STAT216 Activity 3: Flathead Lake Fish

Synopsis

In this activity we will analyze data collected by Montana Fish, Wildlife, and Parks about the population of fish in Flathead Lake. The data that you will load below was downloaded from the FWP data website (if the link doesn't open in Eagle Online, right-click and select open in a new tab.) This data is all from the year 2021.

We want to address a few questions concerning fish in Flathead Lake:

- What percentage of all fish in the lake are lake trout?
- What is the average weight of a lake trout in Flathead lake?

Notice that both of these questions are about populations of fish, but we only have a sample from FWP. In this activity, we will investigate sampling distributions and dip our toes into the water for seeing how sampling distributions are used to use sample data to make inferences about populations.

Getting started

Load packages

Fire up RStudio. We'll use the `tidyverse`, `openintro` and `infer` packages in this lab. We haven't used `infer` yet, so we need to install it first. Depending on your computer/browser combo, you may need to reinstall the other packages as well.

```
install.packages("infer")
```

Next, load `tidyverse`, `openintro`, and `infer` from console. You'll also want to load the packages at the beginning of your document as follows.

```
library(tidyverse)
library(openintro)
library(infer)
set.seed( #PUT YOUR OWN NUMBER IN HERE#)
```

In this report we will be selecting some random samples using R. To make the results consistent, we need to set a seed at the beginning of the report so that the same samples are generated every time you *knit* your document.

A note on setting a seed: Setting a seed will cause R to select the same sample each time you knit your document. This will make sure your results don't change each time you knit, and it will also ensure reproducibility of your work (by setting the same seed it will be possible to reproduce your results). You can set a seed like this:

```
set.seed(48382)
```

The number above is completely arbitrary. You should pick your own seed for your lab report. If you need inspiration, you can use your ID, birthday, or just a random string of numbers. The important thing is that you use each seed only once in a document. Remember to do this **before** you take any random samples below.

1. Install `infer`, load all packages, and set your own seed at the beginning of your lab report.

Creating a reproducible lab report

We will use R Markdown to create a lab report. In RStudio, go to New File -> R Markdown. Then, choose "From Template" and select `Lab Report for OpenIntro Statistics Labs`.

For a bit of additional assistance, check out this video created by OpenIntro (if the link doesn't open in Eagle Online, right-click and open in a new tab).

Don't forget to load the `tidyverse`, `openintro`, and `infer` packages at the beginning of your lab report!

Note: For each exercise, you should have a subsection that begins with `### Exercise` followed by the exercise number and a completely blank like. As an example,

```
[End of Exercise 1 work.]

### Exercise 2

[Your work here.]
```

Any time you want to make calculations using R in your Markdown document, remember that you can create a code chunk by writing ```` ```{r ChodeChunkName} ```` on one line, followed by your code on a new line, then close the code chunk with ```` ``` ```` on another new line. Be sure to give every code chunk a unique name!

The data

Next, we need to load the data into RStudio. Start an R code chunk in your Markdown document, then copy and paste the following into your code chunk:

```
url <- "https://raw.githubusercontent.com/ckaterba/STAT216_activity_data/main/flathead_fish.csv"
if(!file.exists("./flathead_fish.csv")){
  download.file(url, "flathead_fish.csv")}
df <- as_tibble(read.csv("flathead_fish.csv"))
```

Once you've loaded the data, we suggest looking at it using the command `view(df)` in the console.

