Lab 4b - Foundations for statistical inference - Confidence intervals

Michael Y.

March 17th, 2019

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.

```
summarysamp1 <- summary(samp)
summarysamp1
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   894.0  1141.8  1402.0  1528.6  1683.5  3390.0
```

```
minsamp1  <- as.numeric(summarysamp1["Min."])
meansamp1 <- round(as.numeric(summarysamp1["Mean"]),2)
medsamp1  <- as.numeric(summarysamp1["Median"])
maxsamp1  <- as.numeric(summarysamp1["Max."])
iqrsamp1  <- as.numeric(IQR(samp))
cat(paste("Inter-Quartile Range of the sample: ",iqrsamp1,"\n"))
```

```
## Inter-Quartile Range of the sample:  541.75
```

```
stdevsamp1 <- round(as.numeric(sd(samp)),2)
cat(paste("Standard Deviation of the sample: ",stdevsamp1,"\n"))
```

```
## Standard Deviation of the sample:  513.85
```

The 'typical' size of a house in this sample has a Median area of 1402 sq.ft. and a Mean area of 1528.62 sq.ft. Either measure (Mean or Median) is a measure of central tendancy. From the standard deviation of the areas of the houses in the sample we can establish a confidence interval around the sample mean. From this we can infer, to a specified degree of confidence, that the actual mean of the entire population falls within this confidence interval. Of course, if we are extremely "unlucky", our random draw could include only very small or very large houses, which would not be representative of the population, and thus "atypical."

End of response to Exercise 1 .

.#.

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

No, I would not expect another student's distribution to be identical to mine, but I would expect it to be similar. It's likely that it won't be exactly the same since it's a random sample, but it should be somewhat similar since they're coming from the same population.

End of response to Exercise 2 .

.#.

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] 1398.5948 1658.6385
```

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?

(a) The sample observations must be independent. If each random sample is less than 10% of the population, then it is likely that independence will hold. (If the sample size is a large portion of the population then it becomes increasingly likely that multiple samples will contain significant overlap, defeating the principle of independence.)

(b) The sample size must be sufficiently large; according to the textboook, it should contain at leans 30 independent observations.

(c) The population distribution must not be "strongly" skewed; in the case of larger sample size, we can be more lenient with this condition.

End of response to Exercise 3 .

.#.

Confidence levels

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

Suppose that we draw repeated random i.i.d. (independent and identically distributed) samples of a given size from a population with standard deviation σ .

Then the Standard Error of the Mean of the sampling distribution would be $SE = \frac{\sigma}{\sqrt{N}}$.

However, we may not know σ , the actual standard deviation of the population, in which case we have to estimate it using the standard deviation of the sample, $\sigma(\bar{x})$.

For each sample, we compute the sample mean $\bar{x}$ and a "confidence interval" of $\bar{x} \pm 1.96\frac{\sigma(\bar{x})}{\sqrt{N}}$.

Then we expect that the **true** population mean μ would fall within the respective confidence interval $\left(\bar{x} - 1.96\frac{\sigma(\bar{x})}{\sqrt{N}}, \bar{x} + 1.96\frac{\sigma(\bar{x})}{\sqrt{N}}\right)$on 95% of such random samples.

Thus, if we draw any individual random sample, we have 95% confidence that the true population mean μ would like within the confidence interval associated with such sample.

End of response to Exercise 4 .

.#.

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] 1499.6904
```

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?

```
mu <- mean(population)

### lower bound of 95% confidence interval
lower
```

```
## [1] 1398.5948
```

```
### population mean
mu
```

```
## [1] 1499.6904
```

```
### upper bound of 95% confidence interval
upper
```

```
## [1] 1658.6385
```

```
### Does the confidence interval contain the population mean, mu?
result <- contains(lower,upper,mu)
result
```

```
## [1] TRUE
```

YES : 1398.59 <= 1499.69 <= 1658.64 .

End of response to Exercise 5 .

.#.

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.

I would expect 95% of the intervals to capture the true population mean because this is how a 95% confidence interval works.

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.

