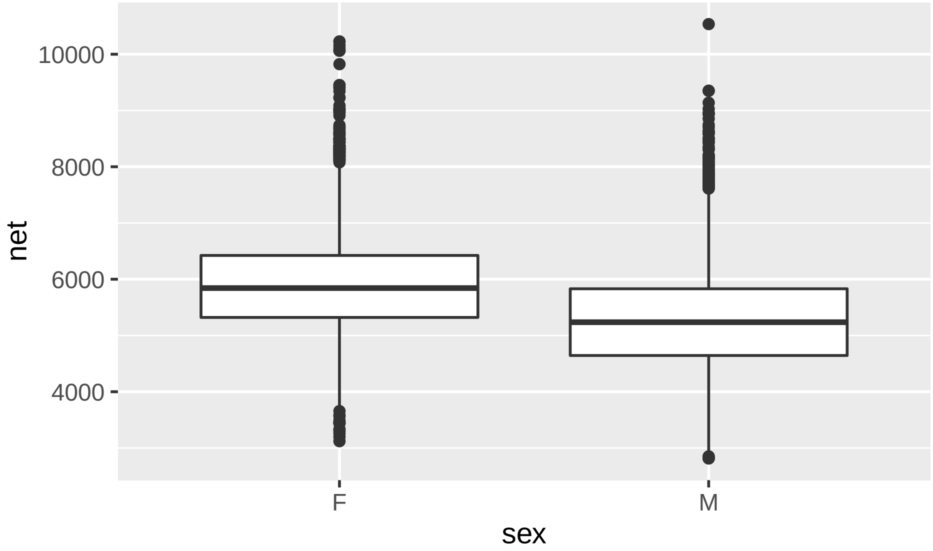
Often when a histogram is presented, the y-axis is labeled as “frequency” or “count” which is the number of observations that fall within a particular bin. However, it is often desirable to scale the y-axis so that if we were to sum up the area (height * width) then the total area would sum to 1. The rescaling that accomplishes this is

$$density = \frac{\#\ observations\ in\ bin}{total\ number\ observations} \cdot \frac{1}{bin\ width}$$

1.1.3 Bivariate - Categorical vs Continuous

We often wish to compare response levels from two or more groups of interest. To do this, we often use side-by-side boxplots. Notice that each observation is associated with a continuous response value and a categorical value.

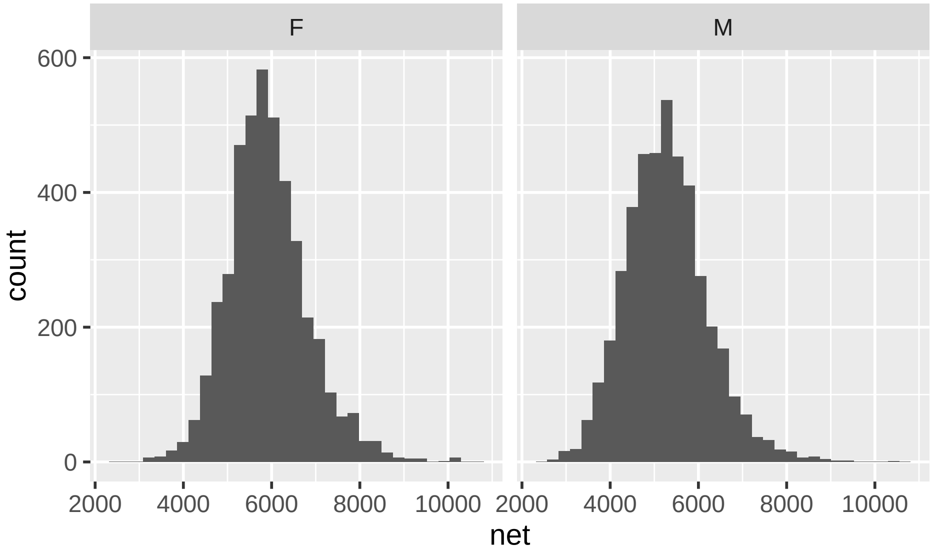
```
ggplot(TenMileRace, aes(x=sex, y=net)) + geom_boxplot()
```



In this graph, the edges of the box are defined by the 25% and 75% quantiles. That is to say, 25% of the data is to the below of the box, 50% of the data is in the box, and the final 25% of the data is to the above of the box. The dots are data points that traditionally considered outliers.[3]

Sometimes I think that box-and-whisker plot obscures too much of the details of the data and we should look at the side-by-side histograms instead.

```
ggplot(TenMileRace, aes(x=net)) +
  geom_histogram() +
  facet_grid( . ~ sex )  # side-by-side plots based on sex

## `stat_bin()` using `bins = 30`.  Pick better value with `binwidth`.
```

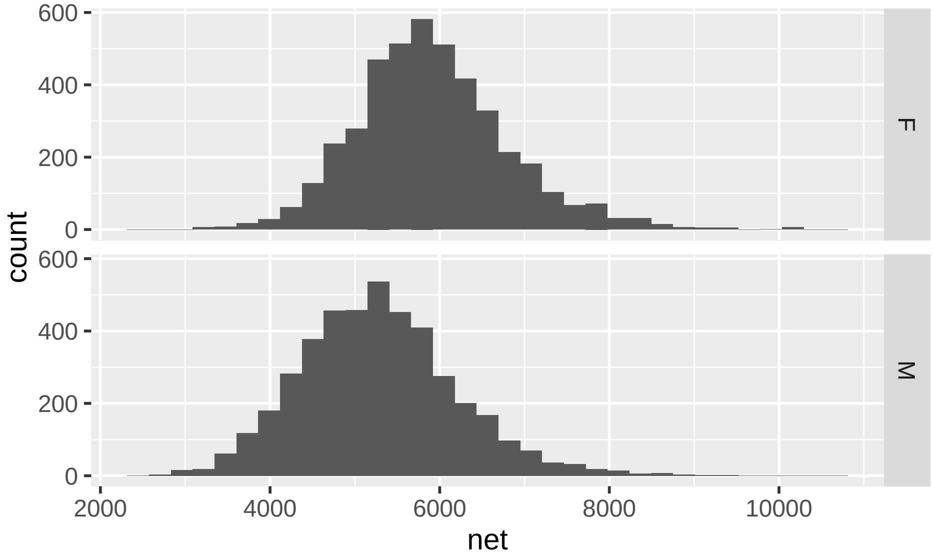


Orientation of graphs can certainly matter. In this case, it makes sense to *stack* the two graphs

[3] Define the Inter-Quartile Range (IQR) as the length of the box. Then any observation more than 1.5*IQR from the box is considered an outlier.

to facilitate comparisons.