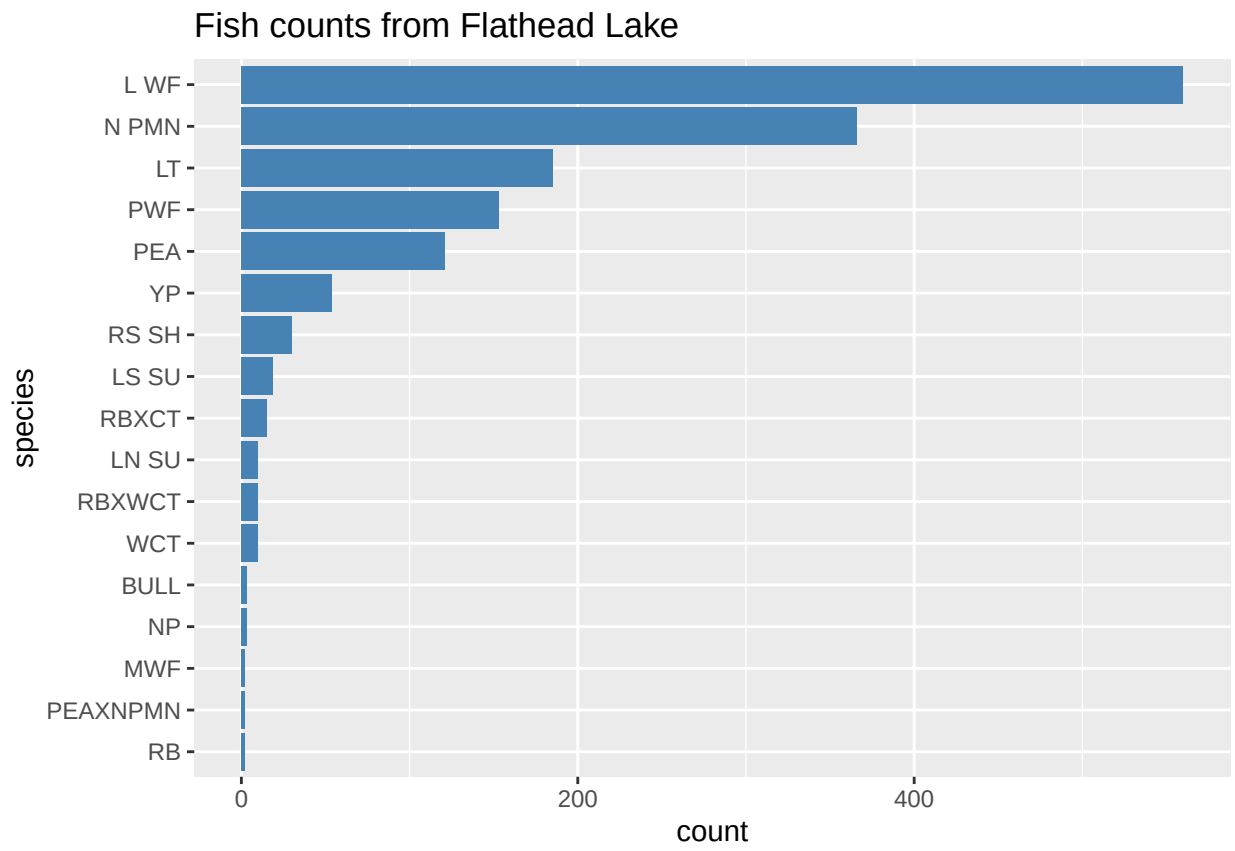
The data set `df` has 3 variables: `species`, `length`, and `weight`. Lengths are measured in millimeters and weights in grams. The `species` variable takes on 17 values, they are abbreviations of fish species:

Abbreviation	Fish
L WF	lake whitefish
N PMN	northern pikeminnow
LT	lake trout
PWF	pygmy whitefish
PEA	peamouth
YP	yellow perch
RS SH	redside shiner
LS SU	largescale sucker
RBXCT	rainbow cross cutthroat trout
WCT	westslope cutthroat trout
LN SU	longnose sucker
RBXWCT	rainbow cross WC trout
BULL	bull trout
NP	northern pike
MWF	mountain whitefish
PEAXNPMN	peamouth cross northern pike minnow
RB	rainbow trout

Note: this may not be all species of fish in the lake!

The code below helps us visualize the sample of fish we have from the lake.

```
# fct_rev(fct_infreq(species)) is shenanigans to make the bar chart display the
# highest count at the top
ggplot(df, aes(x = fct_rev(fct_infreq(species)))) + geom_bar(fill = "steelblue")  + coord_flip() +
  labs(x="species", title = "Fish counts from Flathead Lake ")
```



We can use mutate and summarize to get summary statistics from each species.

```
df %>%
  group_by(species) %>%
  summarize( number = n(),
             avgWeight = mean(weight),
             avgLength = mean(length)
             ) %>%
  mutate( proportion = number / sum(number), .after = number) %>%
  arrange(desc(number))
```

```
## # A tibble: 17 x 5
##    species  number proportion avgWeight avgLength
##    <chr>     <int>      <dbl>     <dbl>     <dbl>
##  1 L WF        560     0.362       584.      371.
##  2 N PMN       366     0.237       228.      273.
##  3 LT          185     0.120      1503.      560.
##  4 PWF         153     0.0990       13.4     126.
##  5 PEA         121     0.0783      142.      241.
```

```
##  6 YP           54    0.0350       88.4     184.
##  7 RS SH        30    0.0194      145.       95.2
##  8 LS SU        19    0.0123      945.      438.
##  9 RBXCT        15    0.00971     336.      327
## 10 LN SU        10    0.00647     463.      356.
## 11 RBXWCT       10    0.00647     258.      303
## 12 WCT          10    0.00647     331.      328.
## 13 BULL          3    0.00194     925.      437.
## 14 NP            3    0.00194    1277.      531.
## 15 MWF           2    0.00129      96       234
## 16 PEAXNPMN      2    0.00129       6        93
## 17 RB            2    0.00129     268       304
```

```
#%>%
 # kable()
```

Remember the table above provides ***sample statistics***. They can serve as point estimates, but we want to make more useful estimates of population parameters, we want to be able to provide a range of values depending on the amount of variability we anticipate in our samples.