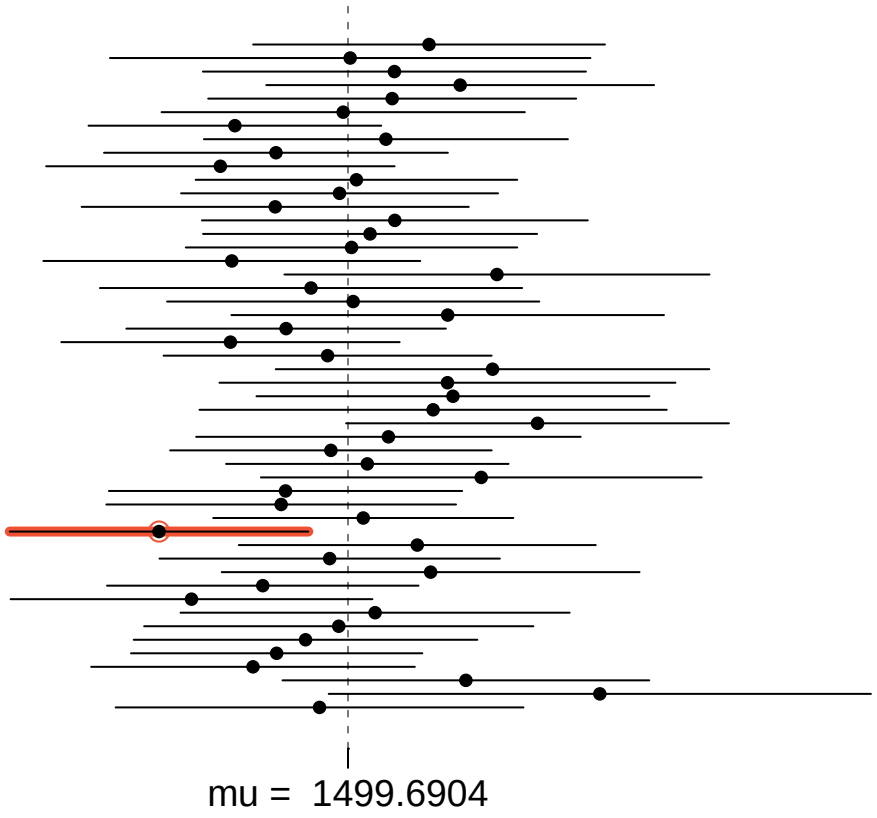
```
c(lower_vector[1], upper_vector[1])
```

```
## [1] 1343.6758 1617.4909
```

On your own

(a) ***Using the following function (which was downloaded with the data set), plot all intervals.***

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




```
##Put the values into a matrix
mat<-cbind(i=seq(50),lower_vector, upper_vector, mu=rep(mean(population),50))

#head(mat)
#tail(mat)

good <- mapply(FUN = contains,lower_vector,upper_vector,mu)
```

```
numgood <- sum(good)
cat("Number of successes: ", numgood, "\n")
```

```
## Number of successes:  49
```

```
pctgood <- (numgood / 50)*100
cat("Percentage successful", pctgood, "%\n")
```

```
## Percentage successful 98 %
```

```
bad <- !good
numbad <- sum(bad)
cat("Number of failures", numbad, "\n")
```

```
## Number of failures 1
```

```
pctbad <- (numbad / 50)*100
cat("Percentage out-of-range: ", pctbad, "%\n")
```

```
## Percentage out-of-range:  2 %
```

```
failing_rows <- mat[bad,]
cat("Failing rows, where population mean is outside of confidence interval:\n")
```

```
## Failing rows, where population mean is outside of confidence interval:
```

```
failing_rows
```

```
##            i lower_vector upper_vector           mu
##      14.0000    1272.6213    1473.0454    1499.6904
```

What proportion of your confidence intervals include the true population mean?

There are 49 samples (out of 50) which include the true population mean; this is 98 percent.

Is this proportion exactly equal to the confidence level? If not, explain why.

No, but it is as close as possible given the number of samples. To obtain exactly 95 percent, the number of samples would have to be a multiple of 20, as the desired result would be 19 out of 20; 38 out of 40; 57 out of 60, etc.

As the number of samples examined here was 50, it is not possible to obtain exactly 47.5 successes. However, the result obtained (49 out of 50) was close. (Repeatedly re-running the simulation will occasionally yield results which are as close as possible, i.e., 47 or 48 successes out of 50.)

