Learning Outcomes

1) Understand how to extract important values/estimates using R Studio
2) Use R Studio to calculate confidence intervals using the Z distribution
3) Use R Studio to calculate confidence intervals using the t distribution

Today we will use R Studio to calculate confidence intervals.

Generating Data and Extracting Values

Copy/type the code below into an R file and make sure you understand what each element is doing.

```
#create a vector x0 of length 1000, drawn from a normal distribution with mean 3, sd 5

x0 <- rnorm(1000,3,5)


# record the size, mean and sd of x0

N0 <- length(x0)

mean0 <- mean(x0)

sd0 <- sd(x0)


#Take a random sample of size 30 from x0

x1 <- sample(x0,30)


# record the size, mean and sd of x1

n1 <- length(x1)

mean1 <- mean(x1)

sd1 <- sd(x1)


#calculate the standard error of mean1 using known and estimated sd0

se_kn <- sd0/sqrt(n1)

se_unkn <- sd1/sqrt(n1)
```

Using R to get Critical Values

The qnorm and qt functions can be used to find the critical values for the Z-distribution and t-distributions, respectively.

By default, the qnorm(p) function returns the z-score for P(x <= p). qnorm(p, lower.tail = FALSE) returns the z-score for P(z >a). "p" here is a letter I am using to represent some probability between 0 and 1.

Copy/type the code below into your R file and run it

```
Z1 <- qnorm(0.975)

Z2 <- qnorm(0.995)

Z1

Z2
```

Now open your Normal distribution table and find 0.975 and 0.995 in the body of the table. Find the z-scores that they correspond to in the row and column headers. Do they correspond to the R output? Make sure you understand why you use qnorm(0.975) to find the z-score for a 95% confidence interval.

The qt(p, df) function is similar to qnorm, but it returns the <u>critical value</u> for for P(x <= p). qt(p, df, lower.tail = FALSE) returns the critical value for P(z >p).

You can look at the help file for qt in R by typing ?qt. When I do this, under "usage" I see:
qt(p, df, ncp, lower.tail = TRUE, log.p = FALSE)
and then it defines that p is a probability and df is the degrees of freedom.

If I type qt(0.975, 10) I get the critical value below which 97.5% of values from a t distribution with 10 degrees of freedom lie. I need to specify the degrees of freedom, not the sample size n (the reason for this is because depending on how I'm using the t distribution, the degrees of freedom will not always be n-1).

Copy/type the code below into your R file and run it

```
t1 <- qt(0.975, 10)

t2 <- qt(0.995, 10)

t1

t2
```

Now open your t distribution table and find the column headers corresponding to the critical values t1 and t2 for 10 degrees of freedom. Are they what you expected? Make sure you understand why you use qt(0.975, df) to find the critical t-value for a 95% confidence interval, but would look up 0.05 in the 2-tailed t-distribution table.

Calculating Confidence Intervals

Copy/type the code below into your R file. Make sure you understand what each element is doing.

```
#Calculate Z-score for 95% CI
Z_score <- qnorm(0.975)

#Calculate critical t-value for 95% CI and x1
t_score <- qt(0.975, n1-1)

#Calculate 95% CIs for x1 using known and unknown population sd, and
t and z distributions
left_Z95_kn <- mean1-Z_score*se_kn
right_z95_kn <- mean1+Z_score*se_kn
left_Z95_unkn <- mean1-Z_score*se_unkn
right_z95_unkn <- mean1+Z_score*se_unkn

left_t95_kn <- mean1-t_score*se_kn
right_t95_kn <- mean1+t_score*se_kn
left_t95_unkn <- mean1-t_score*se_unkn
right_t95_unkn <- mean1+t_score*se_unkn

#show results
paste("var: x0", "mean:", mean0, "sd:", sd0)
paste("var: x1", "mean:", mean1, "sd:", sd1, "se w/known pop sd:",
se_kn, "se w/unknown pop sd:", se_unkn)
paste("95% CIs for estimate of population mean")
paste("Z-distribution w/known pop sd:", left_Z95_kn, right_z95_kn)
paste("Z-distribution w/unknown pop sd:", left_Z95_unkn,
right_z95_unkn)
paste("t-distribution w/known pop sd:", left_t95_kn, right_t95_kn)
paste("t-distribution w/unknown pop sd:", left_t95_unkn,
right_t95_unkn)
```

Plots of Normality

Copy/type the code below into your R file. Make sure you understand what each element is doing.

```
#make a qqplot and a histogram with normal density curve for x0

qqnorm(x0)

qqline(x0)

hist(x0, freq = FALSE)

xfit <- seq(min(x0), max(x0), length = 40)

yfit <- dnorm(xfit, mean = mean(x0), sd = sd(x0))

lines(xfit, yfit)
```