1. How many fish are in our sample? How many are lake trout? Do you think this is a good, representative sample of fish from the lake? What about of lake trout?

Bootstraping

One of the key distinctions to make in statistics is the difference between a population distribution and a sampling distribution. The former describes the distribution of measurements *of individuals in a population* (like the weight of lake trout in Flathead lake) and the latter describes a a distribution of *sample statistics* (like the average weight of a sample of 30 lake trout collected from Flathead lake). **Sampling distributions are the main tools for performing statistical inference because they describe the amount of variability we should expect between samples**.

In this class, we will use employ many theoretical distributions to model sampling distributions. This works well very often in practice. For instance, the Central Limit Theorem provides relatively easy-to-meet criteria for concluding that certain sampling distributions are actually normally distributed; the CLT also gives a recipe for the mean and standard error.

There are, however, certain circumstances out there where statisticians cannot employ theoretical distributions to model sampling distributions. There are various techniques for navigating this dilemma and one of them is called **bootstrapping**.

To **bootstrap** a sampling distribution, say you start with a sample S with n observations. Next, you want to construct many, many samples from S, sampling *with replacement*. Exactly how many samples you create depends on your time and computing power available, along with they type of calculations you're performing. Typically, your new collection of samples each consists of n observations as well, which is why you sample *with replacement* (otherwise, each sample would be identical to S, just shuffled). Calculate and record the sample statistic of interest for each new sample, then analyze the distribution of these sample statistics. This is a model of your sampling distribution.

In what follows, we could very well use theoretical methods, but we will bootstrap sampling distributions with smaller sample sizes to see how distributions change with sample size and to check the predictions the central limit theorem makes.

For a categorical variable

Our goal in this section is to bootstrap a sampling distribution to answer the first question posed in the synopsis: what percentage of fish in Flathead lake are lake trout?

The overall proportion of lake trout in our big sample is about 12%, so the average sample proportion will be about 12%. We want to bootstrap the sampling distribution to get an understanding of the *variability among sample proportions*.

As a first step, let's take two samples, both with size $n = 25$.

```
sample1 <- df %>%
  slice_sample(n = 25) %>% #20 random fish from our original sample
  mutate( species = if_else(species == "LT", "LT", "other")) %>% #change species to bernoulli for LT
  count(species) %>% # counts species
  mutate(p_hat = n / sum(n)) #adds sample proportion column
sample1 %>% kable() #prints table in a well-formated way (not necessary)
```

species	n	p_hat
LT	1	0.04
other	24	0.96

```
sample2 <- df %>%
  slice_sample(n = 25) %>%
  mutate( species = if_else(species == "LT", "LT", "other")) %>%
  count(species) %>%
  mutate(p_hat = n / sum(n))

sample2 %>% kable()
```

species	n	p_hat
LT	4	0.16
other	21	0.84

Since our sample size is fairly small, it is no surprise that these two samples have quite different proportions of lake trout.

2. Copy the code for `sample1` into your own code chunk, but change the sample size to 50. Do you expect the sample proportion to be close to 12%? Explain your answer. Run your code chunk to display you actual observed sample proportion $\hat{p}$. (Feel free to remove everything following the #'s as these are only comments to help you see what the code is doing.)

Now, let's bootstrap our sampling distribution. The code below samples 50 fish from our original data set 5000 times, sampling each time with replacement.

```
sample_props50 <- df %>%
  rep_sample_n(size = 50, reps =5000, replace = TRUE) %>%  #5000 samples of 50 fish
  mutate( species = if_else(species == "LT", "LT", "other")) %>% #replaces species to LT or not
  count(species) %>% #shortcut summary to provide count of species variable
  mutate(p_hat = n / sum(n)) %>% # adds sample proportion column
  filter(species == "LT")  #subsets only rows corresponding to lake trout
```

Let's take a peek at the first six entries in `sample_props50`. The `replicate` column indicates the sample number and `n` is the number of lake trout out of 50 in each replicate.

```
head(sample_props50) %>% kable()
```

replicate	species	n	p_hat
1	LT	7	0.14
2	LT	6	0.12
3	LT	3	0.06
4	LT	3	0.06
5	LT	8	0.16
6	LT	7	0.14

We can use `sample_props50` to estimate the average sample proportion and the standard error (ie the standard deviation of the sampling distribution) with $n = 50$.

