

Foundations for statistical inference - Confidence intervals

Dhananjay Kumar

Sampling from Ames, Iowa

If you have access to data on an entire population, say the size of every house in Ames, Iowa, it's straight forward to answer questions like, "How big is the typical house in Ames?" and "How much variation is there in sizes of houses?". If you have access to only a sample of the population, as is often the case, the task becomes more complicated. What is your best guess for the typical size if you only know the sizes of several dozen houses? This sort of situation requires that you use your sample to make inference on what your population looks like.

The data

In the previous lab, "Sampling Distributions", we looked at the population data of houses from Ames, Iowa. Let's start by loading that data set.

```
load("more/ames.RData")
```

In this lab we'll start with a simple random sample of size 60 from the population. Specifically, this is a simple random sample of size 60. Note that the data set has information on many housing variables, but for the first portion of the lab we'll focus on the size of the house, represented by the variable `Gr.Liv.Area`.

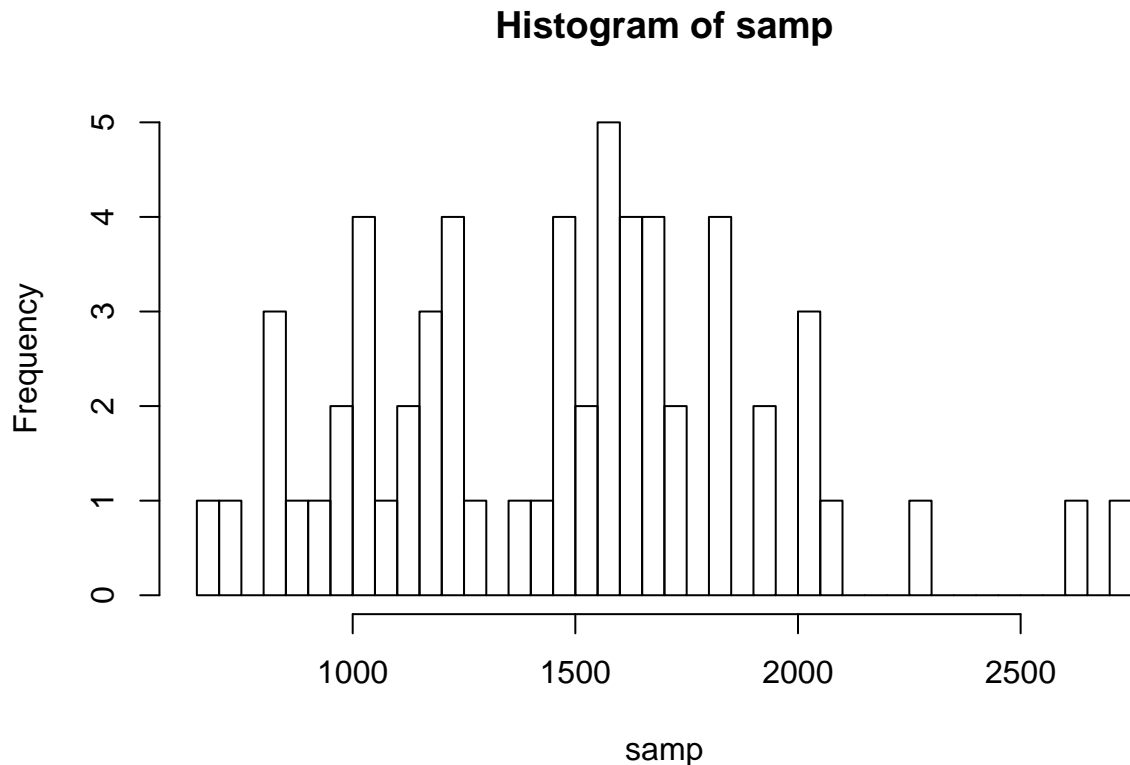
```
population <- ames$Gr.Liv.Area  
samp <- sample(population, 60)
```

1. Describe the distribution of your sample. What would you say is the "typical" size within your sample? Also state precisely what you interpreted "typical" to mean.

```
# Distribution Summary  
summary(samp)
```

```
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.   
##      670   1148   1498   1471   1707   2726
```

```
# Plot  
hist(samp,breaks=50)
```



Answer : The plot of the sample is unimodal and skewed towards the right. The typical area of the houses in Ames falls under 1000 - 1500 range.

2. Would you expect another student's distribution to be identical to yours? Would you expect it to be similar? Why or why not?

Answer : It might be similar but would be not be exactly same because of the small sample size.

