R Demonstration – Confidence Intervals

Objective: The purpose of this session is to demonstrate the calculation of a confidence interval for a sample mean in R and to introduce a script that illustrates the proper interpretation of confidence intervals.

Part I. Calculating a Confidence Interval in R

As shown on page 76 of the Gotelli & Ellison (2004) text, the general formula for a confidence interval is:

$$P\left(\bar{Y} - t_{\alpha[n-1]} \cdot s_{\bar{Y}} \leq \mu \leq \bar{Y} + t_{\alpha[n-1]} \cdot s_{\bar{Y}}\right) = (1-\alpha)$$

where $\bar{Y}$ is the sample mean, $s_{\bar{Y}}$ is the standard error of the mean, n is the sample size, and $t_{\alpha[n-1]}$ is the critical value of a t-distribution with probability $P = \alpha$. Thus, we can use our estimates of the sample mean and standard error to compute a 95% confidence interval (i.e., $\alpha = 0.05$) with the following form:

$$\left[\bar{Y} + t_{0.025[n-1]} \cdot s_{\bar{Y}}, \bar{Y} + t_{0.975[n-1]} \cdot s_{\bar{Y}}\right]$$

Let us now see how to calculate a 95% confidence interval in R using the formula above. We will again use our hypothetical normal distribution of the lengths of the tibial spines of a population of linyphiid spiders that has a mean of 0.253 mm and a standard deviation of 0.04 mm. Start the R software and type the following:

```
> mu <- 0.253
> sigma <- 0.04
```

Next, we will create a random sample of size `n=100` using the rnorm function:

```
> n <- 100
> sample <- rnorm(n, mean=mu, sd=sigma)
```

Now we can calculate the mean and standard deviation of our sample, and then divide the standard deviation by the square root of `n` to get the standard error of the mean :

```
> sample_mean <- mean(sample)
> std_error <- sd(sample)/sqrt(n)
```

Finally, we can use the `qt` function to calculate $t_{0.025[n-1]}$ and $t_{0.975[n-1]}$. To calculate $t_{0.025[n-1]}$, we use the following command:

```
> qt(0.025, df=n-1)
[1] -1.984217
```

This means that the critical value of a t-distribution with `P = 0.025` and `df=n-1` degrees of freedom is equal to -1.984217. We can calculate the critical value $t_{0.975[n-1]}$ for the upper end of the confidence interval in a similar fashion:

```
> qt(0.975, df=n-1)
[1] 1.984217
```

Notice that this is the same as the critical value we calculated for $t_{0.025[n-1]}$, only the sign is the opposite. Since this is always the case, we can simplify our calculation of the 95% confidence interval to the following:

$$\left[\overline{Y} \pm t_{0.975[n-1]} \cdot s_{\overline{Y}}\right]$$

Now we can save our value for $t_{0.975[n-1]}$ in a variable named `t_critical`, and then we can use the formula above to calculate the upper and lower limits of our 95% confidence interval for the sample mean:

```
> t_critical <- qt(0.975, df=n-1)
> CI_upper <- sample_mean + (t_critical*std_error)
> CI_lower <- sample_mean - (t_critical*std_error)
```

Finally, we use the R function *`cat`* to enclose our confidence interval in brackets and print it out to the console (NOTE: your results will be different because of the random nature of the sample created with the `rnorm` function earlier):

```
> cat("[", CI_lower, ", ", CI_upper, "]\n", sep="")
[0.2500039, 0.2680813]
```

The *`cat`* function may look a bit awkward, but basically it tells R to concatenate all of the strings (i.e., the characters between quotation marks) with the variables provided in the parentheses. Thus, in the *`cat`* command shown above, R will write a left-bracket "[" character to the console, then insert the value in the variable `CI_lower`, then a comma followed by a space character, then the value in the variable `CI_upper`, and finally a right-bracket "]" character followed by a code indicating a new line ("\n"). The `sep=""` argument indicates that there should be no separator character between each part of the concatenation sequence. Thus, we see that our 95% confidence interval for the mean of this particular sample has a lower limit of `0.2500039` and upper limit of `0.2680813`. But what exactly does this mean? We will use the R script presented in the following section to illustrate the proper interpretation of a confidence interval.

Part II. R Script: Interpretation of Confidence Intervals

Download the `Confidence_Intervals.R` script from the course website and save it in your `PCB6466` folder. Make sure you've changed the R working directory to point to this folder, open the script, and then select *`Edit Run all`* from the menu to execute it.

```
####################################################################
## Confidence_Intervals.R                                         ##
####################################################################
## This script draws a specified number of random samples of a     ##
## specified size from a normal distribution with a known          ##
## population mean and standard deviation. It then computes 95%    ##
## confidence intervals for all of the samples and plots them.     ##
####################################################################

## USER-DEFINED VARIABLES ##

## number of random samples to generate
num_samples <- 100

## size of each sample
sample_size <- 100

## population mean
mu <- 0.253

## population standard deviation
sigma <- 0.04


## PROGRAM CODE ##

## use the standard error of the mean to calculate the Y-axis limits
std_error <- sigma/sqrt(sample_size)
y_axis_interval <- c(mu-5*std_error, mu+5*std_error)

## set the plotting options and draw a horizontal red line at the population
mean
plot(c(0, num_samples), y_axis_interval, type="n", xlab="Sample number",
ylab="95% CI of sample mean")
abline(h=mu, col="red")

## use a FOR loop to generate the specified number of samples
for(i in 1:num_samples) {

	## draw the sample from the random normal distribution
	sample <- rnorm(sample_size, mean=mu, sd=sigma)

	## calculate the sample mean and standard error
	sample_mean <- mean(sample)
	sample_std_error <- sd(sample)/sqrt(sample_size)

	## calculate the upper and lower limits of the 95% confidence interval
	CI_upper <- sample_mean + qt(0.975, sample_size-1)*sample_std_error
	CI_lower <- sample_mean - qt(0.975, sample_size-1)*sample_std_error

	## plot the 95% confidence interval
	points(c(i, i), c(CI_upper, CI_lower), type="b")
}
```

Part III. Exercise 4

1. Experiment with the `Confidence_Intervals.R` script by changing the values of the four variables in the `USER-DEFINED VARIABLES` section of the script. For example, you may wish to modify the `sample_size` variable (while holding all the other variables constant) to see what effect a larger sample size has on the computation of the confidence intervals. Present your results in an organized table that clearly shows what variable was changed and which were held constant, as well as important outputs. Discuss the implications of what you found for experimental design and data analysis.

2. Modify the `Confidence_Intervals.R` script to compute confidence intervals of different sizes (e.g., 90% and 99%). As a challenge, try to add another variable to the `USER-DEFINED VARIABLES` section of the script that controls the size of the confidence intervals to be calculated. In addition to adding this variable (you may wish to call it `CI_size`, for example), you should only need to modify a few lines of code in the script. Discuss the implications of what you found for experimental design and data analysis.