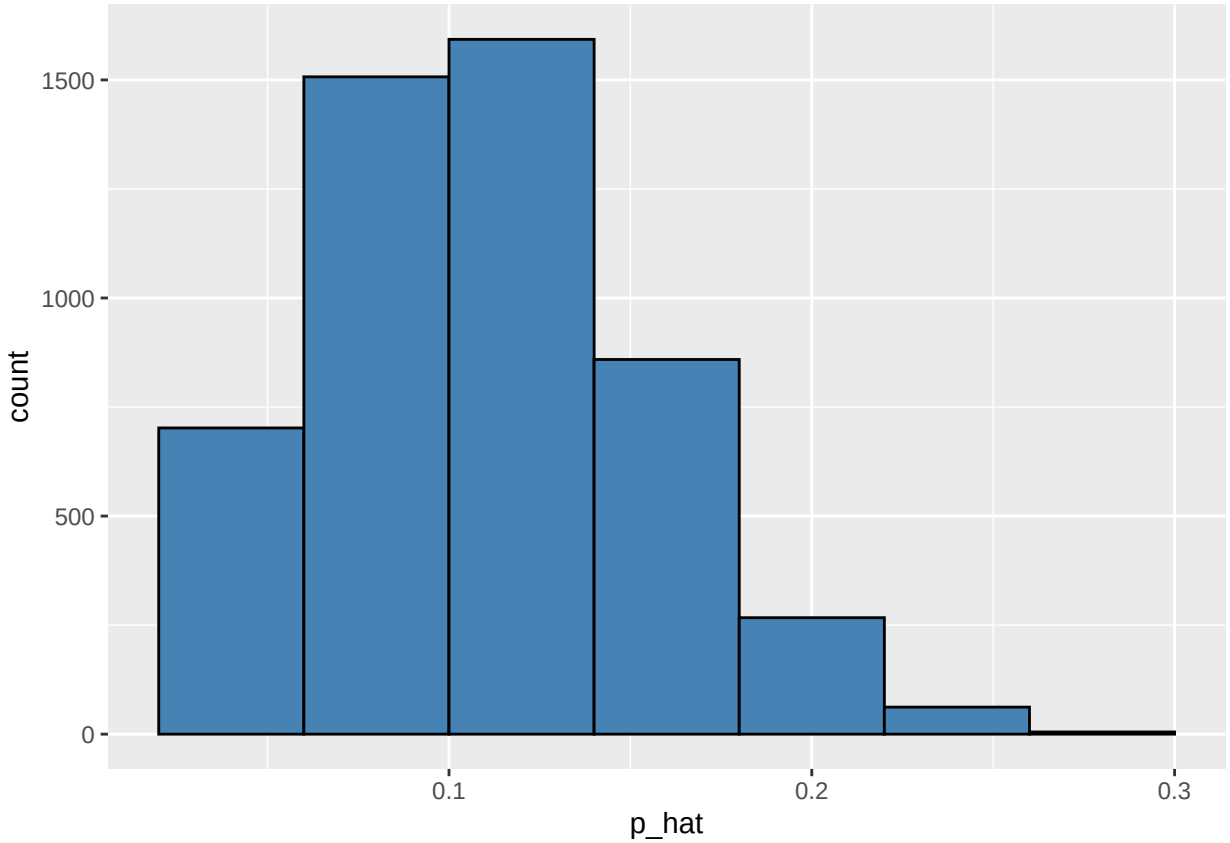
```
sample_props50 %>%
  ungroup() %>%
  summarize( mean_p_hat = mean(p_hat),
             SE_p_hat = sd(p_hat)) %>%
  kable()
```

mean_p_hat	SE_p_hat
0.1188468	0.0456163

Thus, the average sample proportion is about 11.9% and we can expect a change in proportion of about plus or minus 4.6% *between samples*.

Now let's visualize our bootstrapped sampling distribution with a histogram.

```
ggplot(sample_props50, aes(x = p_hat)) +
  geom_histogram(bins = 8, color = "black", fill = "steelblue")
```



Our histogram is right skewed. Is this surprising? No! Recall the Central Limit Theorem for Proportions.

Central Limit Theorem for Proportions:

When your observations are independent and the sample size n is sufficiently large, the sampling distribution of $\hat{p}$ is approximately normally distributed with