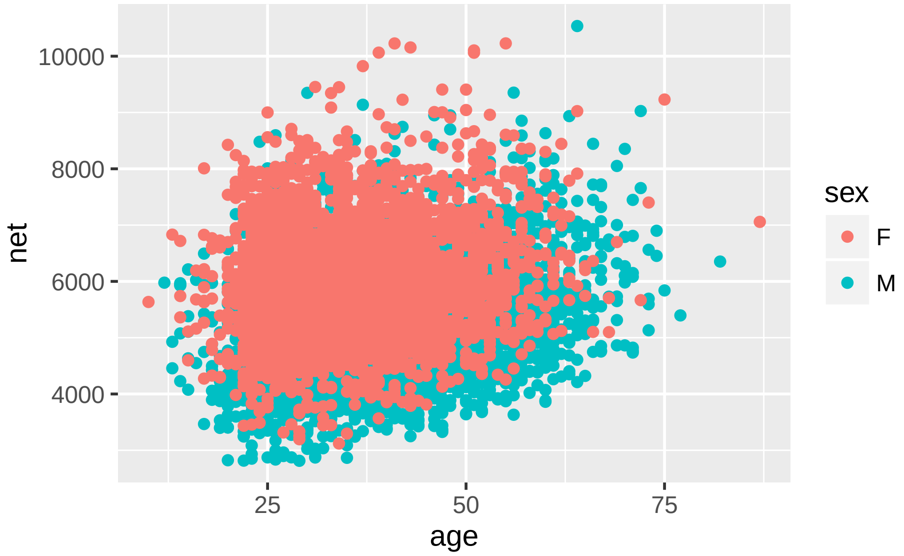
```
ggplot(TenMileRace, aes(x=net)) +
  geom_histogram() +
  facet_grid( sex ~ . )  # side-by-side plots based on sex

## `stat_bin()` using `bins = 30`.  Pick better value with `binwidth`.
```



1.1.4 Bivariate - Continuous vs Continuous

Finally we might want to examine the relationship between two continuous random variables.

```
ggplot(TenMileRace, aes(x=age, y=net, color=sex)) +
  geom_point()
```



1.2 Measures of Centrality

The most basic question to ask of any dataset is 'What is the typical value?' There are several ways to answer that question and they should be familiar to most students.

Mean

Often called the average, or arithmetic mean, we will denote this special statistic with a bar. We define

$$\bar{x}=\frac{1}{n}\sum_{i=1}^{n}x_i=\frac{1}{n}\left(x_1+x_2+\cdots+x_n\right)$$

If we want to find the mean of five numbers $\{3,6,4,8,2\}$ the calculation is

$$\begin{aligned}\bar{x} &= \frac{1}{5}\left(3+6+4+8+2\right)\\ &= \frac{1}{5}\left(23\right)\\ &= 23/5\\ &= 4.6\end{aligned}$$

This can easily be calculated in R by using the function `mean()`. We first extract the column we are interested in using the notation: `DataSet$ColumnName` where the `$` signifies grabbing the column.

```
mean( TenMileRace$net )

## [1] 5599.065
```

Median

If the data were to be ordered, the median would be the middle most observation (or, in the case that n is even, the mean of the two middle most values).

In our simple case of five observations $\{3,6,4,8,2\}$, we first sort the data into $\{2,3,4,6,8\}$ and then the middle observation is clearly 4.

In R the median is easily calculated by the function `median()`.

```
median( TenMileRace$net )

## [1] 5555
```

Mode

This is the observation value with the most number of occurrences.

Examples

- If my father were to become bored with retirement and enroll in my STA 570 course, how would that affect the mean and median age of my 570 students?
  - The mean would move much more than the median. Suppose the class has 5 people right now, ages 21, 22, 23, 23, 24 and therefore the median is 23. When my father joins, the ages will be 21, 22, 23, 23, 24, 72 and the median will remain 23. However, the mean would move because we add in such a large outlier. Whenever we are dealing with skewed data, the mean is pulled toward the outlying observations.

- In 2010, the median NFL player salary was $770,000 while the mean salary was $1.9 million. Why the difference?
  - Because salary data is *skewed* superstar players that make huge salaries (in excess of 20 million) while the minimum salary for a rookie is $375,000. Financial data often reflects a highly skewed distribution and the median is often a better measure of centrality in these cases.

1.3 Measures of Variation

The second question to ask of a dataset is ’How much variability is there?’ Again there are several ways to measure that.

Range

Range is the distance from the largest to the smallest value in the dataset.

```
max( TenMileRace$net ) - min( TenMileRace$net )

## [1] 7722
```

Inter-Quartile Range

The **p-th** percentile is the observation (or observations) that has at most p percent of the observations below it and $(1-p)$ above it, where p is between 0 and 100. The median is the 50th percentile. Often we are interested in splitting the data into four equal sections using the 25th, 50th, and 75th percentiles (which, because it splits the data into four sections, we often call these the 1st, 2nd, and 3rd quartiles).

In general I could be interested in dividing my data up into an arbitrary number of sections, and refer to those as *quantiles* of my data.

```
quantile( TenMileRace$net )

##    0%   25%   50%   75%  100%
##  2814  4950  5555  6169 10536
```

The inter-quartile range (IQR) is defined as the distance from the 3rd quartile to the 1st.

```
IQR( TenMileRace$net )

## [1] 1219
```

Notice that we’ve defined IQR before when we looked at box-and-whisker plots and this is exactly the length of the box.

Variance

One way to measure the spread of a distribution is to ask “what is the average distance of an observation to the mean?” We could define the ith **deviate** as $e_i = x_i - \bar{x}$ and then ask what is the average deviate? The problem with this approach is that the sum (and thus the average) of all deviates is *always* 0.

$$\begin{aligned}\sum_{i=1}^{n}(x_i-\bar{x}) &= \sum_{i=1}^{n}x_i-\sum_{i=1}^{n}\bar{x}\\ &= n\frac{1}{n}\sum_{i=1}^{n}x_i-n\bar{x}\\ &= n\bar{x}-n\bar{x}\\ &= 0\end{aligned}$$

The big problem is that about half the deviates are negative and the others are positive. What we really care is the distance from the mean, not the sign. So we could either take the absolute value, or square it.

There are some really good theoretical reasons to chose the square option[4], so we square the deviates and then find the average deviate size (approximately) and call that the **sample variance**.

$$s^2=\frac{1}{n-1}\sum_{i=1}^{n}\left(x_i-\bar{x}\right)^2$$

Why do we divide by $n-1$ instead of n?

1. If I divide by n, then on average, we would tend to underestimate the population variance σ^2.
2. The reason is because we are using the same set of data to estimate σ^2 as we did to estimate the population mean (μ). If I could use $\frac{1}{n}\sum_{i=1}^{n}\left(x_i-\mu\right)^2$ as my estimator, we would be fine. But since I have to replace μ with $\bar{x}$ we have to pay a price.
3. Because the estimation of σ^2 requires the estimation of one other quantity, and using using that quantity, you only need $n-1$ data points and can then figure out the last one, we have used one *degree of freedom* on estimating the mean and we need to adjust the formula accordingly.

In later chapters we'll give this quantity a different name, so we'll introduce the necessary vocabulary here. Let $e_i=x_i-\bar{x}$ be the *error* left after fitting the sample mean. This is the deviation from the observed value to the "expected value" $\bar{x}$. We can then define the Sum of Squared Error as

$$SSE=\sum_{i=1}^{n}e_i^2$$

and the Mean Squared Error as

$$MSE=\frac{SSE}{df}=\frac{SSE}{n-1}=s^2$$

where $df=n-1$ is the appropriate degrees of freedom.

Calculating the variance of our small sample of five observations $\{3,6,4,8,2\}$, recall that the sample mean was $\bar{x}=4.6$

x_i	$(x_i-\bar{x})$	$(x_i-\bar{x})^2$
3	-1.6	2.56
6	1.4	1.96
4	-0.6	0.36
8	3.4	11.56
2	-2.6	6.76
sum		23.2

[4] First, squared terms are easier to deal with compared to absolute values, but more importantly, the spread of the normal distribution is parameterized via squared distances from the mean. Because the normal distribution is so important, we've chosen to define the sample variance so it matches up with the natural spread parameter of the normal distribution.

and so the sample variance is $23.2/(n-1) = 23.2/4 = 5.8$

Clearly this calculation would get very tedious to do by hand and computers will be much more accurate in these calculations. In R, the sample variance is easily calculated by the function `var()`.

```
var( TenMileRace$net )

## [1] 940233.5
```

Standard Deviation

The biggest problem with the sample variance statistic is that the units are in the original units-*squared*. That means if you are looking at data about car fuel efficiency, then the values would be in mpg^2 which are units that I can't really understand. The solution is to take the positive square root, which we will call the sample standard deviation.

$$s = \sqrt{s^2}$$

But why do we take the jog through through variance? Mathematically the variance is more useful and most distributions (such as the normal) are defined by the variance term. Practically though, standard deviation is easier to think about.