Confidence intervals

One of the most common ways to describe the typical or central value of a distribution is to use the mean. In this case we can calculate the mean of the sample using,

```
sample_mean <- mean(samp)
```

Return for a moment to the question that first motivated this lab: based on this sample, what can we infer about the population? Based only on this single sample, the best estimate of the average living area of houses sold in Ames would be the sample mean, usually denoted as \bar{x} (here we're calling it `sample_mean`). That serves as a good *point estimate* but it would be useful to also communicate how uncertain we are of that estimate. This can be captured by using a *confidence interval*.

We can calculate a 95% confidence interval for a sample mean by adding and subtracting 1.96 standard errors to the point estimate (See Section 4.2.3 if you are unfamiliar with this formula).

```
se <- sd(samp) / sqrt(60)
lower <- sample_mean - 1.96 * se
upper <- sample_mean + 1.96 * se
c(lower, upper)
```

```
## [1] 1358.849 1583.584
```

This is an important inference that we've just made: even though we don't know what the full population looks like, we're 95% confident that the true average size of houses in Ames lies between the values *lower* and *upper*. There are a few conditions that must be met for this interval to be valid.

3. For the confidence interval to be valid, the sample mean must be normally distributed and have standard error s/\sqrt{n} . What conditions must be met for this to be true?

****Answer*** The following conditions should be met: 1. The observations should be independent. 2. The sample size should be more than 30. 3. The sample data should not be strongly skewed.

Confidence levels

4. What does "95% confidence" mean? If you're not sure, see Section 4.2.2.

Answer: It means that we are 95 % sure about the true population mean lying between interval (-2 Sd - +2 Sd) when plotting sampling distribution. See the range below:

```
lower
```

```
## [1] 1358.849
```

```
upper
```

```
## [1] 1583.584
```

In this case we have the luxury of knowing the true population mean since we have data on the entire population. This value can be calculated using the following command:

```
mean(population)
```

```
## [1] 1499.69
```

5. Does your confidence interval capture the true average size of houses in Ames? If you are working on this lab in a classroom, does your neighbor's interval capture this value?

```
summary(population)
```

```
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##      334    1126    1442    1500    1743    5642
```

Answer : Yes, if you see the mean of the population above, it does fall under lower and upper level of the sample calculated using 95% confidence.

6. Each student in your class should have gotten a slightly different confidence interval. What proportion of those intervals would you expect to capture the true population mean? Why? If you are working in this lab in a classroom, collect data on the intervals created by other students in the class and calculate the proportion of intervals that capture the true population mean.

Answer : 95% of the students will capture true population because of the fact that everyone of them would have used the following (Sample Mean + (+/- 1.96 * SE))

Using R, we're going to recreate many samples to learn more about how sample means and confidence intervals vary from one sample to another. *Loops* come in handy here (If you are unfamiliar with loops, review the Sampling Distribution Lab).

Here is the rough outline:

- Obtain a random sample.
- Calculate and store the sample's mean and standard deviation.
- Repeat steps (1) and (2) 50 times.
- Use these stored statistics to calculate many confidence intervals.

But before we do all of this, we need to first create empty vectors where we can save the means and standard deviations that will be calculated from each sample. And while we're at it, let's also store the desired sample size as `n`.

```
samp_mean <- rep(NA, 50)
samp_sd <- rep(NA, 50)
n <- 60
```

Now we're ready for the loop where we calculate the means and standard deviations of 50 random samples.

```
for(i in 1:50){
  samp <- sample(population, n) # obtain a sample of size n = 60 from the population
  samp_mean[i] <- mean(samp)    # save sample mean in ith element of samp_mean
  samp_sd[i] <- sd(samp)        # save sample sd in ith element of samp_sd
}
```

Lastly, we construct the confidence intervals.

```
lower_vector <- samp_mean - 1.96 * samp_sd / sqrt(n)
upper_vector <- samp_mean + 1.96 * samp_sd / sqrt(n)
```

Lower bounds of these 50 confidence intervals are stored in `lower_vector`, and the upper bounds are in `upper_vector`. Let's view the first interval.