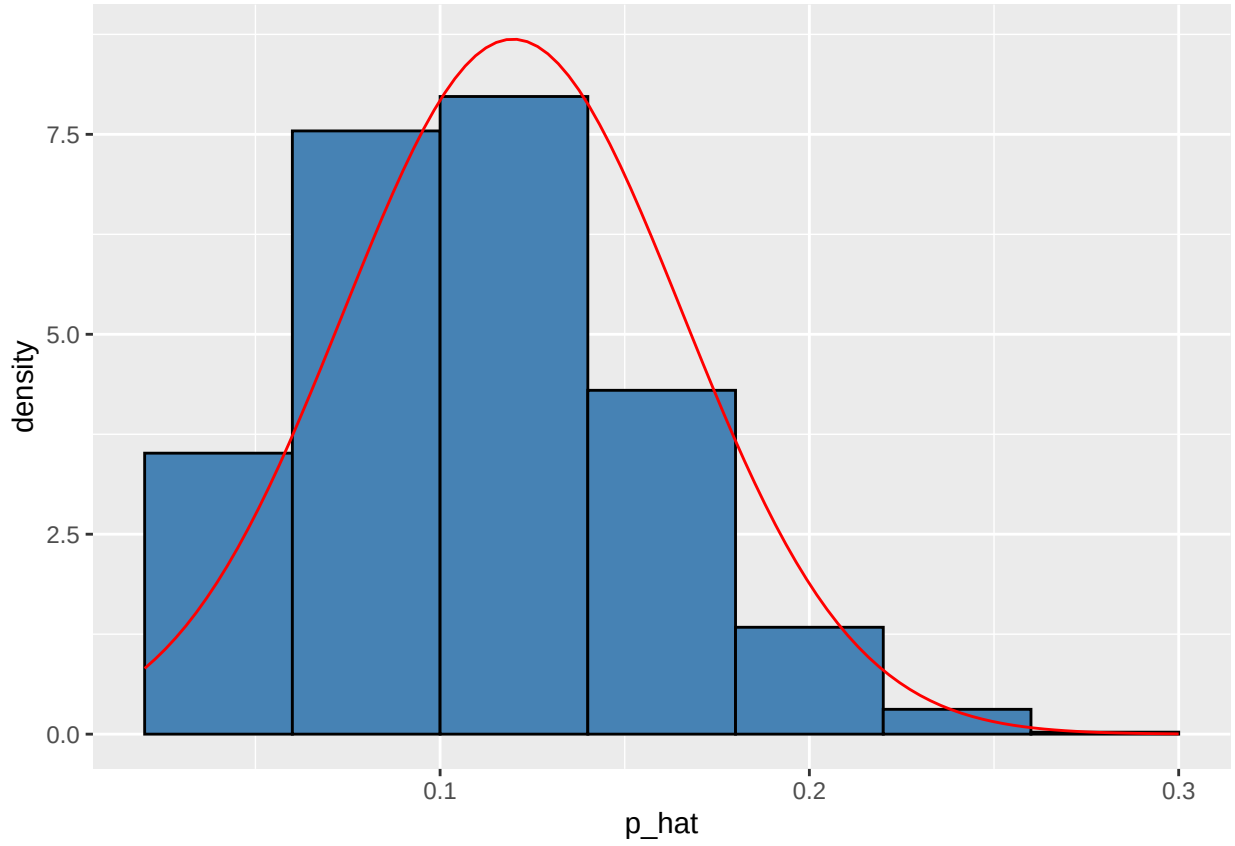
$$\mu_{\hat{p}} = p \quad \text{and} \quad SE_{\hat{p}} = \sqrt{\frac{p(1-p)}{n}}$$

where p is the true population proportion.

The sample size is sufficiently large when your sample has at least 10 (expected) successes and at least 10 (expected) failures

In our case, we expect about 6 lake trout, since about 12% of the large sample were lake trout. Thus, $n = 50$ isn't large enough to guarantee normality by the central limit theorem! The plot below shows our bootstraped sampling distribution with the normal curve the CLT would imply (if our sample size met the criteria).

```
ggplot(sample_props50, aes(x = p_hat)) +
  geom_histogram(aes(y = ..density..), bins = 8, color = "black", fill = "steelblue") +
  geom_function(fun = dnorm, args = list(mean = .1197, sd = sqrt(.1197*(1-.1197)/50)), color = "red")
```



Not the best fit we've seen.

3. Copy and paste the code defining `sample_props50` into your own code chunk, but change the sample size from 50 to 200. (again, you can delete everything on a line after a # since it's just a comment to help you).
   - What is the your average sample proportion? How does this compare to what the CLT predicts?
   - What is your estimated standard error? How does this compare to what the CLT predicts?
   - Make a histogram to visualize your bootstrapped sampling distribution. Does it look more normal to you?

Finally, we can answer our first question: what proportion of fish in Flathead lake are lake trout? Recall that 95% of all measurements fall within 1.96 standard deviations of the mean in a normal distribution. Applying this to your bootstrapped and approximately normal sampling distribution, about 95% of a sample proportions fall with in 1.96 standard deviations of the mean of the sampling distribution. This observation allows us to incorporate the innate variability due to sampling while providing a range of plausible values for the true population proportion.

4. Use your standard error estimate from the previous question to give a range of plausible values of the true proportion of lake trout in Flathead lake. The formula use should use is point est $\pm 1.96 \times SE$ for a 95% confidence level. Interpret this confidence interval using complete sentences.

Note: As mentioned above, we could have made an estimate very similar only using theoretical techniques in this case. This example serves as a computational check on the central limit theorem and as a theoretical check on bootstrapping.