The sample standard deviation is important enough for R to have function that will calculate it for you.

```
sd( TenMileRace$net )

## [1] 969.6564
```

Coefficient of Variation

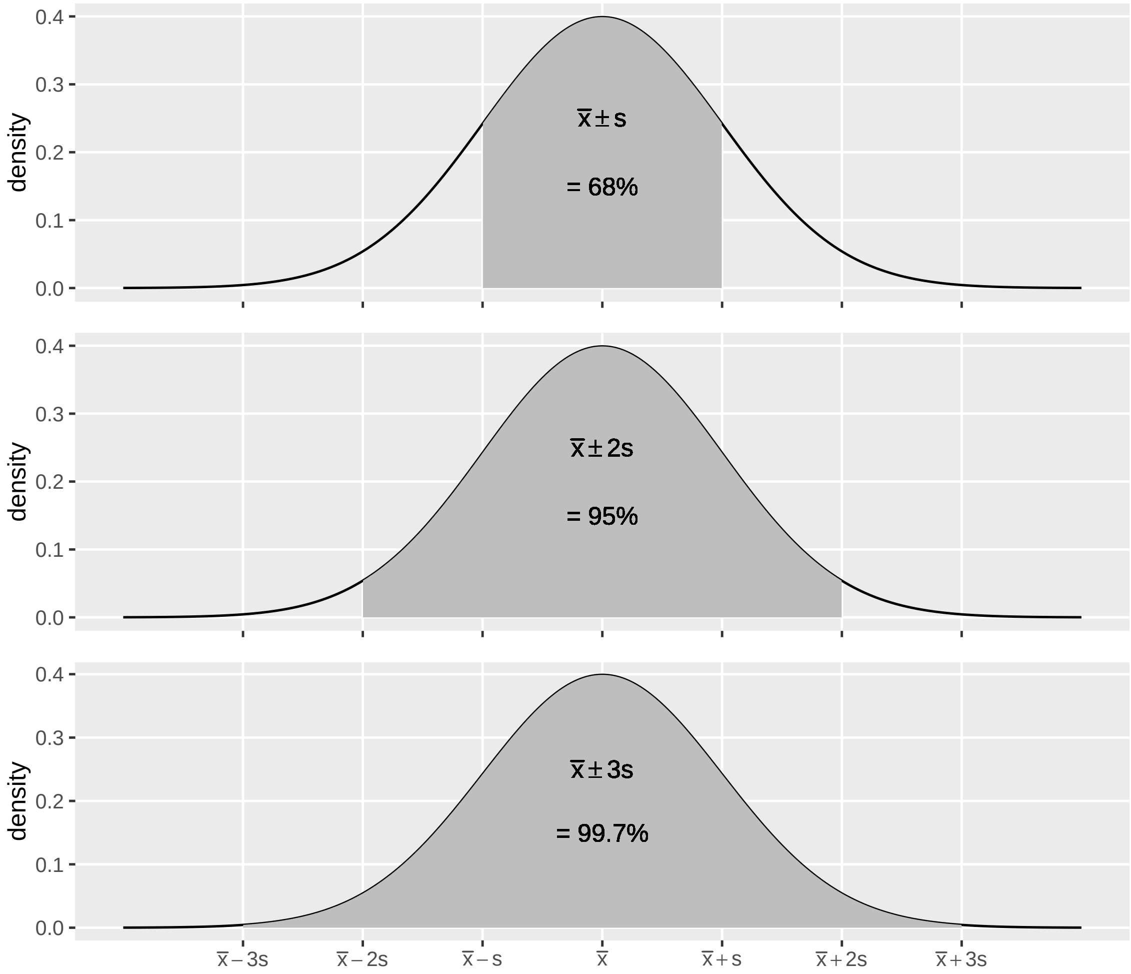
Suppose we had a group of animals and the sample standard deviation of the animals lengths was 15 cm. If the animals were elephants, you would be amazed at their uniformity in size, but if they were insects, you would be astounded at the variability. To account for that, the **coefficient of variation** takes the sample standard deviation and divides by the absolute value of the sample mean (to keep everything positive)

$$CV = \frac{s}{|\bar{x}|}$$

Empirical Rule of Thumb

For any mound-shaped sample of data the following is a reasonable rule of thumb:

Interval	Approximate percent of measurements
$\bar{x} \pm s$	68%
$\bar{x} \pm 2s$	95%
$\bar{x} \pm 3s$	99.7%


density
0.0
0.1
0.2
0.3
0.4
$\bar{x} \pm s$
= 68%
$\bar{x} \pm 2s$
= 95%
$\bar{x} \pm 3s$
= 99.7%
$\bar{x} - 3s$
$\bar{x} - 2s$
$\bar{x} - s$
$\bar{x}$
$\bar{x} + s$
$\bar{x} + 2s$
$\bar{x} + 3s$

Chapter 2

Confidence Intervals Using Bootstrapping

Often our research goal isn't to compare our data to some specific hypothesized value of a parameter, but rather to take the data we've observed and ask "What values of the parameter are consistent with the data?"

2.1 Observational Studies

Unfortunately it is not always possible to perform a designed experiment. It might be unethical (randomly assigning people to receive a dangerous dose of radiation) logistically difficult (assigning a heating treatment to hectare level landscapes), or just too expensive or time consuming.

Instead we often settle for a inferior study method of taking a random sample from the population of interest and observing the relationships between the variables of interest. In the designed experiment, the random assignment to treatment groups was critical to our inference. In an observational study, the random sample from the population is critical trusting that the observed sample data is representative of the population of interest.

As with experimental data, we will want to test whether the observed data (and test statistic d) is consistent with some null hypothesis H_0. However we will have to modify our process slightly to account for the difference between just observing randomly sampled data versus the random assignment of treatments.

Suppose that we had a population of interest and we wish to estimate the mean of that population (the population mean we'll denote as μ). We can't observe every member of the population (which would be prohibitively expensive) so instead we take a random sample and from that sample calculate a sample mean (which we'll denote $\bar{x}$). We believe that $\bar{x}$ will be a good estimator of μ, but it will vary from sample to sample and won't be exactly equal to μ.

Next suppose we wish to ask if a particular value for μ, say μ_0, is consistent with our observed data? We know that $\bar{x}$ will vary from sample to sample, but we have no idea *how much it will vary* between samples. However, if we could understand how much $\bar{x}$ varied sample to sample, we could answer the question. For example, suppose that $\bar{x} = 5$ and we know that $\bar{x}$ varied about ± 2 from sample to sample. Then I'd say that possible values of μ_0 in the interval 3 to 7 (5 ± 2) are reasonable values for μ and anything outside that interval is not reasonable.