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

```r
confidence_interval = 0.99
upper_tail = qnorm(0.995)
upper_tail
```

```
## [1] 2.5758293
```

```r
lower_tail = qnorm(0.005)
lower_tail
```

```
## [1] -2.5758293
```

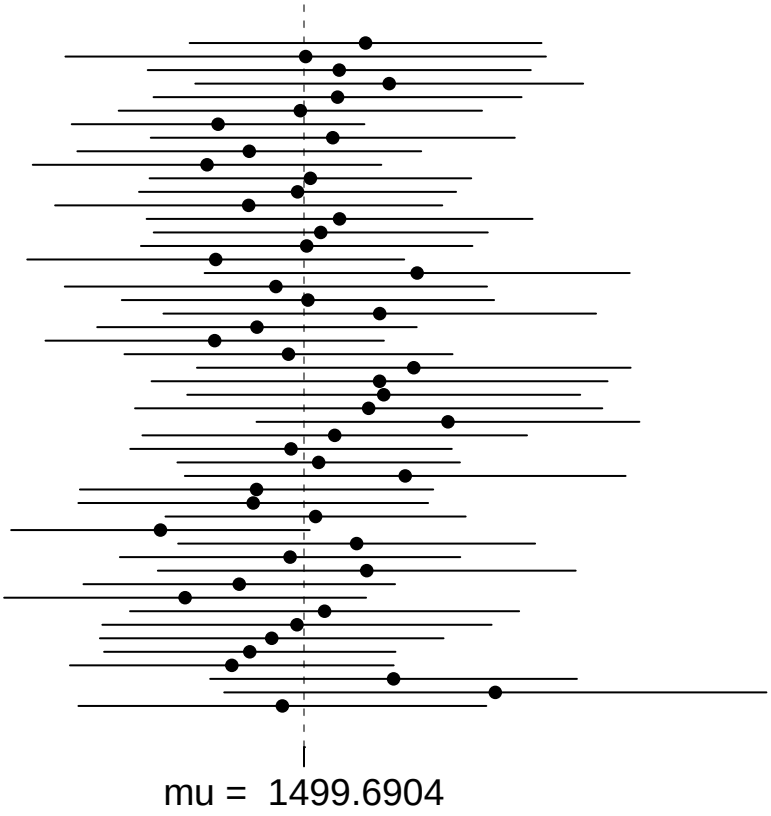
For a 99 percent confidence interval, the critical value is 2.58 (rather than the 1.96 used for the 95 percent confidence interval.)

(c) ***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.***

```r
lower_vector99 <- samp_mean - 2.58 * samp_sd / sqrt(n)
upper_vector99 <- samp_mean + 2.58 * samp_sd / sqrt(n)
```

Using the `plot_ci` function, plot all intervals

```r
plot_ci(lower_vector99, upper_vector99, mean(population))
```



calculate the proportion of intervals that include the true population mean.

```
##Put the values into a matrix
mat99<-cbind(i=seq(50),lower_vector99, upper_vector99, mu=rep(mean(population),50))

good99 <- mapply(FUN = contains,lower_vector99,upper_vector99,mu)
numgood99 <- sum(good99)
cat("Number of successes at 99% confidence: ", numgood99, "\n")
```

```
## Number of successes at 99% confidence:  50
```

```
pctgood99 <- (numgood99 / 50)*100
cat("Percentage successful at 99% confidence: ", pctgood99, "%\n")
```

```
## Percentage successful at 99% confidence:  100 %
```

```
bad99 <- !good99
numbad99 <- sum(bad99)
cat("Number of failures at 99% confidence: ", numbad99, "\n")
```

```
## Number of failures at 99% confidence:  0
```

```
pctbad99 <- (numbad99 / 50)*100
cat("Percentage out-of-range: ", pctbad99, "%\n")
```

```
## Percentage out-of-range:  0 %
```

```
failing_rows99 <- mat99[bad99,]
cat("Failing rows, where population mean is outside of 99% confidence interval:\n")
```

```
## Failing rows, where population mean is outside of 99% confidence interval:
```

```
failing_rows99
```

```
##      i lower_vector99 upper_vector99 mu
```

How does this percentage compare to the confidence level selected for the intervals?

Given that there were 50 samples, at a 99% confidence interval we would expect the number of failures (i.e., samples which do not include the population mean in the confidence interval) to be 0.5 samples.

As it is not possible to achieve this fractional result, the closest we can come would be 0 or 1 failure.

For this sample, the number of failures was 0 while the number of successes was 50 .

The equivalent percentages are failures: 0% ; successes: 100%.

This is as close as we can come to the confidence interval using 50 samples; we would need to increase the number of samples to a multiple of 100 in order to have an exact result of 99% success; 1% failure.