For a numerical variable

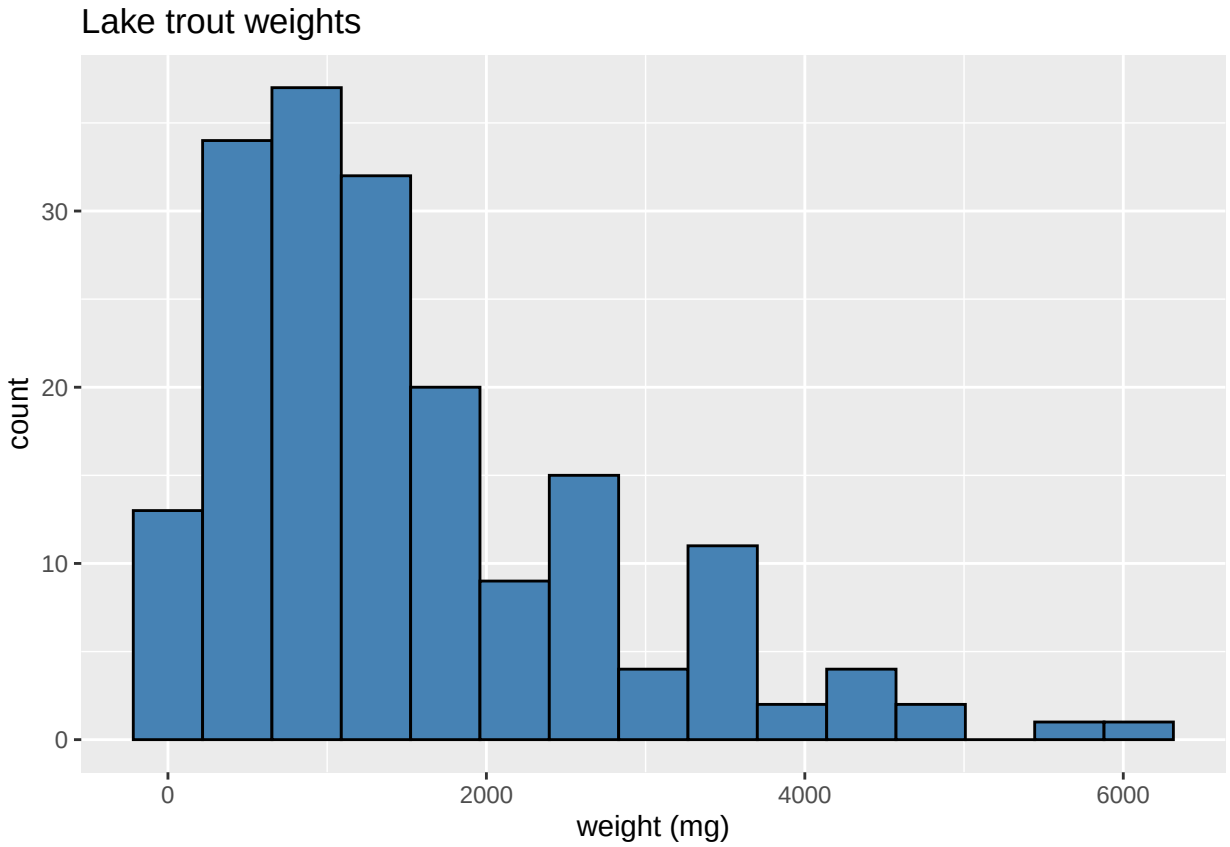
Now we'll use bootstrapping to provide a range of plausible values for the true average weight of a lake trout in Flathead lake. Since we're only interested in lake trout, we need to select these from our original data set. We can do this in two equivalent ways:

```
lakeTrout <- subset(df, species == "LT")
#or
lakeTrout <- df %>% filter(species == "LT")
#print summary statistics of LT weights
summary(lakeTrout$weight)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##      51     649    1154    1503    2096    6148
```

The histogram below shows the distribution of lake trout weights.

```
ggplot(lakeTrout, aes(x = weight)) + geom_histogram(color = "black", fill= "steelblue", bins = 15)  +
  labs(x = "weight (mg)", title = "Lake trout weights")
```



This distribution is unimodal, but right-skewed. We could think of this histogram as an approximation of the population distribution. To make inferences, however, we need to use the sampling distribution.

Again, we will bootstrap the sampling distribution.

5. Before reading on, what do you think the sampling distribution will look like? What do you think the mean sample average will be? Try to relate these values from our sample of lake trout.