Therefore, if we could take many, many repeated samples from the population and calculate our test statistic $\bar{x}$ for each sample, we could rule out possible values of μ. Unfortunately we don't have the time or money to repeatedly sample from the actual population, but we could sample from our best approximation to what the population is like.

```
## Loading required package:  dplyr
##
## Attaching package:  ’dplyr’
##
## The following objects are masked from ’package:stats’:
##
##    filter, lag
##
## The following objects are masked from ’package:base’:
##
##    intersect, setdiff, setequal, union
##
## Loading required package:  lattice
## Loading required package:  ggplot2
## Loading required package:  car
## Loading required package:  mosaicData
##
## Attaching package:  ’mosaic’
##
## The following object is masked from ’package:car’:
##
##    logit
##
## The following objects are masked from ’package:dplyr’:
##
##    count, do, tally
##
## The following objects are masked from ’package:stats’:
##
##    D, IQR, binom.test, cor, cov, fivenum, median, prop.test,
##    quantile, sd, t.test, var
##
## The following objects are masked from ’package:base’:
##
##    max, mean, min, prod, range, sample, sum
```

Suppose we were to sample from a population of shapes, and we observed 4/9 of the sample were squares, 3/9 were circles, and a triangle and a diamond. Then our best guess of what the population that we sampled from was a population with 4/9 squares, 3/9 circles, and 1/9 of triangles and diamonds.

Using this approximated population (which is just many many copies of our sample data), we can repeated sample $\bar{x}^*$ values to create the sampling distribution of $\bar{x}$.

Because our approximate population is just an infinite number of copies of our sample data, then sampling from the approximate population is equivalent to sampling *with replacement* from our sample data. If I take n samples from n distinct objects with replacement, then the process can be thought of as mixing the n objects in a bowl and taking an object at random, noting which it is, replace it into the bowl, and then draw the next sample. Practically, this means some objects will be selected more than once and some will not be chosen at all. To sample our observed data with replacement, we'll use the `resample()` function in the `mosaic` package. We see that some rows will be selected multiple times, and some will not be selected at all.

```
## Loading required package:  grid
```

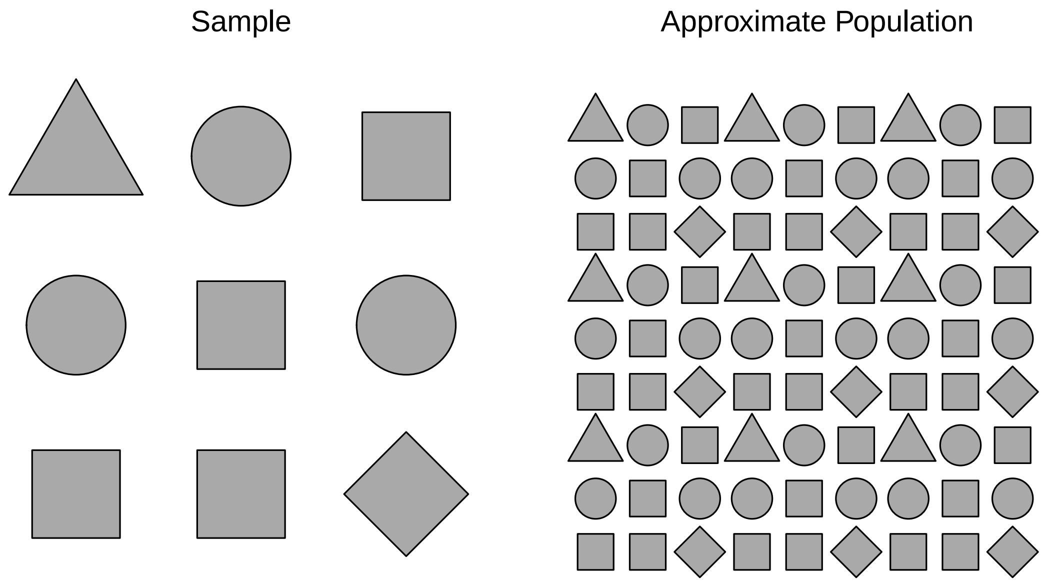


Figure 2.1.1: A possible sample from a population of shapes. Because 4/9 of our sample were squares, our best estimate is that the population is also approximately 4/9 squares. We can think of the approximated population as just many many copies of the observed sample data.

```
Testing.Data <- data.frame(
  name=c('Alison','Brandon','Chelsea','Derek','Elise'))
Testing.Data

##      name
## 1  Alison
## 2 Brandon
## 3 Chelsea
## 4   Derek
## 5   Elise
```

```
# Sample rows from the Testing Data (with replacement)
resample(Testing.Data)

##        name orig.id
## 1    Alison       1
## 4     Derek       4
## 3   Chelsea       3
## 1.1  Alison       1
## 5     Elise       5
```

Notice `Alison` has selected twice, while `Brandon` has not been selected at all. We can use the `resample()` function similarly as we did the `shuffle()` function.

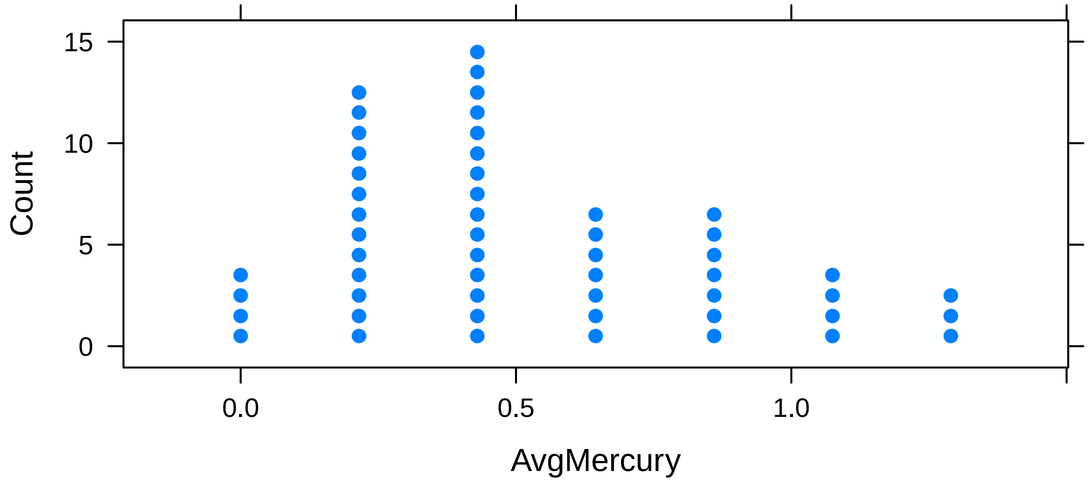
The sampling from the estimated population via sampling from the observed data is called *bootstrapping* because we are making no distributional assumptions about where the data came from, and the idiom "Pulling yourself up by your bootstraps" seemed appropriate.

Example: Mercury Levels in Fish from Florida Lakes

A data set provided by the Lock^5 textbook looks at the mercury levels in fish harvested from lakes in Florida. There are approximately 7,700 lakes in Florida that are larger than 10 acres. As part of a study to assess the average mercury contamination in these lakes, a random sample of $n = 53$ lakes, an unspecified number of fish were harvested and the average mercury level (in ppm) was calculated for fish in each lake. The goal of the study was to assess if the average mercury concentration was greater than the 1969 EPA "legally actionable level" of 0.5 ppm.

```
# as always, our first step is to load the mosaic package
library(mosaic)

# read the Lakes data set
Lakes <- read.csv('http://www.lock5stat.com/datasets/FloridaLakes.csv')
```

```
# make a nice picture
dotPlot( ~ AvgMercury, data=Lakes)
```



We can calculate mean average mercury level for the $n = 53$ lakes

```
mean( ~ AvgMercury, data=Lakes )

## [1] 0.5271698
```

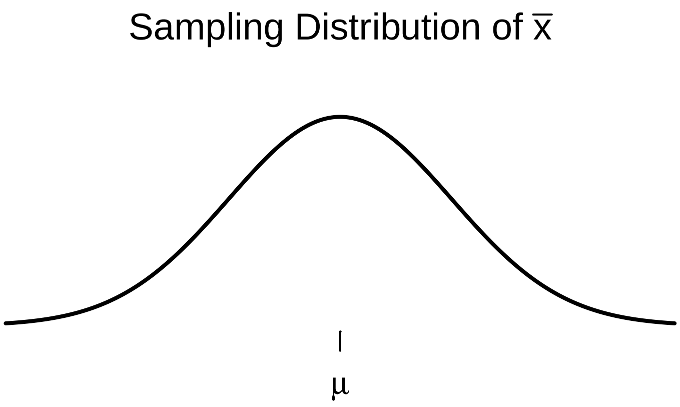
The sample mean is greater than 0.5 but not by too much. Is a true population mean concentration μ_{Hg} that is 0.5 or less incompatible with our observed data? Is our data sufficient evidence to conclude that the average mercury content is greater than 0.5? Perhaps the true average mercury content is less than (or equal to) 0.5 and we just happened to get a random sample that with a mean greater than 0.5?