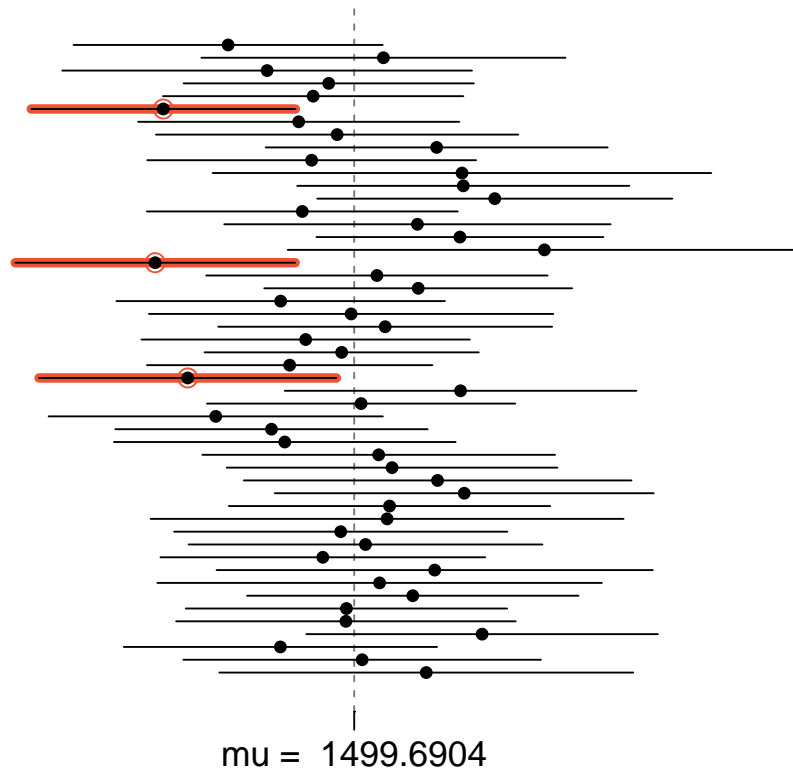
```
c(lower_vector[1], upper_vector[1])
```

```
## [1] 1399.979 1705.888
```

On your own

- Using the following function (which was downloaded with the data set), plot all intervals. What proportion of your confidence intervals include the true population mean? Is this proportion exactly equal to the confidence level? If not, explain why.

```
plot_ci(lower_vector, upper_vector, mean(population))
```



Answer : In the first run I got 46/50 i.e. 92% which is lower than 95%. The 95% confidence level does not mean that every run would have population mean lying between the sample mean interval. There would be cases where the population mean would be lying outside interval (Sample Mean + (+/- 1.96 * SE)).

- Pick a confidence level of your choosing, provided it is not 95%. What is the appropriate critical value?

Answer : We can choose the confidence level of 99% by following the given formula: (Sample Mean + (+/- 2.58 * SE)).

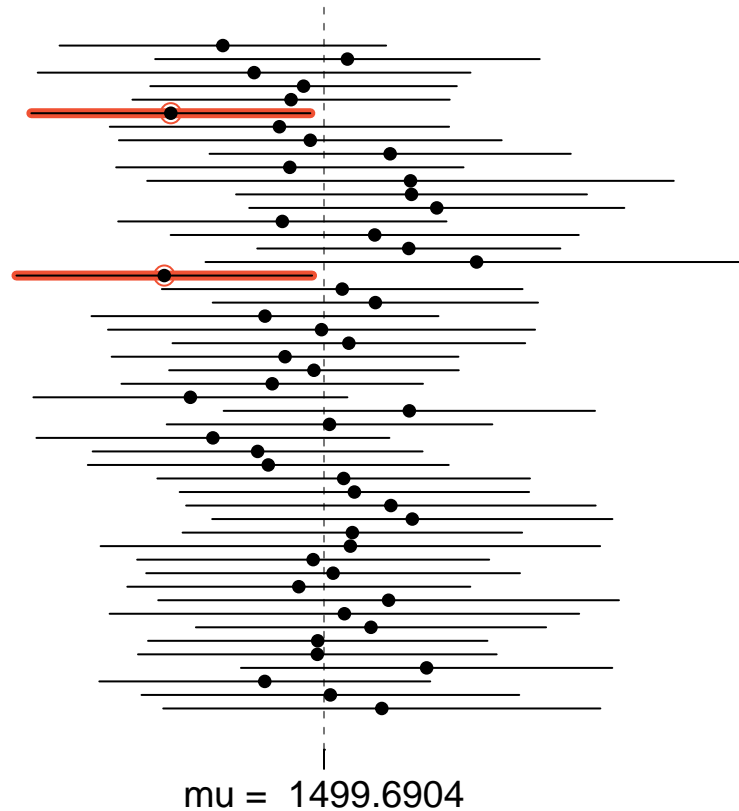
- Calculate 50 confidence intervals at the confidence level you chose in the previous question. You do not need to obtain new samples, simply calculate new intervals based on the sample means and standard deviations you have already collected. Using the `plot_ci` function, plot all intervals and calculate the proportion of intervals that include the true population mean. How does this percentage compare to the confidence level selected for the intervals?

Answer :

```

lower_vector <- samp_mean - 2.58 * samp_sd / sqrt(n)
upper_vector <- samp_mean + 2.58 * samp_sd / sqrt(n)
plot_ci(lower_vector, upper_vector, mean(population))

```



This is a product of OpenIntro that is released under a Creative Commons Attribution-ShareAlike 3.0 Unported. This lab was written for OpenIntro by Andrew Bray and Mine Çetinkaya-Rundel.