This time, we will use the entire sample of lake trout in our bootstrap procedure.

```
xbarBootStrap <- lakeTrout %>%
  rep_sample_n(size = dim(lakeTrout)[1], reps = 50000, replace = TRUE) %>%
  select(replicate, weight) %>%
  summarize(xbar = mean(weight))

head(xbarBootStrap) %>% kable()
```

replicate	xbar
1	1614.849
2	1499.389
3	1385.027
4	1440.319
5	1581.746
6	1573.038

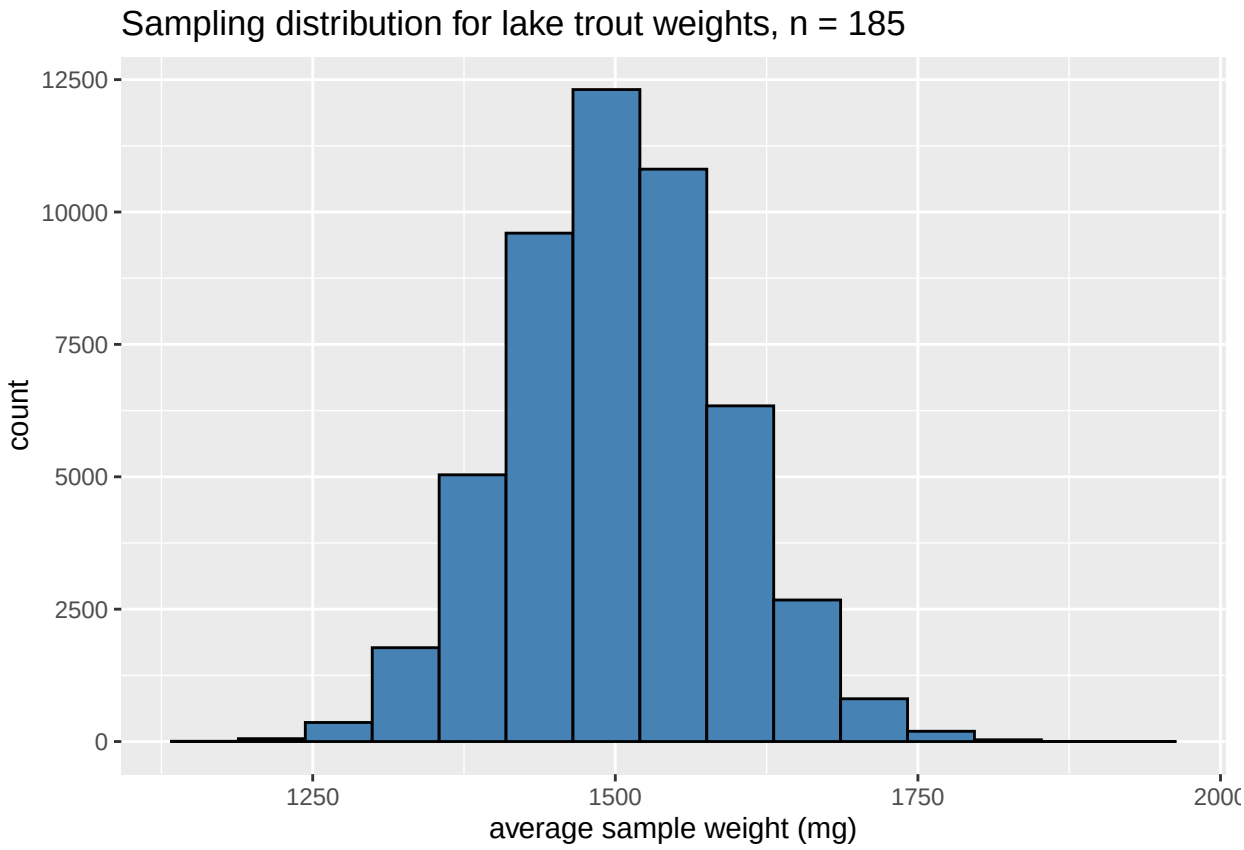
The mean sample average and estimated standard error are

```
round( c( mean(xbarBootStrap$xbar), sd(xbarBootStrap$xbar)), 3)
```

```
## [1] 1502.812   87.905
```

The histogram below approximates the sampling distribtion of $\overline{x}$.

```
ggplot(xbarBootStrap, aes(x = xbar)) + geom_histogram(color = "black", fill= "steelblue", bins = 15)  +
  labs(x = "average sample weight (mg)", title = "Sampling distribution for lake trout weights, n = 185
```



And, lo! The sampling distribution is approximately normal again - even though the population distribution appears to right skewed. We're witnessing the central limit theorem in action again.