The first step in answering these questions is to create the sampling distribution of $\bar{x}_{Hg}$. To do this, we will sample from the approximate population of lakes, which is just many many replicated copies of our sample data.

```
# create the sampling distribution of xbar
SamplingDist <- do(10000) * mean( ~ AvgMercury, data=resample(Lakes) )

# what columns does the data frame "SamplingDist" have?
str(SamplingDist)

## Classes 'do.data.frame' and 'data.frame': 10000 obs. of  1 variable:
##  $ mean: num  0.511 0.504 0.525 0.536 0.499 ...
##  - attr(*, "lazy")=List of 2
##   ..$ expr: language mean(~AvgMercury, data = resample(Lakes))
##   ..$ env :<environment: R_GlobalEnv>
##   ..- attr(*, "class")= chr "lazy"
##  - attr(*, "culler")=function (object, ...)

# show a histogram of the sampling distribution of xbar
histogram( ~result, data=SamplingDist, main='Estimated Sampling distribution of xbar' )

## Error in eval(expr, envir, enclos):  object ’result’ not found
```

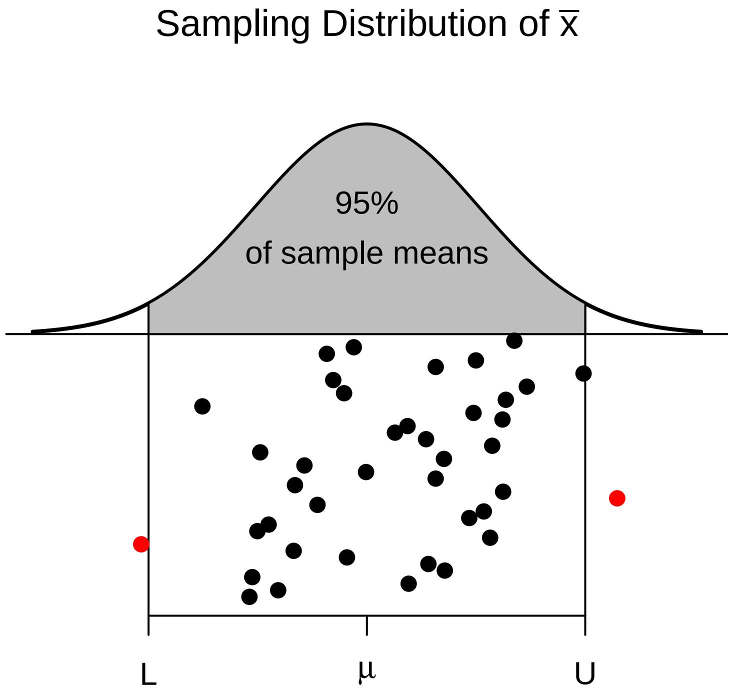
2.2 Using Quantiles of the Estimated Sampling Distributions to create a Confidence Interval

In many cases we have seen, the sampling distribution of a statistic is centered on the parameter we are interested in estimating and is symmetric about that parameter[1]. For example, we expect that the sample mean $\bar{x}$ should be a good estimate of the population mean μ and the sampling distribution of $\bar{x}$ should look something like the following.

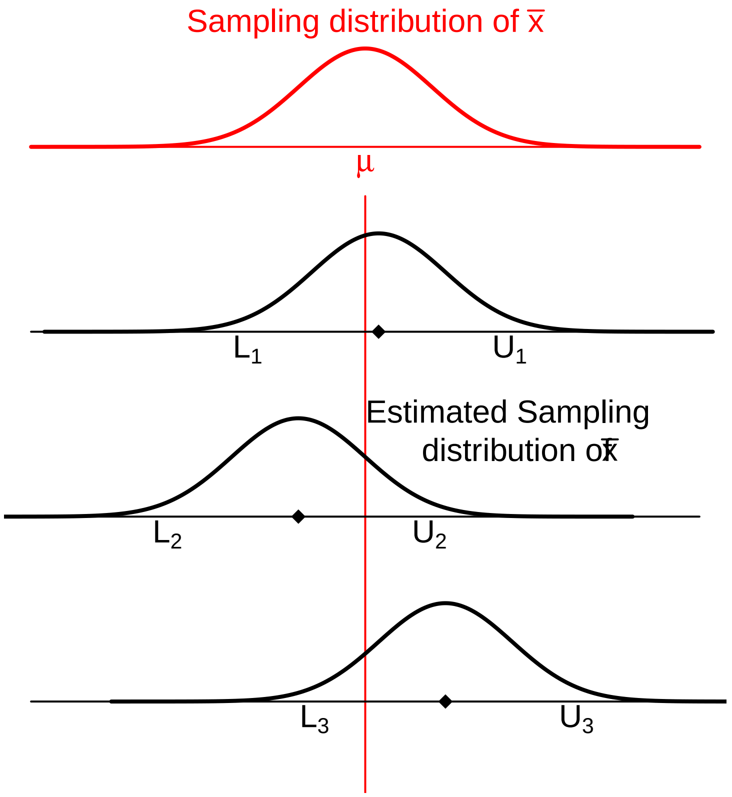


There are two points, (call them L and U) where for our given sample size and population we are sampling from, where we expect that 95% of the sample means to fall within. That is to say, L and U capture the middle 95% of the sampling distribution of $\bar{x}$.

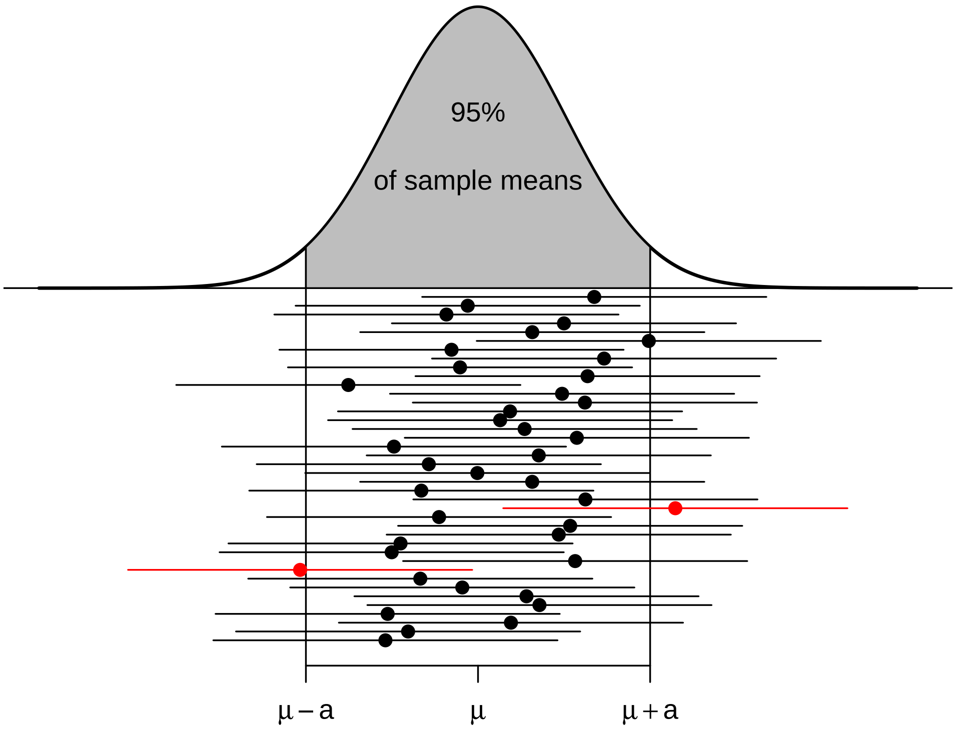
[1] There are actually several ways to create a confidence interval from the estimated sampling distribution. The method presented here is called the “percentile” method and works when the sampling distribution is symmetric and the estimator we are using is unbiased.



