

Lab 4b - Foundations for statistical inference - Confidence intervals

Michael Y.

October 21st, 2018

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.  
##  334.0  1111.0  1326.0  1491.8  1803.0  4316.0
```

```
minsamp1 <- as.numeric(summarysamp1["Min."])  
meansamp1 <- round(as.numeric(summarysamp1["Mean"]),2)  
medsamp1 <- as.numeric(summarysamp1["Median"])  
maxsamp1 <- as.numeric(summarysamp1["Max."])  
iqr samp1 <- as.numeric(IQR(samp))  
cat(paste("Inter-Quartile Range of the sample: ",iqr samp1,"\n"))
```

```
## Inter-Quartile Range of the sample: 692
```

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

```
## Standard Deviation of the sample: 627.97
```

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?

End of response to Exercise 2 .

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.

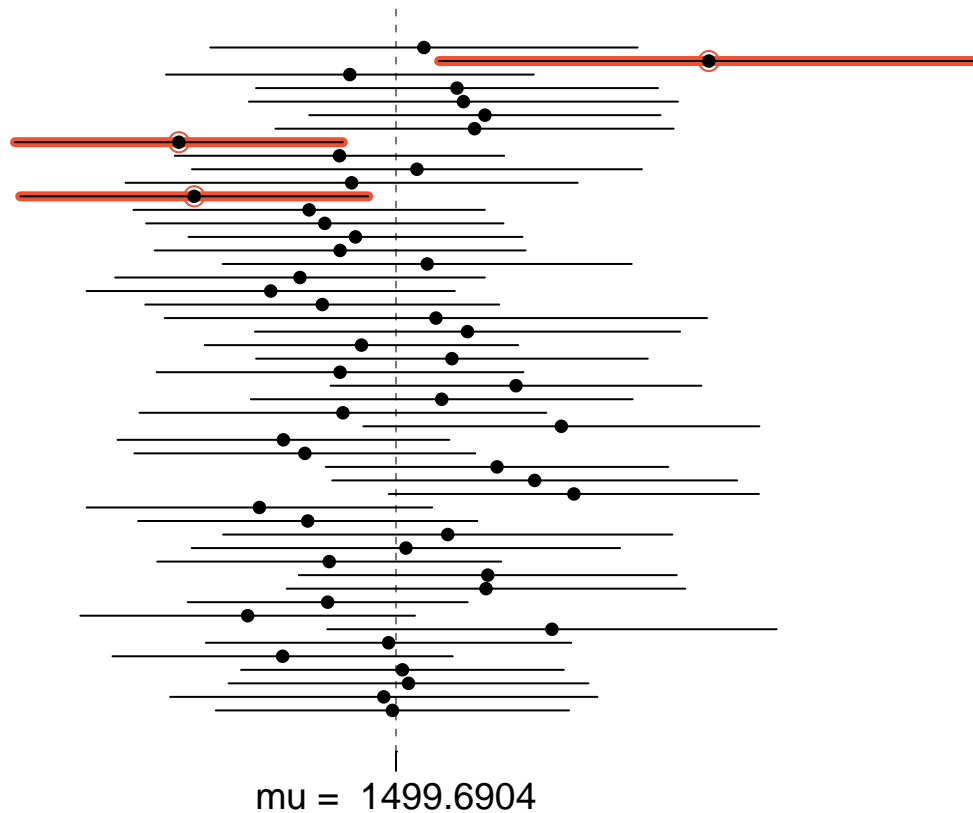
```
c(lower_vector[1], upper_vector[1])
```

```
## [1] 1383.2164 1611.5836
```

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: 47

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

## Percentage successful 94 %

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

## Number of failures 3

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

## Percentage out-of-range: 6 %

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
## [1,] 39    1256.7485    1481.7182 1499.6904
## [2,] 43    1253.3579    1465.3087 1499.6904
## [3,] 49    1527.5500    1876.3500 1499.6904

```

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

There are 47 samples (out of 50) which include the true population mean; this is 94 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 (47 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?*

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

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

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

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

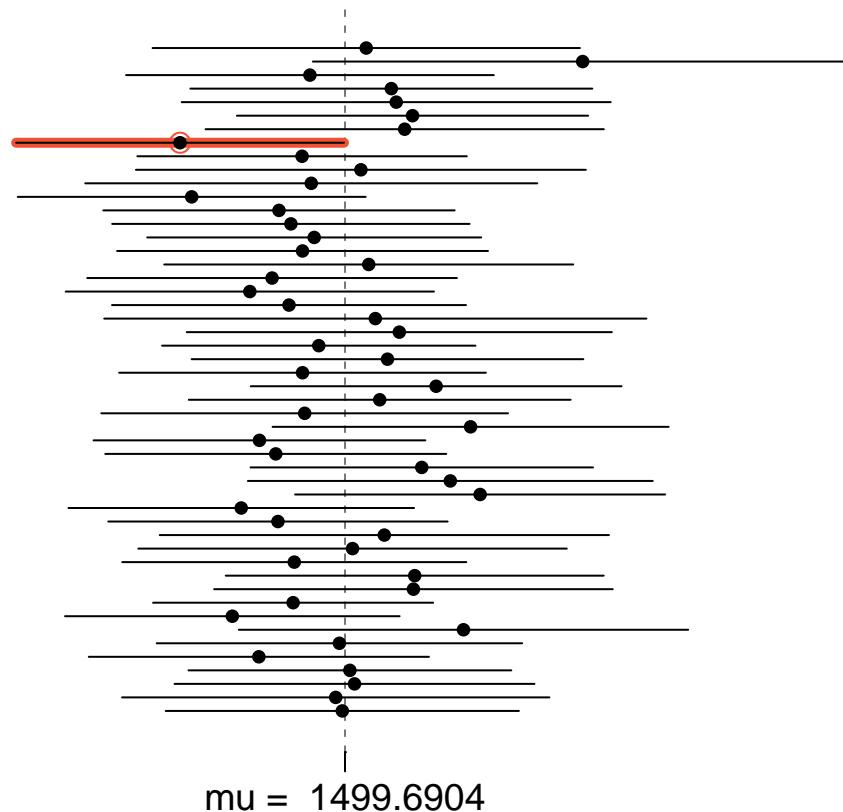
For a 99 percent confidence interval, the critical value is 2.58 (rather than the 1.96 used for the 95 percent confidence interval.)

(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.*

```
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

```
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: 49

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

## Percentage successful at 99% confidence: 98 %

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

## Number of failures at 99% confidence: 1

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

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



```
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
##      43.0000      1219.8351      1498.8316      1499.6904
```

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 1 while the number of successes was 49 .

The equivalent percentages are failures: 2% ; successes: 98%.

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.

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.