Central Limit Theorem for Means:

When your observations are independent and the sample size n is sufficiently large, the sampling distribution of $\overline{x}$ is approximately normally distributed with

$$\mu_{\overline{x}} = \mu \quad \text{and} \quad SE_{\overline{x}} = \sqrt{\frac{\sigma^2}{n}} = \frac{\sigma}{\sqrt{n}} \approx \frac{s}{\sqrt{n}}$$

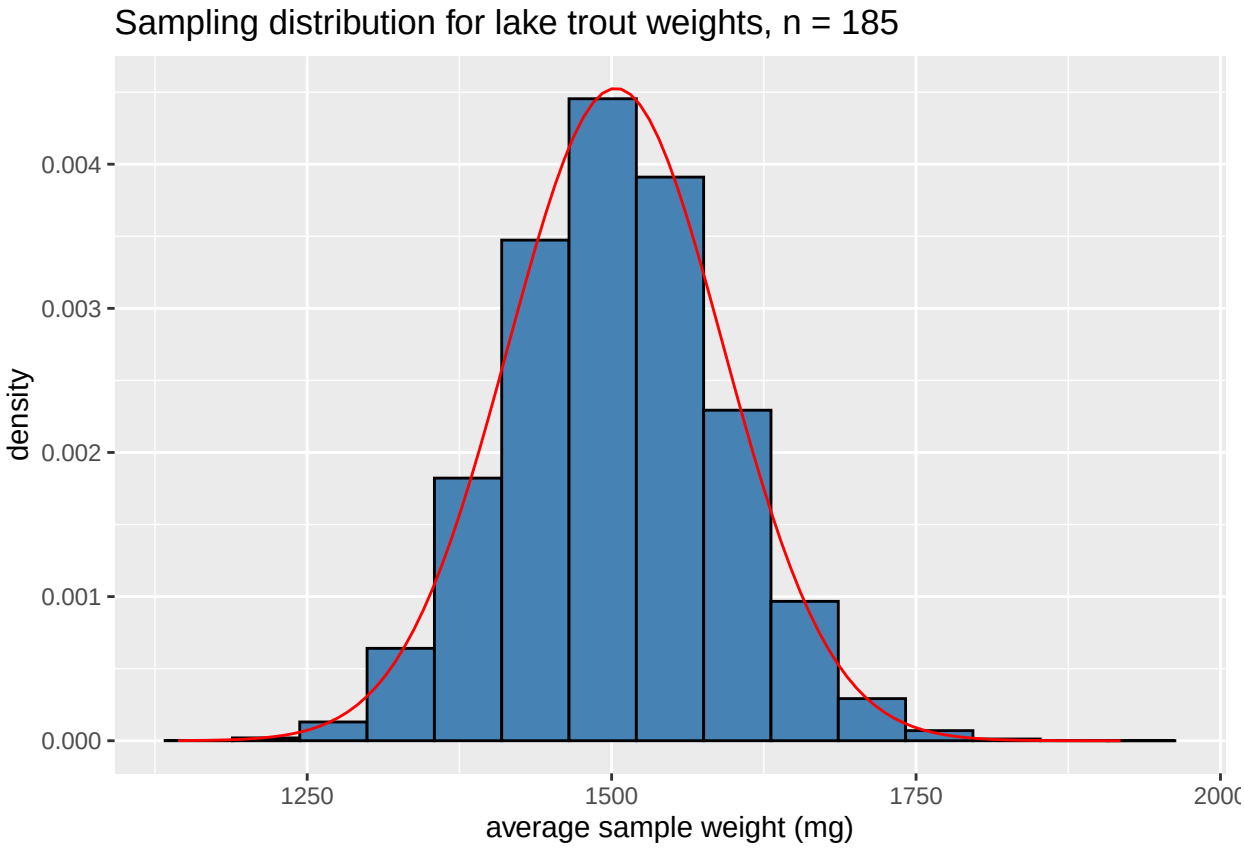
where μ, σ^2, and σ are the true population average, variance, and standard deviation, and s is the sample standard deviation.

The sample size is considered sufficiently large when your sample has 30 measurements.

The figure below shows our bootstrapped sampling distribution along with the normal distribution predicted by the Central Limit Theorem:

```
mu <- mean(lakeTrout$weight)
s <- sd( lakeTrout$weight)
n <- dim(lakeTrout)[1]

ggplot(xbarBootStrap, aes(x = xbar)) +
  geom_histogram( aes(y = ..density..), color = "black", fill= "steelblue", bins = 15)  +
  labs(x = "average sample weight (mg)", title = "Sampling distribution for lake trout weights, n = 185
  geom_function(fun = dnorm, args = list(mean = mu, sd = s/sqrt(n)), color = "red")
```



7. Use the information from above to calculate a 95% confidence interval to estimate the average weight

of lake trout in Flathead Lake. Recall the formula point est $\pm$ 1.96 $\times$ SE for a 95% confidence level. Interpret your confidence interval using complete sentences.

A final note on bootstrapping

Most of sampling distributions we bootstrapped could be approximated well using only the theoretical techniques we're learning in this class. You might be wondering what circumstances one would actually *need* bootstrapping. Some examples from wikipedia:

- When your sample size is too small to apply theoretical tools.
- When the distribution of your sample statistic is cumbersome, complicated, or unknown. An easy, yet surprising example of this is using the sample median to estimate a population median.

This Markdown template was taken from the Openintro Stats Labs.