These sample means are randomly distributed about the population mean μ. Given our sample data and sample mean $\bar{x}$, we can examine how our *simulated* values of $\bar{x}^*$ vary about $\bar{x}$. I expect that these simulated sample means $\bar{x}^*$ should vary about $\bar{x}$ in the same way that $\bar{x}$ values vary around μ. Below are three estimated sampling distributions that we might obtain from three different samples and their associated sample means.


Sampling distribution of $\bar{x}$
μ
L_1
U_1
Estimated Sampling distribution of $\bar{x}$
L_2
U_2
L_3
U_3

For each possible sample, we could consider creating the estimated sampling distribution of $\bar{x}$ and calculating the L and U values that capture the middle 95% of the estimated sampling distribution. Below are twenty samples, where we've calculated this interval for each sample.



Most of these intervals contain the true parameter μ, that we are trying to estimate. In practice, I will only take one sample and therefore will only calculate one sample mean and one interval, but I want to recognize that the method I used to produce the interval (i.e. take a random sample, calculate the mean and then the interval) will result in intervals where only 95% of those intervals will contain the mean μ. Therefore, I will refer to the interval as a 95% *confidence interval.*

After the sample is taken and the interval is calculated, the numbers lower and upper bounds of the confidence interval are fixed. Because μ is a constant value and the confidence interval is fixed, nothing is changing. To distinguish between a future random event and the fixed (but unknown) outcome of if I ended up with an interval that contains μ and we use the term confidence interval instead of probability interval.

```
# create the sampling distribution of xbar
SamplingDist <- do(10000) * mean( ~ AvgMercury, data=resample(Lakes) )

# what columns does the data frame "SamplingDist" have?
str(SamplingDist)

## Classes 'do.data.frame' and 'data.frame': 10000 obs. of  1 variable:
##  $ mean: num  0.593 0.535 0.543 0.535 0.527 ...
##  - attr(*, "lazy")=List of 2
##   ..$ expr: language mean(~AvgMercury, data = resample(Lakes))
##   ..$ env :<environment: R_GlobalEnv>
##   ..- attr(*, "class")= chr "lazy"
##  - attr(*, "culler")=function (object, ...)

# show a histogram of the sampling distribution of xbar
histogram( ~result, data=SamplingDist, main='Estimated Sampling distribution of xbar' )

## Error in eval(expr, envir, enclos):  object 'result' not found

# calculate the 95% confidence interval using middle 95%
quantile( SamplingDist$result, probs=c(.025, .975) )

##  2.5% 97.5%
##    NA    NA
```

There are several ways to interpret this interval.

1. The process used to calculate this interval (take a random sample, calculate a statistic, repeatedly resample, and take the middle 95%) is a process that results in an interval that contains the parameter of interest on 95% of the samples we could have collected, however we don't know if the particular sample we collected and its resulting interval of (0.44, 0.62) is one of the intervals containing μ.

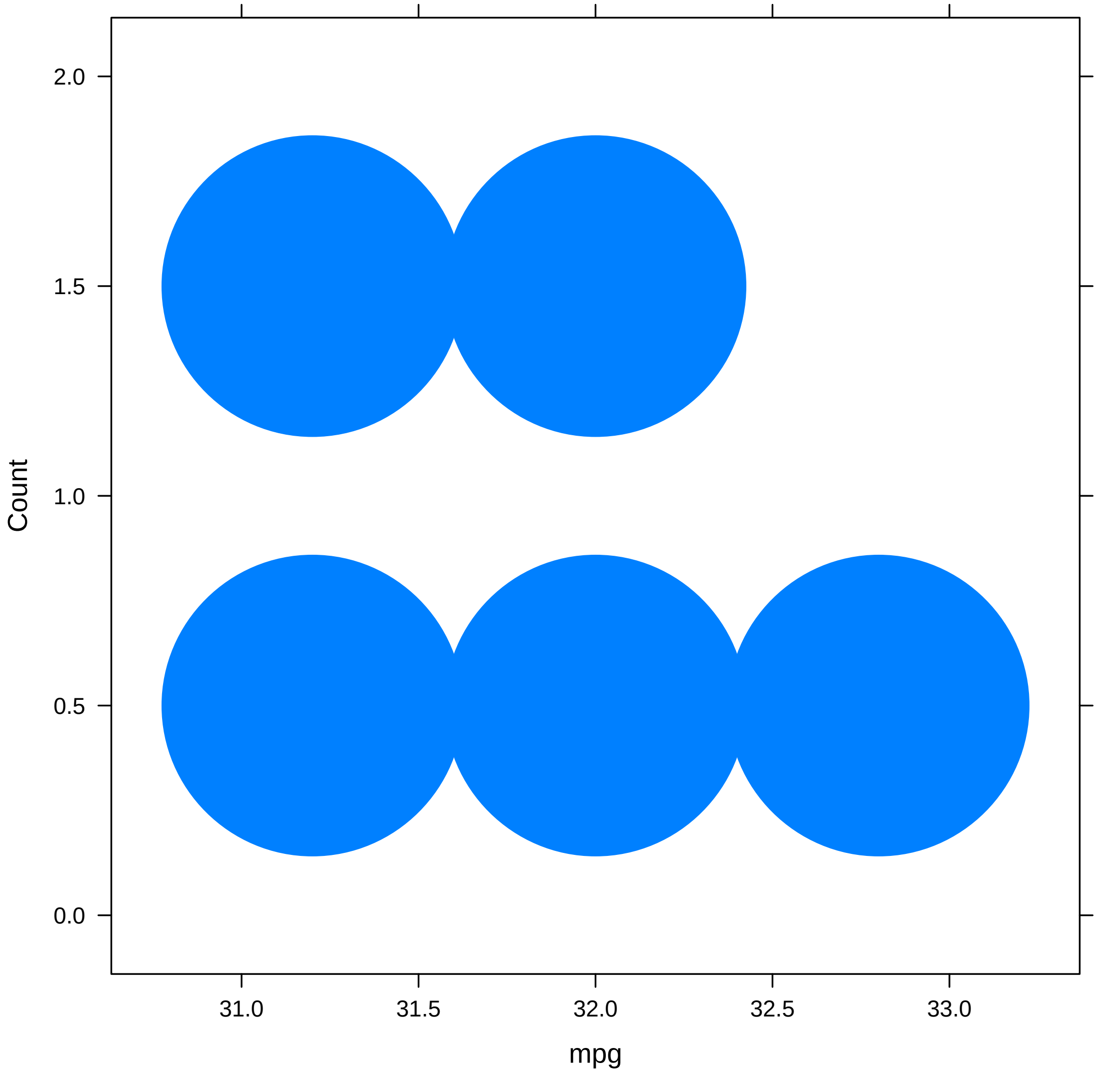
2. We are 95% confident that μ is in the interval (0.44, 0.62). This is delightfully vague and should be interpreted as a shorter version of the previous interpretation.

3. The interval (0.44, 0.62) is the set of values of μ that are consistent with the observed data at the 0.05 threshold of statistical significance for a two-sided hypothesis test.

Example:

Suppose we have data regarding fuel economy of 5 new vehicles of the same make and model and we wish to test if the observed fuel economy is consistent with the advertised 31 mpg at highway speeds. We the data are

```
CarMPG <- data.frame( ID=1:5, mpg = c(31.8, 32.1, 32.5, 30.9, 31.3) )
dotPlot( ~ mpg, data=CarMPG )
```



```
mean( ~ mpg, data=CarMPG )

## [1] 31.72
```

We will use the sample mean to assess if the sample fuel efficiency is consistent with the advertised number. Because these cars could be considered a random sample of all new cars of this make, we will create the estimated sampling distribution using the bootstrap resampling of the data.

```
SamplingDist <- do(10000) * mean( ~ mpg, data=resample(CarMPG) )
histogram(~result, data=SamplingDist,
          main='Estimated Sampling distribution of xbar' )

## Error in eval(expr, envir, enclos):  object 'result' not found
```

```
quantile( SamplingDist$result, probs=c(.025, .975) )

##  2.5% 97.5% 
##    NA    NA
```

We see that the 95% confidence interval is (31.2, 32.2) and does not actually contain the advertised 31 mpg. However, I don't think that in this case we would object to a car manufacturer sell us a car that is *better* than was advertised.

Example

Recall the pollution ratio data from homework 1 (O&L 3.21). The ratio of DDE (related to DDT) to PCB concentrations in bird eggs has been shown to have had a number of biological implications. The ratio is used as an indication of the movement of contamination through the food chain. The paper "The ratio of DDE to PCB concentrations in Great Lakes herring gull eggs and its us in interpreting contaminants data" reports the following ratios for eggs collected at 13 study sites from the five Great Lakes. The eggs were collected from both terrestrial and aquatic feeding birds.

	DDE to PCB Ratio										
Terrestrial	76.50	6.03	3.51	9.96	4.24	7.74	9.54	41.70	1.84	2.5	1.54
Aquatic	0.27	0.61	0.54	0.14	0.63	0.23	0.56	0.48	0.16	0.18	

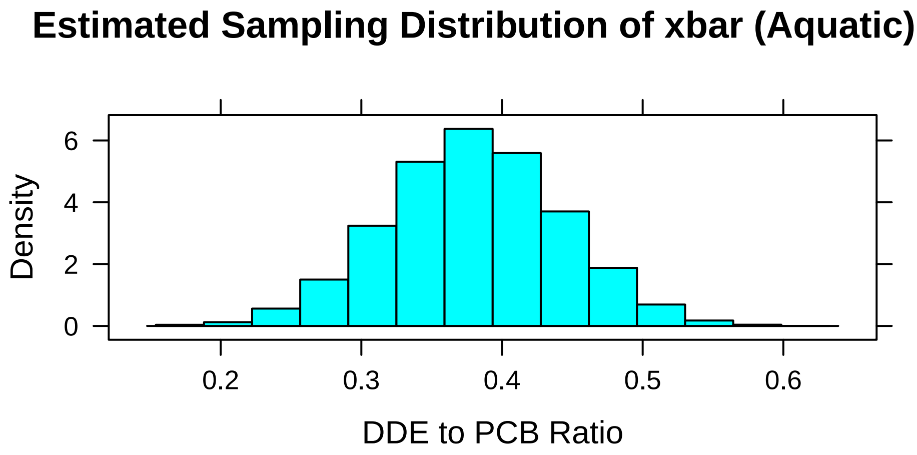
Suppose that the eggs were collected at random and we observe both the ratio and the feeding type. That is to say, we didn't decide to sample 11 Terrestrial birds and 10 Aquatic, that was just how it ended up. To create confidence intervals in that case, we should resample from the 21 eggs.

```
# write a data frame as before
PollutionRatios <- data.frame(
  Ratio = c(76.50, 6.03, 3.51, 9.96, 4.24, 7.74, 9.54, 41.70, 1.84, 2.5, 1.54,
             0.27, 0.61, 0.54, 0.14, 0.63, 0.23, 0.56,  0.48, 0.16, 0.18       ),
  Type  = c( rep('Terrestrial',11), rep('Aquatic',10) ) )

# what happens when I calculate multiple means...
do(3) * mean( Ratio ~ Type, data=resample(PollutionRatios) )

##     Aquatic Terrestrial
## 1 0.3890909    20.31200
## 2 0.4536364    11.40800
## 3 0.2814286    17.35714
```

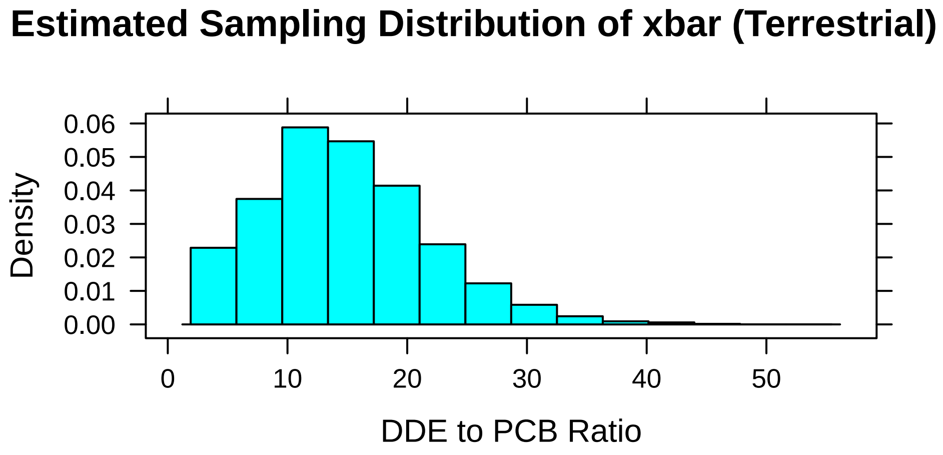
```
SamplingDist <- do(10000) * mean( Ratio ~ Type, data=resample(PollutionRatios) )
```

As always we can look at the estimated sampling distributions of the means

```
histogram( ~ Aquatic, data=SamplingDist,
           main='Estimated Sampling Distribution of xbar (Aquatic)',
           xlab='DDE to PCB Ratio')
```



```
histogram( ~ Terrestrial, data=SamplingDist,
           main='Estimated Sampling Distribution of xbar (Terrestrial)',
           xlab='DDE to PCB Ratio')
```

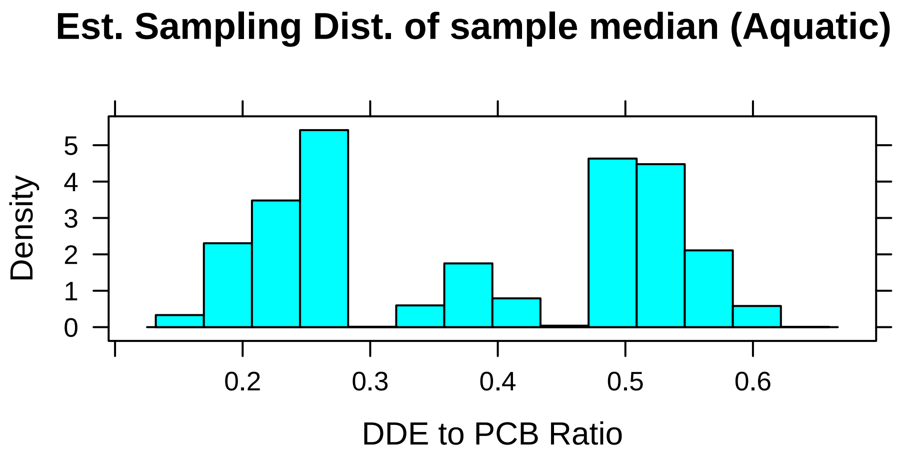


```
# Calculate confidence intervals
quantile( SamplingDist$Aquatic,     probs=c(.025, .975) )

##      2.5%     97.5%
## 0.2566667 0.5011111

quantile( SamplingDist$Terrestrial, probs=c(.025, .975) )

##      2.5%     97.5%
##  4.351103 30.827600
```

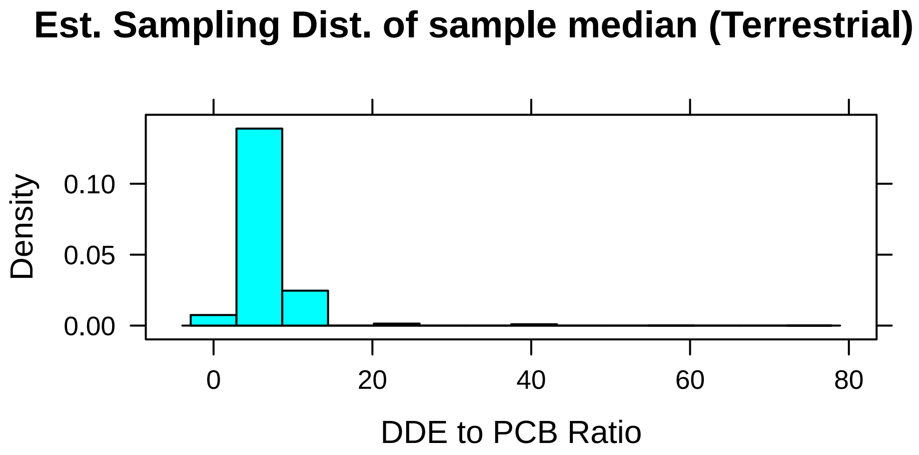
Because the terrestrial eggs had those large outliers we had recommend that the median might be a better measure of the "center" of the terrestrial observations. With the resampling method of calculating confidence intervals, it is easy to create a 95% confidence interval for the medians.

```
SamplingDist <- do(10000) * median( Ratio ~ Type, data=resample(PollutionRatios) )
```

```
histogram( ~ Aquatic, data=SamplingDist,
           main='Est. Sampling Dist. of sample median (Aquatic)',
           xlab='DDE to PCB Ratio')
```



```
histogram( ~ Terrestrial, data=SamplingDist,
           main='Est. Sampling Dist. of sample median (Terrestrial)',
           xlab='DDE to PCB Ratio')
```



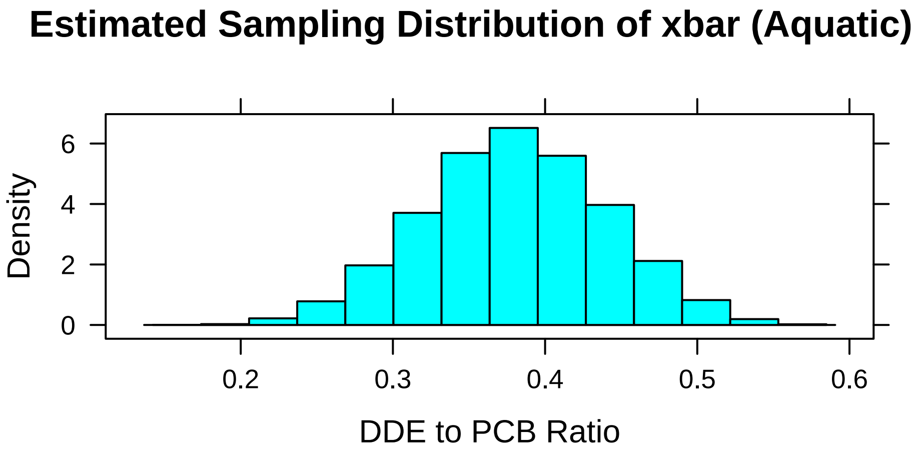
```
# Calculate confidence intervals
quantile( SamplingDist$Aquatic,     probs=c(.025, .975) )

##  2.5% 97.5%
## 0.180 0.575

quantile( SamplingDist$Terrestrial, probs=c(.025, .975) )

##  2.5% 97.5%
##  2.50  9.96
```

Suppose that the researchers had *deliberately* chosen to sample 11 terrestrial birds and 10 aquatic. Then our resampling method should respect that choice and always produce datasets with 11 terrestrial and 10 aquatic eggs. This can be done by the `groups=` argument to the resample command.

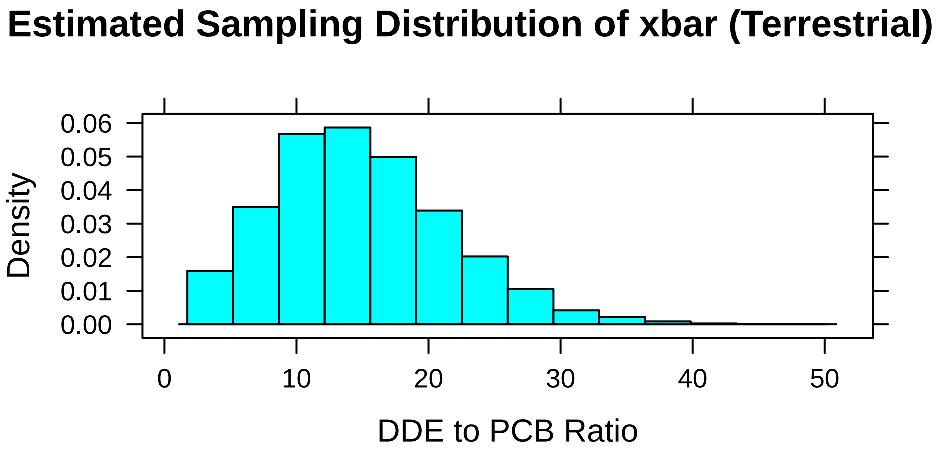
```
SamplingDist <- do(10000) *
   mean( Ratio ~ Type,
         data = resample(PollutionRatios, groups=Type) )
```

As always we can look at the estimated sampling distributions of the means

```
histogram( ~ Aquatic, data=SamplingDist,
           main='Estimated Sampling Distribution of xbar (Aquatic)',
           xlab='DDE to PCB Ratio')
```



```
histogram( ~ Terrestrial, data=SamplingDist,
           main='Estimated Sampling Distribution of xbar (Terrestrial)',
           xlab='DDE to PCB Ratio')
```



```
# Calculate confidence intervals
quantile( SamplingDist$Aquatic,     probs=c(.025, .975) )

##  2.5% 97.5%
## 0.262 0.497

quantile( SamplingDist$Terrestrial, probs=c(.025, .975) )

##      2.5%     97.5%
##  4.507227 30.361068
```

In this case, the difference between requiring the bootstrap datasets to conform to the 11 Terrestrial and 10 Aquatic didn't make much of a difference because that is what would happen on average sampling from the full 21 observations, but it could make a difference if the two groups were not as balanced.